



ZILMAR SOUSA SILVA

**CONSTRUINDO ROTEADORES WI-MESH
COM GNU/LINUX E OLSR**

LAVRAS – MG

2011

ZILMAR SOUSA SILVA

CONSTRUINDO ROTEADORES WI-MESH COM GNU/LINUX E OLSR

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do Curso de Pós-Graduação *Lato Sensu* em Administração de Redes *Linux*, para a obtenção do título de Especialista em Administração de Redes *Linux*.

Orientador

Prof. Dr. Joaquim Quinteiro Uchôa

LAVRAS - MG

2011

ZILMAR SOUSA SILVA

CONSTRUINDO ROTEADORES WI-MESH COM GNU/LINUX E OLSR

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do Curso de Pós-Graduação Lato Sensu em Administração de Redes *Linux*, para a obtenção do título de Especialista em Administração de Redes *Linux*.

Aprovada em 24 de Setembro de 2011

Prof. Msc. Denilson Vedoveto Martins

Prof. Msc. Herlon Ayres Camargo

Prof. Dr. Joaquim Quinteiro Uchôa
(Orientador)

LAVRAS - MG

2011

Dedico...

Aos meus pais Arnaldo e Ana Maria por sempre terem me incentivado os estudos e me mostrado o caminho correto a seguir.

Aos meus irmãos e sobrinhos, para que isso sirva algum dia de motivação para que os mesmos sigam os caminhos dos estudos.

A minha querida esposa Nelma, porque sempre estive comigo nos momentos que mais precisei, e principalmente pela paciência nas noites mal dormidas.

A minha querida filha Ketlyn, para que isso te sirva de inspiração na suas jornada estudantil.

A minha querida filha Luana, que me ensinou o amor verdadeiro e me motivou a continuar esta louca viagem que se chama viver.

AGRADECIMENTOS

Agradeço primeiramente a Deus que me deu forças e sabedoria para chegar até o fim deste curso de Pós-Graduação.

Agradeço à Departamento de Ciência da Computação da Universidade Federal de Lavras por ter me dado o prazer de estudar em umas das melhores universidades do país.

Um agradecimento especial ao coordenador do curso de Pós-Graduação em Administração de Redes Linux, Prof. Dr. Joaquim Quinteiro Uchôa por acreditar na educação e ter me dado a oportunidade de ser seu aluno.

Agradeço a todos os professores, que nos enriqueceram com seus conhecimentos e experiências profissionais, que tenho certeza irá me ajudar muito na jornada daqui para frente.

Agradeço a meu pai Arnaldo José da Silva que apesar de nunca ter tido a oportunidade de estudar, sempre lutou de sol a sol, para que eu pudesse ir a escola, e construir um futuro melhor.

A minha mãe Ana Maria Sousa Silva, que com toda sua doçura, sempre teve uma palavra amiga, um conselho para dar e sempre mostrou que um futuro melhor só se constrói com a educação.

Agradeço a minha esposa pelo companheirismo, compreensão, e principalmente pela paciência, pois foram várias noites mal dormidas me esperando terminar de estudar.

Por fim, agradeço a todos que contribuíram direta ou indiretamente para a realização desta trabalho.

“A internet é um tipo de serviço que só é opcional para quem não a utiliza. Pela sua importância, em face daquilo que ela pode proporcionar, o acesso à internet deveria ser considerado um direito básico do cidadão”.

VEDANA, 2005.

“Redes sem fio estão por toda a parte. Algum dia, teremos receptores sem fio embutidos em nossas cabeças”

SCHRODER, 2009.

“O temor do Senhor é o princípio do conhecimento...”

Provérbios 1:7.

RESUMO

Redes sem fio estão por toda a parte. Isso se deve a grandes avanços tecnológicos e as quedas de preços nos produtos disponíveis para configuração de WLANs. Não obstante a isso tecnologias disponíveis para WMANs não acompanham os seus primos WLANs no quesito preço, haja vista que são tecnologias menos populares e que muitas vezes necessitam de mais recursos, e isso torna muito caro a disponibilização de WMANs em altas velocidades. Nos últimos tempos por necessidades ou modismo, várias cidades iniciaram e desenvolveram projetos de construção de cidades digitais, disponibilizando acesso a *internet* para a comunidade. Alguns casos foram bem sucedidos outros nem tanto. O fato é que muitas empresas e projetistas de redes a fim de economizar no custo final de implementação preferem utilizar tecnologias não recomendadas para construção de uma infraestrutura mais robusta, já que estes equipamentos são muito mais baratos que os convencionais utilizados em MANs sem fio. Estes equipamentos são principalmente antenas e roteadores sem fio, pois são estes os equipamentos com custo mais considerável dentro de um grande projeto de rede sem fio. Implementações recentes demonstram que a configuração de roteadores *wireless* que utilizam protocolos proativos em plataforma *free software* com GNU/Linux, para montagem de redes sem fio utilizando um topologia em malha, facilitam a configuração projetos de cidades digitais a um custo de infraestrutura reduzido. Neste será provado que é possível a utilização de um dispositivo altamente robusto, confiável, personalizável, gerenciável e disponibilizando uma topologia que utilize o mesmo conceito de roteamento utilizado na *internet*, através da implementação e configuração do protocolo proativo OLSR em plataforma *free software* GNU/Linux.

Palavras Chave: OLSR, WMANs, *Wi-Mesh*, Redes Virais, MANETs.

LISTA DE FIGURAS

Figura 1	Espectro de frequências.....	25
Figura 2	Rede sem fio em modo infraestrutura.....	29
Figura 3	Rede sem fio em modo <i>Ad Hoc</i> (sem infraestrutura).....	30
Figura 4	Logo que identifica produtos da <i>Wi-Fi Alliance</i> em todo o mundo.	33
Figura 5	Descoberta de rotas a partir da fonte S para o destino D.....	37
Figura 6	Formato de um pacote de mensagens HELLO.....	41
Figura 7	Inundação Normal e Inundação com MPR.....	43
Figura 8	Formato de um pacote de mensagens TC.....	45
Figura 9	Formato de um pacote de mensagens HNA.....	46
Figura 10	<i>Layout</i> inicial da rede experimental de testes.....	52
Figura 11	Visão do arquivo de configuração do OLSRD.....	56
Figura 12	Visão do arquivo de configuração do OLSRD nos clientes.....	59
Figura 13	Configuração de endereços Windows.....	61
Figura 14	Conectando na rede wimesh.....	62
Figura 15	<i>Screenshot</i> da tela de configuração do aplicativo OLSR <i>Swith</i>	62
Figura 16	<i>Layout</i> de rede apresentado para teste no cenário 1.....	64
Figura 17	<i>Screenshot</i> do <i>host B</i> , mostrando as rotas para seus vizinhos.....	64
Figura 18	<i>Screenshot</i> do <i>host A</i> , mostrando as rotas para seus vizinhos.....	65
Figura 19	<i>Screenshot</i> do <i>host C</i> , mostrando as rotas para seus vizinhos.....	66
Figura 20	<i>Layout</i> de rede apresentado para teste no cenário 2.....	66
Figura 21	<i>Screenshot</i> do <i>host B</i> , mostrando as rotas para seus vizinhos.....	67
Figura 22	<i>Screenshot</i> do <i>host A</i> , mostrando as rotas para seus vizinhos.....	68
Figura 23	<i>Screenshot</i> do <i>host D</i> , mostrando os <i>hosts</i> vizinhos.....	68
Figura 24	<i>Screenshot</i> do <i>host D</i> , mostrando as rotas para seus vizinhos.....	69
Figura 25	<i>Layout</i> de rede apresentado para teste no cenário 2.....	10
Figura 26	<i>Screenshot</i> do <i>host B</i> , mostrando os <i>hosts</i> vizinhos.....	10
Figura 27	<i>Screenshot</i> do <i>host A</i> , mostrando os <i>hosts</i> vizinhos	72

Figura 28 <i>Screenshot</i> do <i>host C</i> , mostrando os <i>hosts</i> vizinhos	73
Figura 29 <i>Screenshot</i> do <i>wireshark</i> , pacote de mensagem HELLO.....	74
Figura 30 <i>Screenshot</i> do <i>wireshark</i> , pacote de mensagem HNA.....	75
Figura 31 <i>Screenshot</i> do <i>wireshark</i> , pacote HNA enviado por <i>broadcast</i>	75
Figura 32 <i>Screenshot</i> do <i>wireshark</i> , pacote de mensagem TC e HELLO.....	76
Figura 33 <i>Layout</i> da RMP/Cidade Digital.....	84

LISTA DE TABELAS

Tabela 1	Classificação de frequências.....	26
Tabela 2	Exemplos de redes <i>mesh</i> comunitárias com o OLSRD.....	50
Tabela 3	Configurações dos <i>hosts</i> dentro do <i>layout</i> da rede.....	63

LISTA DE ABREVIATURAS E SIGLAS

ANATEL	Agência Nacional de Telecomunicações
AODV	<i>Ad Hoc On-Demand Distance Vector Routing</i>
AODV-UU	<i>AODV - Universidade de Uppsala</i>
AP	<i>Access Point</i>
APT	<i>Advanced Packaging Tool</i>
BSA	<i>Basic Service Area</i>
BSS	<i>Basic Service Set</i>
C	Linguagem de programação de uso geral
C++	Linguagem de programação orientada a objetos de uso geral
CRC	Centro de Pesquisa do Canadá de Comunicação
DSDV	<i>Destination-Sequenced Distance Vector Routing</i>
DSL	<i>Digital Subscriber Line</i>
DSR	<i>Dynamic Source Routing Protocol</i>
EHF	<i>Extremely High Frequency</i>
ELF	<i>Extremely Low Frequency</i>
ESSID	<i>Extended Service Set Identifier</i>
Ghz	<i>Gigahertz</i>
GNU	<i>GNU is Not Unix</i>
GRC	Universidade Politécnica de Valencia
HF	<i>High Frequency</i>
HNA	<i>Host and Network Association</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
INRIA	<i>Institut National de Recherche en Informatique et en Automatique</i>
IPv4	<i>Internet Protocol version 4</i>
IPv6	<i>Internet Protocol version 6</i>
ISM	<i>Industrial, Scientific, Medical</i>

ITU	<i>International Telecommunication Union</i>
ITU-T	<i>Telecommunication Standardization Sector of ITU</i>
Kbps	<i>Kilobits por segundo</i>
LAN	<i>Local Area Network</i>
LF	<i>Low Frequency</i>
LRI	<i>Laboratoire de Recherche en Informatique</i>
MAN	<i>Metropolitan Area Network</i>
MANET	<i>Mobile Ad Hoc Network</i>
Mbps	<i>Megabits por segundo</i>
MF	<i>Medium Frequency</i>
MID	<i>Multiple Interface Declaration</i>
MPR	<i>Multipoint Relays</i>
NAT	<i>Network Address Translation</i>
NRL	<i>Naval Research Laboratory</i>
NS2	<i>Network Simulator</i>
OLPC	<i>One Laptop per Child</i>
OLSR	<i>Optimized Link State Routing Protocol</i>
ONG	Organização não Governamental
OOLSR	<i>Implementation of the OLSR</i>
OSI	<i>Open Systems Interconnection</i>
PC	<i>Personal Computer</i>
PDA	Personal digital assistant
RFC	<i>Request for Comments</i>
RMP	Rede Municipal de Pesquisa
RREQ	<i>Route Request</i>
SHF –	<i>Super High Frequency</i>
TCP	<i>Transmission Control Protocol</i>
TCP/IP	<i>Transmission Control Protocol / Internet Protocol.</i>
TI	Tecnologia da Informação

UDP	<i>User Datagram Protocol</i>
UHF	<i>Ultra High Frequency</i>
VHF	<i>Very High Frequency</i>
VLf	<i>Very Low Frequency</i>
WAN	<i>Wide Area Network</i>
Wi-Fi	<i>Wireless Fidelity</i>
WiMax	<i>Worldwide Interoperability for Microwave Access</i>
Wi-Mesh	<i>Wireless Mesh Network</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
WLAN	<i>Wireless Local Area Network</i>
WMAN	<i>Wireless Metropolitan Area Network</i>
WWAN	<i>Wireless Wide Area Network</i>

SUMÁRIO

1 INTRODUÇÃO.....	16
1.1 Objetivos.....	17
1.2 Metodologia.....	17
1.3 Estrutura do Texto.....	18
2 CIDADES DIGITAIS.....	20
2.1 Comentários iniciais.....	20
2.2 Benefícios de uma cidade digital.....	20
2.3 Como funcionam as cidades digitais.....	21
2.4 Tecnologias aplicadas às cidades digitais.....	22
2.5 Comentários Finais.....	23
3 REDES WIRELESS.....	25
3.1 Classificação das redes wireless	28
3.1.1 Redes wireless com infraestrutura.....	28
3.1.2 Redes wireless sem infraestrutura.....	29
3.2 Redes wireless - Classificação por área de abrangência.....	30
3.2.1 Wireless wide area network (Wireless WANs).....	30
3.2.2 Wireless metropolitan area network (Wireless MANs).....	31
3.2.3 Wireless local area network (Wireless LANs).....	31
3.3 Padrões de redes wireless.....	31
3.3.1 IEEE 802.11 - Wi-Fi.....	32
3.3.2 IEEE 802.16 – WiMAX.....	33
3.3.3 IEEE 802.11s – Wi-Mesh.....	34
3.4 Redes em Malha – Wi-Mesh.....	34
3.4.1 Protocolos de roteamento em redes mesh.....	35
3.4.2 Protocolos proativos.....	35

3.4.3 DSDV - Destination-sequenced distance vector routing.....	36
3.4.4 Protocolos reativos.....	36
3.4.5 DSR - Dynamic source routing protocol.....	36
3.4.6 AODV – Ad Hoc on-demand distance vector routing.....	37
3.5 OLSR: Optimized Link State Routing Protocol.....	38
3.5.1 Visão geral.....	38
3.5.2 Vantagens do OLSR.....	40
3.5.3 Descoberta de hosts vizinhos (neighbor sensing).....	41
3.5.4 Link state routing.....	42
3.5.5 MultiPoint Relay (MPR).....	43
3.5.6 Cálculo de rotas.....	44
3.5.7 Acessos externos (HNA).....	46
3.6 Comentários Finais.....	47
4 MATERIAL E MÉTODOS	48
4.1 Escolha da implementação do protocolo.....	48
4.2 O OLSRD	49
4.3 Apresentação do ambiente de testes.....	51
4.4 Instalação e configuração do protocolo OLSR no GNU/Linux	53
4.5 Configuração de clientes GNU/Linux em modo Ad Hoc	59
4.6 Configuração de clientes MS/Windows em modo Ad Hoc	60
5 RESULTADOS E DISCUSSÃO.....	63
5.1 Cenário 1.....	63
5.2 Cenário 2.....	66
5.3 Cenário 3.....	69
5.4 Observação da movimentação dos pacotes de mensagens na rede.....	73
5.5 Análise e resultados.....	77
6 CONCLUSÃO.....	78

REFERÊNCIAS.....	80
APÊNDICE A - Projeto RMP/Cidade Digital.....	83
ANEXO A – Arquivo olsrd.conf Default.....	86

1 INTRODUÇÃO

Em tempos onde as conexões à *internet* se tornaram cada vez mais populares, sendo as mesmas disponibilizadas das mais variadas formas, o projeto e manutenção de grandes redes sem fio, se tornou um dos grandes problemas para administradores e gerentes de redes.

Uma rede de grande porte consome muitos recursos financeiros e, em geral, na busca de se conseguir conectividade estável e segura, grandes partes destes são gastos com *hardware* e *software* proprietário. Administradores e gerentes normalmente buscam soluções robustas e confiáveis, e isso é perfeitamente possível, a um custo baixo e que não necessite de dispositivos proprietários caros.

Nesse contexto, o GNU/Linux sempre surge como a solução mais aceita e utilizada, pois o mesmo com um *hardware* PC dá todo o poder e flexibilidade que se precisa para a maioria das soluções de problemas corporativos de pequeno, médio e grande porte.

Isso é especialmente verdade em projetos que envolvem grandes redes para cidades digitais, devido ao alto custo de equipamentos a serem utilizados na infraestrutura dessas redes, como pontos de acesso, roteadores, *gateways*, *proxies*, etc. O custo de manutenção e licenciamento cai consideravelmente utilizando soluções robustas de *software* livre como o caso de GNU/Linux.

Na busca de encontrar soluções que se adequem ao perfil dessas redes, no Brasil e no mundo existem diversas iniciativas, que estudam e testam tecnologias de redes para serem utilizadas em projetos de cidades digitais. Uma das alternativas e que tem ganhado cada dia mais adeptos são as redes em malha, também conhecida como *Wi-Mesh*, MANETs ou redes virais.

Essa tecnologia aplica às redes sem fio a mesma arquitetura de roteamento distribuído encontrada na *internet*, com vários roteadores sem fio

(repetidores) conectados a nós ou pontos de acesso inteligentes, fornecendo uma cobertura a uma área maior do que a possibilitada com um ponto de acesso independente (*hotspot*), utilizados em rede *Wi-Fi* convencionais.

1.1 Objetivos

Compreender a importância das redes de dados para o desenvolvimento da sociedade moderna e conhecer o potencial de alguns dos vários protocolos utilizados em redes *Wi-Mesh* é de fundamental importância para administradores de redes, gerentes de TI e principalmente para projetistas de redes MANs sem fio. Muito além do funcionamento e configuração dos protocolos responsáveis pela comunicação em malha, também é de grande valia conhecer e implementar outros serviços, que devem coexistir dentro da infraestrutura da rede, afim de promover um correto funcionamento desses equipamentos.

Especificamente é objeto de estudo deste trabalho, descrever uma tecnologia com a qual consiga-se montar roteadores *wireless* robustos e gerenciáveis a um custo baixo. Os convencionais *access points* com *software* embarcado disponíveis no mercado para rede *Wi-Mesh*, em geral são caros. Fazendo uso de *hardware* PC, um sistema operacional *GNU/Linux* com uma implementação do protocolo OLSR é possível reduzir consideravelmente este custo. A priori, pretende-se que este roteador *wireless*, seja utilizado no projeto RMP/Cidade Digital (APÊNDICE A).

1.2 Metodologia

Para a realização desse trabalho foram realizadas estudos teóricos através de revisões bibliográficas a cerca da disponibilidade e o funcionamento

das tecnologias utilizadas em projetos de cidades digitais. As revisões buscam apresentar uma visão mais ampla de tecnologias e topologias para redes *Wi-Mesh*. Por meio dessas revisões, chega-se a uma tecnologia específica que será colocada a prova em um ambiente de testes. Através da análise do comportamento da rede de testes, busca-se algumas conclusões sobre as vantagens do uso de tal tecnologia. Para as revisões bibliográficas realizadas foram utilizados livros e artigos técnicos a respeito do assunto, além da própria RFC3626 que padroniza o protocolo escolhido para os experimentos. Para os testes práticos foram utilizadas máquinas reais com *hardware* i386, rodando o Sistema Operacional GNU/Linux Debian 6.0 (6.0.2.1-i386) codinome “*squeeze*” e *kernel* 2.6.32, além do GNU/Linux Ubuntu Desktop 10.04 e o Microsoft Windows XP Professional. Para obtenção dos resultados, as máquinas reais foram monitoradas, havendo a captura de pacotes com analisadores de protocolos e posterior análise desses pacotes.

1.3 Estrutura do Texto

Com propósito de organização, o conteúdo deste trabalho encontra-se estruturado da seguinte forma: O Capítulo 2 aborda as cidades digitais, mostrando os benefícios e seu funcionamento, em seguida fazendo uma reflexão sobre a forma correta que as mesmas deveriam ser tratadas na sociedade. O Capítulo 3 trata das tecnologias que podem ser utilizadas em projetos de cidades digitais, suas classificações e padrões de mercado, seguindo com a exposição da topologia em malha, o seu funcionamento e referenciando alguns protocolos proativos e reativos para redes *Wi-Mesh*. O Capítulo ainda faz o detalhamento do protocolo proativo OLSR, fazendo uma revisão bibliográfica do seu funcionamento, detalhando os tipos de pacotes utilizados. O Capítulo 4 justifica a escolha de uma implementação do protocolo proativo OLSR, apresentando em

seguida um ambiente de testes onde o esta implementação foi instalada junto ao GNU/Linux, criando assim os roteadores *Wi-Mesh* para serem colocados em testes, experimentando o seu funcionamento. O Capítulo 5 detalha os resultados obtidos com os experimentos realizados dentro da ambiente de testes, fazendo uma análise desses resultados. Por fim no Capítulo 6 são apresentadas as conclusões do trabalho e perspectiva para outros trabalhos e experimentos com o protocolo OLSR. Ainda no APÊNDICE A têm-se uma descrição do projeto RMP/Cidade digital onde espera-se que o protocolo objeto de estudo do trabalho seja utilizado e no ANEXO A uma cópia default do arquivo de configuração da implementação do protocolo proativo OLSR utilizada nos teste do GNU/Linux.

2 CIDADES DIGITAIS

2.1 Comentários iniciais

A importância da *internet* para a sociedade da informação e para desenvolvimento de toda a nação é um fator muito estudado atualmente. São necessários diversas pesquisas e estudos a fim de detectar os benefícios que se espera gerar com a implantação de um serviço como os que deveriam ser disponibilizado nas cidades digitais.

Justamente devido à falta de planejamento, todo o processo ocorre na maioria das vezes por puro modismo – “meu vizinho tem, e eu quero também” - e muitos dos projetos não tem estudos aprimorados sobre o público-alvo, o conteúdo a ser disponibilizado e as tecnologias a serem implantadas.

A urgência de universalização do acesso à *internet* está por trás de muitos projetos públicos municipais de provisão de *internet* sem fio para a população. Mundo afora, governos locais encaram o desafio de instalar redes metropolitanas de banda larga sem fio nutrindo a esperança de inclusão digital, de desenvolvimento econômico e até de uma nova possibilidade de participação política de seus cidadãos e cidadãs.

2.2 Benefícios de uma cidade digital

Hoje é comum em grande parte dos municípios brasileiros, a implantação e disponibilização de acesso gratuito à *internet* para os cidadãos daquela cidade/município: são as chamadas cidades digitais. Porém quase nunca uma cidade digital quer dizer apenas acesso gratuito a *internet* para aos cidadãos. Mesmo que esse seja o primeiro pensamento que nos vêm em mente.

Quando se fala em cidade digital, pensa-se imediatamente que se trata de oferecer à população carente acesso à tecnologia. É isto? Também. Mas é muito mais. Ser uma cidade digital significa modernizar a gestão pública e oferecer novos serviços e facilidades para as pessoas, e significa principalmente levar aos seus habitantes uma nova perspectiva de cidadania. Os benefícios abrangem todas as áreas, da administração pública à educação, passando pela saúde e segurança, e estendendo-se à economia do município. (Guia das Cidades Digitais , 2010).

Rocha (2009) mostra perfeitamente que cidade digital, vai muito além de acesso gratuito a *internet*. A aplicação da tecnologia, quando bem utilizada na administração pública, traz grandes benefícios para todos. Os benefícios trazidos para os municípios que implantam projetos para se tornarem "Cidades Digitais" são inúmeros, principalmente quando se vai além da tecnologia.

Assim sendo as cidades têm a oportunidades de melhorar a gestão pública através da rede, não apenas proporcionando à população acesso à *internet* de qualidade e barato, mas principalmente elevando o nível dos serviços prestados pelas prefeituras nas áreas de educação, saúde, segurança, turismo, etc. Além de tudo isso, as Cidades Digitais criam oportunidades para fomentar o empreendedorismo e o desenvolvimento econômico dos municípios.

2.3 Como funcionam as cidades digitais

Projetos que envolvem cidades digitais, de uma maneira geral, misturam argumentos econômicos, além de debates sobre direitos humanos na chamada "Sociedade do Conhecimento" e até mesmo de estratégias de *marketing* político. O que ocorre é que na sustentação dessas iniciativas, é possível encontrar as mais diversas parcerias entre o governo, o mercado e a sociedade civil organizada.

Na verdade o que muito dos gestores enxergam é apenas o acesso

gratuito a *web*, acesso gratuito que nesse sentido quer dizer grátis apenas para o cidadão. Alguém precisa pagar a conta, e na maioria das vezes é a administração pública municipal é quem arca com todas as despesas de manutenção do *link* de serviço e a mão de obra qualificada para administração e gerência da rede. Em outros casos parcerias com ONGs, empresas privadas ou universidades que ajudam a custear os altos valores de uma infraestrutura dessas.

Para além do debate sobre a cidade sonhada, é sabido que a própria participação de governos locais na provisão de *Internet* sem fio é motivo de discórdia. Já são conhecidos os casos de cidades americanas que iniciaram e interromperam total ou parcialmente os seus projetos – Filadélfia, São Francisco, Chicago, entre outras. Os argumentos recorrentes incluem a possibilidade de inibir o desenvolvimento do mercado local de telecomunicações, o risco de investimento público em uma área fora da competência técnica da maioria das prefeituras, além da possível inviabilidade financeira do projeto em longo prazo; ou seja, a pergunta básica é: quem paga a conta de uma infraestrutura de rede tão grande e com tantos usuários? (JOSGRILBERG, 2009, p. 8)

Muito além do que se refere às questões técnicas, o sonho do espaço urbano coberto por uma “nuvem digital”, com cidadãos e cidadãs tirando proveito das múltiplas possibilidades da comunicação móvel, é freado pelo custo das mídias (*laptops*, PDAs, celulares, etc.) e um alto custo de implementação e manutenção, devido a grandes restrições dos equipamentos portáteis atuais na recepção de sinais, em geral limitados à frequência de 2.4 Ghz.

2.4 Tecnologias aplicadas às cidades digitais

Em geral para projetos desse porte existem duas saídas: A primeira é investir em equipamentos apropriados, projetados para redes WMANs de grande dimensões como os do padrão WiMAX. Estes em geral usam *hardware*

proprietário com *software* embarcado, que necessita de mão de obra especializada e que custa muito caro. A segunda, mais utilizada que em geral é a utilização de soluções mais baratas, que em geral são soluções convencionais utilizadas em redes WLANs e não recomendadas para utilização em redes com distâncias e dimensões maiores.

A segunda alternativa é muito mais aplicada, pelo fato do alto custo da primeira. Tendo em vista essa situação, algumas tecnologias surgiram, afim de estudar e implementar protocolos que diminuam os problemas causados pelas ineficiências dos já existentes. Utilizar protocolos alternativos junto com soluções *free software* robustas é uma alternativa viável para projetar soluções robustas e eficientes e que tenham um baixo custo de implementação e manutenção.

2.5 Comentários Finais

Existem no Brasil e no mundo diversas iniciativas, que estudam e testam tecnologias de redes para serem utilizadas em projetos de cidades digitais¹. Uma dessas alternativas são redes em malha, também conhecida como *Wi-Mesh*, MANETs ou Redes Virais e que são detalhadas no Capítulo 3. Isso porque as redes em *Wi-Mesh* aplicam às redes sem fio a mesma arquitetura de roteamento distribuído encontrada na *internet*.

Redes em malha funcionam com vários roteadores sem fio (repetidores) conectados a nós ou pontos de acesso (*gateways*) inteligentes. A rede em malha fornece cobertura a uma área maior do que a possibilitada com um ponto de acesso independente, ou *hotspot*, utilizados em rede *Wi-Fi* convencionais.

Conforme observa-se no Capítulo 3, redes em malha são projetadas para continuar funcionando com eficiência mesmo quando alguns nós encontrarem

¹ Guia das Cidades Digitais, 2010.

problemas na transmissão. O roteamento em redes *mesh* não param, pois sempre os nós (que também são responsáveis pelo roteamento) procuram um novo caminho para enviarem a mensagem ao destinatário.

Observa-se também que recentes experimentos em redes *mesh*, funcionando com soluções que envolvem o uso do *software* livre, têm mostrado a mesma como uma solução ideal para inundar áreas urbanas com acesso sem fio de alto desempenho e baixo custo. Este tipo de redes já são utilizadas em cidades de outros países para fornecer acesso universal à *internet*.

3 REDES WIRELESS

É quase impossível encontrar uma pessoa que nunca tenha usufruído dos benefícios de uma rede sem fio. O uso de controles remotos, uso do celular, telefones sem fio, são exemplos do quanto as redes *wireless* estão presentes no nosso dia a dia.

Analisando a Figura 1, podemos notar que mesmo aquelas pessoas que não são usuárias de computadores estão em constante uso de meios de comunicação que utilizam espectros de frequência, seja estes por ondas de rádio, infravermelho ou satélite.

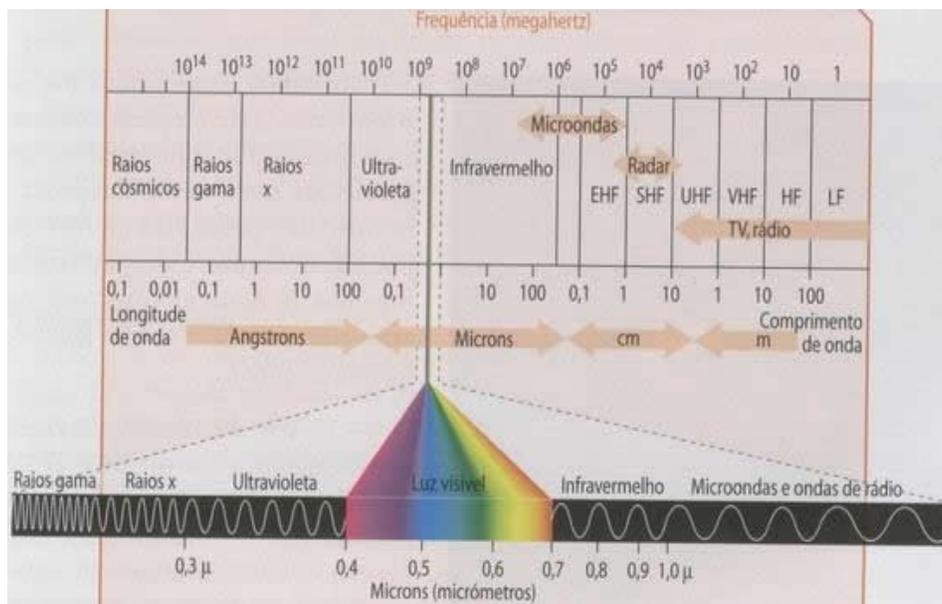


Figura 1 Espectro de frequências

Fonte: (Stallings, 2000)

Segundo Sousa (2004), as frequências eletromagnéticas são divididas em faixas. Como pode ser visto no Tabela 1, estas faixas são conhecidas popularmente como baixa, média ou alta frequência. Cada tipo de transmissão

(rádio, TV, satélite, microondas, entre outras) possui a sua frequência específica.

É nessa frequência que cada tipo de equipamento está projetado a trabalhar, pois devido as próprias características da transmissão, cada tipo de serviço (dados, voz, imagem, etc) utiliza-se de uma frequência específica.

Tabela 1: Classificação de frequências

Fonte: (Sousa, 2004, p. 26 e 27)

Classificação	Nome Popular	frequências	Utilização
<i>ELF – Extremely Low Frequency</i>	ondas longas	300Hz a 10000Hz	Sonares
<i>VLF – Very Low Frequency</i>	ondas longas	10Khz a 0Khz	Sonares
<i>LF – Low Frequency</i>	ondas longas	30Khz a 300Khz	Navegação aérea, rádio difusão
<i>MF – Medium Frequency</i>	ondas médias	300Khz a 3000Khz	Navegação aérea, rádio difusão
<i>HF – High Frequency</i>	ondas curtas	3Mhz a 30Mhz	Rádio difusão, comunicação marítima
<i>VHF – Very High Frequency</i>	-	30Mhz a 300Mhz	TV, FM, radio amadores
<i>UHF – Ultra High Frequency</i>	microondas	300Mhz a 3000Mhz	Comunicações públicas e privadas
<i>SHF – Super High Frequency</i>	microondas	3Ghz a 30Ghz	Comunicações públicas e privadas
<i>EHF – Extremely High Frequency</i>	microondas	30Ghz a 300Ghz	Comunicações públicas e privadas
Região Experimental	-	300Ghz a 1000Ghz	Comunicações públicas e privadas

Como podemos observar na Tabela 1, sejam utilizando frequências mais altas ou mais baixas, as tecnologias de rede sem fio estão presentes no cotidiano das pessoas há muito tempo.

Como o foco deste trabalho são as redes de computadores, todas as vezes que nos referimos a redes sem fio, mais formalmente tem-se redes de

computadores sem fio.

Conforme Ross (2003), uma rede local sem fio exige, de alguma maneira, um conjunto diferente de componentes de *hardware* de uma rede com fio tradicional. Evidentemente, a maior diferença é a ausência de fios entre o servidor de rede, os computadores de clientes *wireless* e outros dispositivos que compõem a rede, mas essa não é a única diferença. Uma rede *wireless* também deve usar transmissores e receptores de rádio como *interface* com a rede e entre as porções com fio e sem fio rede.

Redes de computadores que utilizam transmissão sem fio utilizam um espectro de frequências que em geral é conhecido como ondas de rádio.

O espectro de frequências de rádio inclui as faixas de rádio AM, rádio FM, TV aberta e telefonia móvel celular. Nesse espectro as ondas passam facilmente por obstáculos, como paredes, e podem alcançar longas distâncias, especialmente quando as ondas são refratadas na ionosfera. A transmissão via rádio utiliza antenas onidirecionais, ou seja, os sinais são transmitidos em todas as direções. Nesse caso, não há necessidade de que o transmissor e o receptor estejam alinhados ponto-a-ponto, como ocorre nas transmissões que utilizam frequências mais altas. (MAIA, 2009, p. 54)

Apesar da largura de banda do espectro de frequências de ondas de rádio ser muito grande, apenas uma pequena banda passante é utilizada para fins de transmissão em redes de computadores. Redes locais sem fio que seguem o padrão IEEE 802.11 funcionam em faixas de frequência ISM². O mais comum é que elas utilizem frequências de 2.4 Ghz e 5 Ghz, que fazem parte do espectro de frequência ISM.

Segundo Brodsky (1997), as bandas ISM oferecem largura de banda

2 Algumas faixas de frequências chamadas ISM (Industrial, Scientific, Medical), podem ser utilizadas livremente, desde que a transmissão seja feita em baixa potência para evitar interferências. Redes locais sem fio que seguem o padrão IEEE802.11e telefones sem fio funcionam em faixas ISM. (MAIA, 2009, p. 54)

desde serviços T1 tradicional (384 Kbps) até a velocidade máxima (10 Mbps).

A utilização de um espectro de frequências é regulamentado pelo governo de cada país, e no Brasil a ANATEL é o órgão autorizado pelo governo a regulamentar a utilização do mesmo. Faixas de frequências, chamadas de ISM (*Industrial, Scientific, Medical*), podem ser utilizadas livremente, desde que a transmissão seja feita em baixa potência para evitar interferências e são regulamentadas internacionalmente pelo ITU-T.

3.1 Classificação das redes *wireless*

Todas as redes sem fio baseadas em ondas de rádio estão suscetíveis à interferências, baixa banda de comunicação, limitação de processamento devido ao pequeno tamanho e restrição quanto ao uso de energia. Assim sendo podemos classificá-las em dois modos de operação: infraestrutura e sem infraestrutura ou ponto-a-ponto.

3.1.1 Redes *wireless* com infraestrutura

A comunicação em modo infraestrutura utiliza-se um dispositivo centralizador equivalente a um *hub/switch* das redes cabeadas. Nesse caso a rede sem fio é conectada à rede física por meio de controlador *gateway*. O mesmo recebe o nome de ponto de acesso (*access point*).

O modo infraestrutura utiliza o conceito de BSA (*Basic Service Area*) que é a área na qual os nós da rede podem se comunicar e também o BSS (*Basic Service Set*) que são as áreas de alcance que cada ponto de acesso pode abranger. O AP como é conhecido o ponto de acesso também é responsável por controlar o uso das ondas de rádio entre as estações.

Muitas são as redes que utilizam este modo de operação, como podemos citar o exemplo de redes de acesso público (*hotspot*), muito populares em locais como aeroportos, hotéis e restaurantes. Este é um modo de operação que se tornou eficiente, barato e de fácil configuração e por isso mesmo muito popular no mundo todo e principalmente no Brasil.

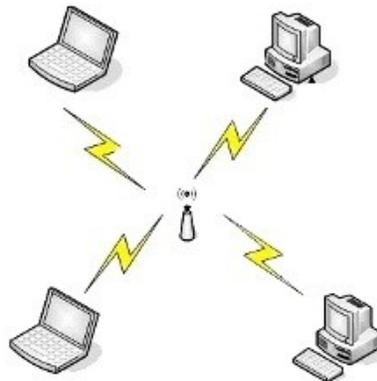


Figura 2 Rede sem fio em modo infraestrutura
Fonte: (JARDIM, 2007, p. 45)

3.1.2 Redes *wireless* sem infraestrutura

Este tipo de modo de operação popularmente conhecido do jargão da informática como *Ad Hoc*, é uma rede ponto-a-ponto onde não há a necessidade de uma rede estruturada, pois todos os nós da rede formam toda a dinâmica de comunicação da rede sem a dependência de um ponto central. Nesse modo de operação um dispositivo conectado a rede se comunica diretamente com o outro, não havendo então uma topologia determinada e nenhum controle centralizado. Também não há a necessidade de *gateways* para rotear os pacotes, como os utilizados em redes infraestrutura. Em redes *Ad Hoc* cada dispositivo é um roteador em potencial, ou seja, eles mesmos são responsáveis por transferir pacotes para outros nós da rede.

Devido a limitações no protocolo, uma rede operando em modo *Ad Hoc*,

seria recomendado apenas para pequenas redes domésticas. Porém, com o surgimento de novos padrões como o IEEE 802.11s para redes sem fio em malha no modo *Ad Hoc*, essa premissa já não é mais verdadeira.



Figura 3 Rede sem fio em modo *Ad Hoc* (sem infraestrutura)
Fonte: (JARDIM, 2007, p. 44)

3.2 Redes *wireless* - Classificação por área de abrangência

Assim como em redes cabeadas com fios, as redes sem fios recebem uma classificação quanto à sua área de abrangência. É uma classificação baseada no alcance das ondas e dos tipos de dados trafegados. Ainda é comum encontrar um maior ou menor detalhamento quando a estas classificações, porém no geral elas são classificadas em 3 tipos: WWANs, WMANs e WLANs.

3.2.1 *Wireless wide area network (Wireless WANs)*

As WWANs, como são conhecidas, são baseadas no modo infraestrutura e em geral as estações base, estão espalhadas por uma área muito grande dando conexão a todos os nós da rede. Um exemplo desse tipo de WWANs são as redes

para dispositivos móveis como a rede de telefonia celular.

3.2.2 *Wireless metropolitan area network (Wireless MANs)*

As WMANs são redes metropolitanas sem fio. Este tipo de rede geralmente é criada para realizarem a ligação entre pontos fixos, como em edifícios e escritórios comerciais e industriais. Podem ser utilizadas até mesmo para ligarem pontos em cidades diferentes.

3.2.3 *Wireless local area network (Wireless LANs)*

Redes WLANs são projetadas em pequenos locais, em geral salas com poucos metros de algumas centenas de metros. Em geral utilizam o modo de operação em infraestrutura, mas não é raro encontrar WLANs operando em modo *Ad Hoc* especialmente em pequenas redes de apenas algumas dezenas de nós.

3.3 Padrões de redes *wireless*

Com a popularização das redes de sem fio, várias especificações e padrões de mercado surgiram afim de organizar a comunicação dos diversos dispositivos que foram testados e produzidos. Podemos citar os esforços de empresas como o consórcio *Wi-Fi* e de organizações como a IEEE, que tem buscado garantir a padronização e a especificação dessas tecnologias.

Justamente como resultados desses esforços algumas dessas especificações estão de fato se tornando padrões de mercado para redes WMANs e WLANs.

Em geral redes WLANs utilizam as especificações do IEEE 802.11 e

utilizam o equipamentos padrão *Wi-Fi*.

Especificações e padrões não são importantes para usuários comuns. Estes simplesmente querem usar a tecnologia quando precisarem sem ter que conhecer a fundo como funcionam estes equipamentos. Mas para profissionais envolvidos em projetos de redes o conhecimentos desses padrões e pré requisito para projetos de redes *wireless* bem sucedidos.

Até certo ponto, é viável lidar com suas redes *wireless* (redes sem fio) como um conjunto de caixas pretas que você poderá ativar e usar, sem precisar conhecer a fundo o seu funcionamento. É dessa maneira que a maioria das pessoas se relaciona com a tecnologia com a qual convivem – você não precisa se preocupar com a especificação 802.11b para conectar seu computador *laptop* a uma rede. Em um mundo ideal (ah!), ela funcionaria assim que você ligasse o computador. (ROSS, 2003, p. 1)

Em se tratando de mercado atual, têm-se duas principais tecnologias *wireless* (padrões homologados pelo órgão Internacional – IEEE) destinadas à transmissão de dados entre dispositivos computacionais, que são a *Wi-Fi* e *WiMAX*.

3.3.1 IEEE 802.11 - *Wi-Fi*

Desde os primórdios da popularização das redes de computadores, sempre houveram esforços para padronizar as tecnologias. Com o advento do exploração das redes sem fio para as redes de computadores pessoais isso não demorou a acontecer. O *Wi-Fi* é um exemplo prático dessa rápida padronização.

Segundo informações da própria *Wi-Fi Alliance (2010)*, em 1999, vários empresas líderes se uniram para formar uma organização global sem fins lucrativos com o objetivo de orientar a adoção de um padrão único em todo o mundo para a alta velocidade em rede local sem fio. Surge ai a *Wi-Fi Alliance*,

que hoje já passam de 350 companhias que estão presentes em dezenas de países. Hoje, em todos os continentes, um em cada dez pessoas ao redor do mundo usam *Wi-Fi* em casa, no trabalho, de inúmeras maneiras.



Figura 4 Logo que identifica produtos da *Wi-Fi Alliance* em todo o mundo
Fonte: <http://www.Wi-Fi.org/>

Wi-Fi, que vem do termo *Wireless Fidelity* (fidelidade sem fios), tornou-se a tecnologia de mais rápida adoção do mundo *wireless* dos últimos anos e já se encontra em muitos dispositivos computacionais, como por exemplo os atuais PDAs e *notebooks*. Sempre que em um produto encontramos o logo *Wi-Fi* mostrado na Figura 4, é porque este é compatível com o padrão. Ainda segundo Jardim (2007), *Wi-Fi* é o nome comercial usado para designar um conjunto de padrões de rede *wireless* desenvolvida pelo comitê 802.11 do IEEE onde o mesmo é dividido em três principais padrões: 802.11b, 802.11a e 802.11g.

3.3.2 IEEE 802.16 – *WiMAX*

WiMAX, é um acrônimo para *Worldwide Interoperability for Microwave Access* que quer dizer Interoperabilidade Mundial para Acesso por Microondas. De acordo com Figueiredo (2001), *WiMAX* é uma tecnologia de banda larga sem fio, que atua nas áreas de redes comunitárias e provimento de acesso de última milha. É uma rede WMAN que possui ampla uso e aceitação no mercado internacional, já que projeções futuras a veem como um padrão que deve atuar como alternativa a tecnologias como cabo e DSL.

Embora atualmente esta tecnologia não esteja tão difundida no mercado internacional, ela promete ser um forte concorrente pela já consolidação do *Wi-Fi*, uma vez que possui algumas características que se sobressaem perante esta última, que são: taxa de transferência de até 75 Mbps, contra os 54 Mbps da 802.11g; raio de alcance de quase 50 Km em área livre e em área de densidade populacional de 8 a 10 km e capacidade de atender milhares de usuários com uma única estação base, contra apenas centenas de máquinas pela *Wi-Fi*. (JARDIM, 2007, p. 36)

A maior desvantagem em relação do *WiMAX* perante o *Wi-Fi* é que a o mesmo possui um valor de implementação dezenas de vezes maior, e com isso a *Wi-Fi* continua fortíssima no mercado.

3.3.3 IEEE 802.11s – *Wi-Mesh*

Redes *Wi-Mesh*, utilizam especificações IEEE 802.11s, e são tecnologias que aplicam às redes sem fio a mesma arquitetura de roteamento distribuído encontrada na *internet*, com vários roteadores sem fio (repetidores) conectados a nós ou pontos de acesso inteligentes, fornecendo uma cobertura a uma área maior do que a possibilitada com um ponto de acesso independente (*hotspot*), utilizados em rede *Wi-Fi* convencionais. Redes *Wi-Mesh* também são conhecidas como rede *Wi-Fi* metropolitana. As mesmas serão mais detalhadas na seção a seguir.

3.4 Redes em Malha – *Wi-Mesh*

Redes de computadores, possuem muitas classificações e definições, O foco desse estudo são as redes sem fio em malha, também conhecida como *Wi-Mesh* ou redes virais, operando em modo *Ad Hoc* (sem infraestrutura) utilizando o padrão IEEE 802.11, mais especificamente o IEEE 802.11s. Redes *Ad Hoc*

também recebem a nomenclatura MANET ou *Mobile Ad Hoc Network*.

3.4.1 Protocolos de roteamento em redes *mesh*

Redes *Wi-Mesh* necessitam que seus nós estabeleçam métodos de encontrar rotas para todos os outros nós ativos na rede. Devido a esta particularidade, cada nó é na verdade um roteador em potencial. Os responsáveis por tais tarefas são na verdade os protocolos, que são responsáveis por encontrar e gerenciar as rotas. Esses protocolos são classificados em dois grupos: proativo e reativo.

3.4.2 Protocolos proativos

No caso de protocolos proativos, cada nó mantém uma tabela e nela constam as rotas para todos os nós da rede, que são atualizadas em determinado intervalo de tempo. Esse tipo de protocolo é interessante pois como o mesmo mantém as rotas atualizadas nas tabelas, sempre que necessário a rota estaria disponível, sem a necessidade de atualização na hora do envio. Porém numa rede mais densa com uma certa quantidade de nós, a tarefa de atualização das rotas poderia gerar um auto *overhead* na rede por causa das mensagens de *broadcast* enviadas de cada uns dos outros nós da rede.

Em geral em locais de grande mobilidade, protocolos que utilizam este tipo de abordagem, tornam a rede de certa forma desvantajosa, já que a atualização das rotas prejudicaria o desempenho da rede devido ao aumento considerável do tráfego. Nesse tipo de rede podemos destacar dois protocolos: OLSR e DSDV. O OLSR apresenta especial interesse para este trabalho e será apresentado com mais detalhes na seção 3.5.

3.4.3 DSDV - *Destination-sequenced distance vector routing*

O DSDV é um protocolo que atua nas camadas 2 e 3 do modelo de referência OSI. O protocolo mantém tabelas de roteamento com indicações de número de *hops* para cada destino. Todas as entradas das tabelas de roteamento para cada destino são numeradas em sequência e as informações de roteamento são transmitidas de forma incremental.

3.4.4 Protocolos reativos

Protocolos reativos também são conhecidos como *on-demand* e funcionam descobrindo rotas apenas quando as mesmas são necessárias, ou seja, só são atualizadas no momento do envio das mensagens (isto se a mesma for necessária). No processo de descobrimento de rotas (*Route Discovery*) dos protocolos reativos, a vantagem é que o protocolo gera uma redução considerável no fluxo de mensagens da rede diminuindo o *overhead* da mesma. Porém esse benefício só é vantajoso em casos de redes de baixa mobilidade. São exemplos de protocolos reativos: DSR e AODV.

3.4.5 DSR - *Dynamic source routing protocol*

O DSR foi padronizado com o lançamento da RFC 4728 pelo IETF³. Conforme Johnson e Malts (2011), o mesmo funciona de forma semelhante ao

3 RFC é um acrônimo para Request for Comments (ou seja, “Solicitação de Comentários”). Uma RFC define uma série de padrões e modos de operação de protocolos e serviços de rede, e podem ser escritos e submetidos por qualquer indivíduo. Uma vez finalizada, uma RFC deve ser submetida ao IETF (*Internet Engineering Task Force*), que analisa o documento e o aceita ou não para publicação oficial. Se aprovada, a RFC recebe um número único e é publicada. Uma vez publicada, uma RFC não pode mais ser alterada. Alterações a uma RFC publicada são feitas por meio de novas RFCs.

AODV, porém possui um mecanismo de descoberta de rotas que difere na forma como a informação de rota é tratada pelas estações. A Figura 5 a seguir mostra a descoberta de rotas a partir da fonte S para o destino D.

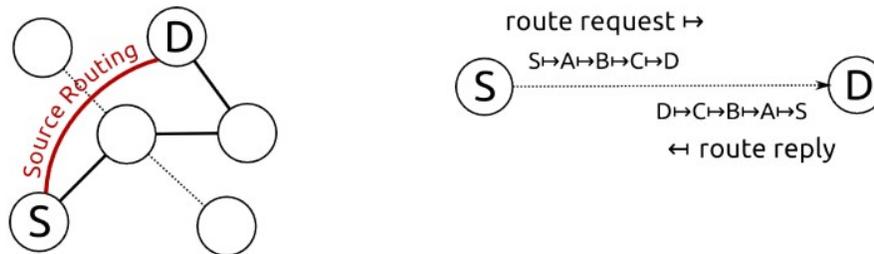


Figura 5 Descoberta de rotas a partir da fonte S para o destino D
 Fonte: <http://www.des-testbed.net/content/dynamic-source-routing-dsr>

Na Figura 5, quando não há entrada em sua tabela para uma determinada rota, o nó de origem envia um pacote de RREQ para o seu nó vizinho que verifica em suas tabelas se há alguma entrada para a rota pretendida, caso o nó vizinho não tenha a rota, a mesma é atualizada para a estação origem e adiciona no RREQ o seu IP reenviado para a próxima estação. À medida que o RREQ vai sendo transmitido a todos os nós, a rota fica preestabelecida de forma a construir o caminho até o destino. Caso um nó que recebe o RREQ tenha a rota até o destino, ele pega o caminho criado pelo RREQ com a rota para o destino e coloca-os em um RREP que é então transmitido em *unicast* para a origem. Se o nó de destino receber a mensagem de RREQ, ele mesmo gera o RREP com o caminho gerado pelo RREQ e reenvia em *unicast* para a origem.

3.4.6 AODV – *Ad Hoc on-demand distance vector routing*

Conforme Royer e Perkins (2011), o AODV é um protocolo de roteamento muito popular para MANETs. É um protocolo reativo, de rotas criadas sob demanda, e apenas as rotas ativas são mantidas. Isto reduz a

sobrecarga de roteamento, mas introduz algumas latências, devido à configuração *on-demand* das mensagens. AODV foi padronizada no IETF na RFC 3561. Há diversas implementações disponíveis, por exemplo, AODV-UU da Universidade de *Uppsala*, uma adaptação do AODV que foi proposto para redes sem fios da malha.

3.5 OLSR: *Optimized Link State Routing Protocol*

O *Optimized Link State Routing Protocol* (OLSR) é um protocolo que surge como alternativa para ser utilizado em projetos de cidades digitais. Este protocolo foi desenvolvido para ser utilizado principalmente em grandes e densas redes móveis do tipo *Ad Hoc*, também conhecidas como MANET's (*Mobile Ad Hoc Network*).

Em geral, projetos de grandes redes sem fio, necessitam de um sistema de roteamento mais eficiente do que os tradicionais métodos utilizados em *hotspots* convencionais encontrados em redes do tipo *Wi-Fi* (IEEE 802.11a, IEEE 802.11b e IEEE 802.11g).

3.5.1 Visão geral

Conforme relatam *Zhang, Luo e Hu (2007)*, o OLSR é um protocolo de roteamento para redes sem fio *Ad Hoc* proativas, desenvolvido pelo INRIA e padronizado pelo IETF através RFC 3626 como protocolo experimental. *Agha, Lassous e Pujolle (2006)* complementam dizendo que o OLSR é um promissor algoritmo selecionado pelo IETF pelo sua capacidade de roteamento em redes móveis em modo *Ad Hoc* (MANETs).

Sua finalidade é calcular e manter rotas de todos os nós em uma rede sem fio utilizando uma topologia em malha. O OLSR consegue isso, executando

em cada nó um *loop*, descobrindo caminhos para qualquer outro nó da rede.

Portanto, por convergência, cada nó preenche uma tabela de roteamento que indica a ele, como chegar a qualquer outro nó de destino.

O OLSR é proativo, o que quer dizer que o mesmo troca informações com outros nós da rede regularmente, atualizando todas as tabelas de roteamento, sempre que nós novos são ativos na rede como quando outros saem da rede. Uma vez que muitas mensagens de *broadcast* são transmitidas a toda a rede, o protocolo necessita um poderoso e eficaz mecanismo de controle de inundações.

Em redes que utilizam este tipo de protocolo, é aplicado um conceito conhecido como *MultiPoint Relay* (MPR), que é uma técnica para controle de inundações nos nós, e é isso que o diferencia de outros protocolos proativos.

Normalmente em redes *Ad Hoc* quando um nó recebe informações de atualização na rede, ele envia uma mensagem de *broadcast*, retransmitindo informações na rede aos seus nós vizinhos (mecanismo conhecido como inundações). Por ser um protocolo proativo (sempre trocando informações), isso é realizado várias vezes, assim então um nó recebe os mesmos pacotes repetidas vezes desnecessariamente. Tal tarefa gera um alto *overhead* na rede.

A diferença do OLSR de outros protocolos proativos é justamente o MPRs. O objetivo dos MPRs é minimizar esse problema através da seleção de nós que irão fazer a inundações. MPRs serão detalhadas na seção 3.5.5.

O OLSR é o mais indicado para trabalhar em grandes redes móveis do tipo MANETs, porque o mesmo é otimizado para isso. Como principais características podemos destacar:

- Amplamente utilizado em MANETs;
- Faz uso de *Link State Routing*;
- Utiliza *MultiPoint Relay* para otimizar o *Link State Routing*;
- É proativo;

- É hierárquico (nós possuem diferenças com relação aos outros);
- Roda sobre UDP na porta 698.

3.5.2 Vantagens do OLSR

Como dito anteriormente, o OLSR é um protocolo proativo, por isso tem no algoritmo de roteamento a grande vantagem. O mesmo possibilita a disponibilização de rotas imediatamente assim que as mesmas sofrem modificações na rede. O protocolo foi projetado para aperfeiçoar a rede realizando as mudanças topológicas de uma forma mais eficiente.

Sendo assim o mesmo é adequado para redes de grande densidade, pois a otimização imposta pelos MPRs funciona bem para redes neste contexto. Quanto maior e mais densa a rede, mais otimização das rotas pode ser alcançada.

O OLSR foi projetado para trabalhar de uma forma completamente distribuída sem dependência de uma entidade central, ou seja sem depender de um dispositivo para efetuar o controle da rede como os utilizados em redes IEEE 802.11a/b/g no modo infraestrutura.

O protocolo não precisa de transmissão de mensagem de controles confiáveis com TCP. Ele faz toda a comunicação usando a porta UDP 698 para transmissão de mensagens não confiáveis sendo as mesmas enviadas periodicamente. As perdas de algumas mensagens não geram grandes problemas para a rede. Isso é outra vantagem que o protocolo oferece.

O OLSR também não exige a entrega das mensagens em sequência, uma vez que cada mensagem contém um número sequencial e assim o destinatário da mensagem pode controlar o recebimento das mensagens, a sequência dos pacotes, e caso necessário solicitar a retransmissão de algum pacote perdido.

É importante ressaltar que o OLSR não faz qualquer alteração sobre a pilha de protocolos TCP/IP, pois o mesmo interage apenas com tabelas de

gerenciamento na camada 2 da modelo de referência *internet*. Redes com OLSR suportam endereços IPs como identificadores únicos para cada nó. A utilização de múltiplas *interfaces* também é suportada, porém um IP deve ser escolhido como sendo o principal para o roteamento, além de ter suporte para *IPv4* e *IPv6*.

3.5.3 Descoberta de *hosts* vizinhos (*neighbor sensing*)

Os nós numa rede *Ad Hoc com OLSR*, devem detectar os nós vizinhos com os quais tem uma relação direta e bidirecional. As incertezas sobre a propagação do sinal de rádio podem fazer com que algumas comunicações se restrinjam a ligações unidirecionais, porém, todas as ligações devem ser verificadas em ambos os sentidos, para que seja considerado um *link* válido.

Para conseguir isso, cada nó periodicamente envia uma mensagem conhecida como *HELLO*. O formato do pacote de uma mensagem HELLO pode ser observado na Figura 6 a seguir. Mensagens *HELLO* contém as informações sobre seus vizinhos e o seu estado e são transmitidas em modo de *broadcast*.

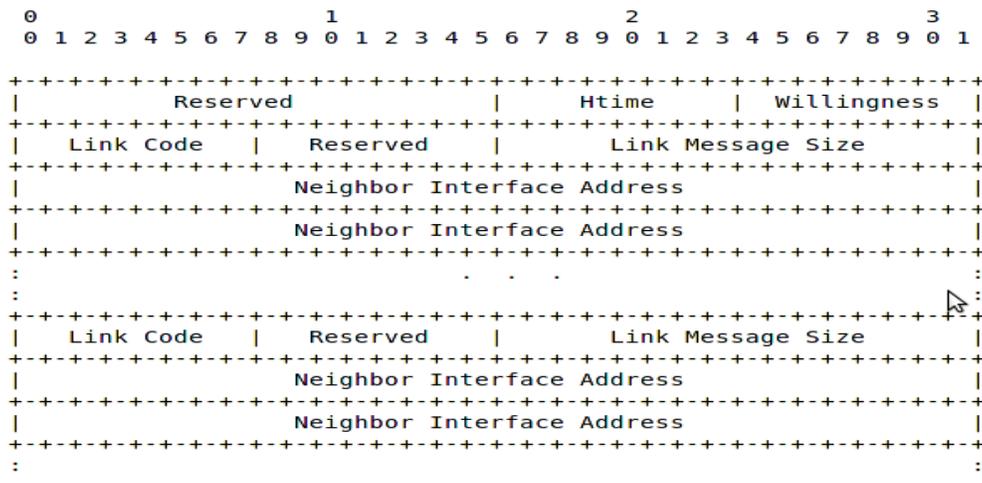


Figura 6 Formato de um pacote de mensagens HELLO

Fonte: (CLAUSEN e JACQUET, 2003, p. 43)

As mensagens *HELLO* enviadas por um nó da rede são recebidos por todos os nós vizinhos a um *hop* de distância, mas não são retransmitidas por estes. Conforme *Clausen et al. (2007)*, uma mensagem *HELLO* contém:

- A lista de endereços dos vizinhos para os quais existe uma ligação bidirecional válida;
- A lista de endereços dos vizinhos que são ouvidos por este nó, mas que o *link* ainda não está validado como bidirecional;
 - se um nó tem o seu próprio endereço em uma mensagem *HELLO*, considera o *link* para o nó remetente como bidirecional.

3.5.4 *Link state routing*

Os protocolos proativos herdam a estabilidade do algoritmo e é conhecido como protocolo de estado de ligação. Devido à sua natureza dinâmica, possui a vantagem de ter as rotas disponíveis imediatamente, quando necessário. Num protocolo de estado de ligação, todas as ligações com os nós vizinhos são declarados e são inundadas em toda a rede.

Já o protocolo OLSR é uma otimização de um protocolo de estado de ligação para redes móveis *Ad Hoc*, que é muito mais eficiente em redes MANETs.

Segundo *Clausen e Jacquet (2003)*, o OLSR reduz o tamanho dos pacotes de controle, e então, em vez de todas as ligações, ele declara apenas um subconjunto das relações com seus vizinhos. Como consequência, ele minimiza as inundações, fazendo o controle de tráfego utilizando apenas os nós selecionados, chamados MPRs, para difundir suas mensagens nas redes. Apenas MPRs de um nó retransmitem suas mensagens de *broadcast*. Esta técnica reduz significativamente o número de retransmissões de uma inundação ou

procedimentos de transmissão.

3.5.5 *MultiPoint Relay (MPR)*

O principal objetivo dos MPRs é a minimização do *overhead* em redes *mesh*, causado pela inundação de mensagens *HELLO*, vindas de todos os nós ativos. No OLSR somente os nós selecionados como MPRs são responsáveis pelo encaminhamento e controle de tráfego destinados a toda a rede (*broadcast*), evitando assim a inundação de informações redundantes. Cada nó da rede possui seus MPRs que são responsáveis pela retransmissão dos pacotes.

Para a seleção do MPRs, cada nó da rede *Wi-Mesh* com OLSR seleciona um conjunto de nós simétricos a um salto de distância. É utilizada a premissa que um nó deve alcançar todos os nós de segunda ordem através do menor número de MPRs possível, sendo que através dos MPRs um nó de origem deve chegar a qualquer nó a dois *hops* de distância. Assim os MPRs tornam o uso da rede muito mais eficiente, pois evita o envio de mensagens desnecessárias, controlando e diminuindo o *overhead*.

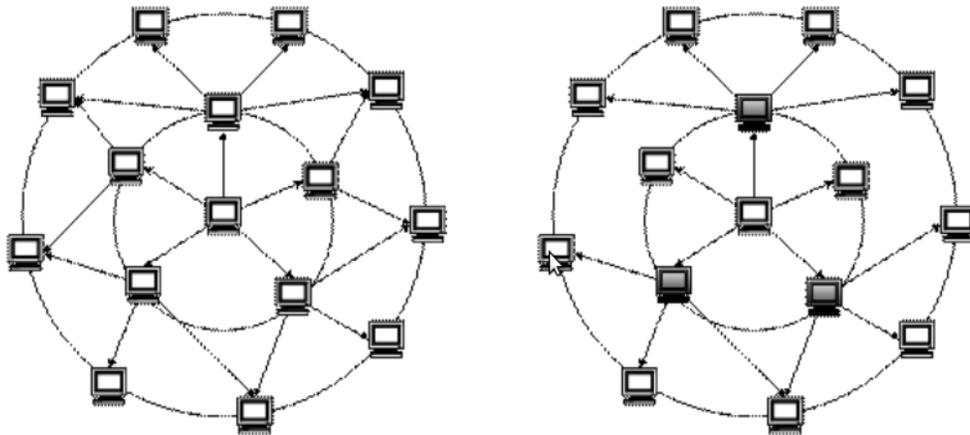


Figura 7 Inundação Normal e Inundação com MPR
 Fonte: http://www.gta.ufrj.br/grad/06_2/felipe/OLSR.htm

Conforme podemos verificar na Figura 7, na primeira situação, é realizada a inundação sem o uso do MPRs, e todos os nós da rede enviam e retransmitem mensagem de *broadcast* na rede, realizando uma inundação de mensagens repetidas, desnecessárias e aumentando consideravelmente o *overhead*. Na segunda, o processo de inundação já utiliza as MPRs e somente estações específicas (em negrito) retransmitem pacotes de inundação, aumentando a eficiência do protocolo⁴.

Nós vizinhos do nó N que não pertencem ao mesmo MPRs, recebem e processam mensagens de *broadcast* mas não retransmitem mensagem de *broadcast* enviadas pelo nó N. Cada nó escolhe seu vizinho para formação de sua MPRs efetuando 1 *hop* simétrico. Isso é selecionado de forma que haja cobertura no rádio *enlace* de todos os nós simétricos a 2 *hops* de distância.

Cada nó de MPR(N) mantém informações sobre o conjunto de nós vizinhos pertencentes a MPR(N). O conjunto de nós de um MPRs é conhecido como "*Multipoint Relay Selector Set*" de um nó. Um nó descobre estas informações a partir de informações periódicas recebidas de seus vizinhos. Uma mensagem de *broadcast* destinada a ser difundida na rede toda, provenientes de qualquer das MPR do nó N assume ser retransmitida para o nó N, caso N ainda não tenha recebido a mensagem. Estes conjuntos podem mudar ao longo do tempo (ou seja, quando um nó escolher outro MPR) e é indicado pelo seu nó seletor sua mensagem *HELLO*.

Para o encaminhamento do tráfego utiliza-se a premissa de que um nó só irá retransmitir um pacote OLSR se tiver sido escolhido como MPR pelo ultimo nó que retransmitiu o pacote e se esse pacote tiver TTL maior que 0 (zero).

3.5.6 Cálculo de rotas

⁴ Demonstration of MPR flooding. Disponível em: <<http://hipercom.inria.fr/olsr/mpr-flooding.html>>.

Cada nó da rede *Wi-Mesh* com OLSR mantém a informação sobre a topologia da rede. Esta informação é adquirida a partir de mensagens do tipo *TC* e é usado para o cálculos da tabela de roteamento. O formato de um pacote de mensagens TC pode ser observado na Figura 8 a seguir.

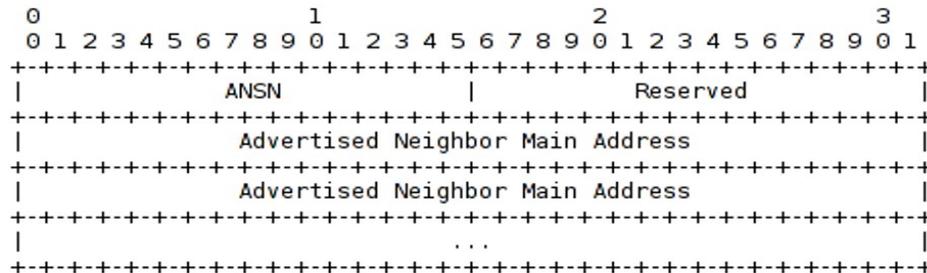


Figura 8 Formato de um pacote de mensagens TC
 Fonte: (CLAUSEN e JACQUET, 2003, p. 43)

Um nó mantém uma tabela de roteamento que permite encontrar um caminho destinados a outros nós na rede. Está tabela de roteamento é montada com base nas informações contidas na *Local Link Information Base*. Segundo Clausen e Jacquet (2003), *Local Link Information Base* é um base local que armazena informações sobre os caminhos para os nós vizinhos.

Se qualquer um desses caminhos são alterados, a tabela de roteamento é recalculada para atualizar as informações de rota sobre cada destino da rede. Clausen e Jacquet (2003), definem na própria RFC3626, que as entradas de rota são registrados na tabela de roteamento no seguinte formato:

1. *R_dest_addr* *R_next_addr* *R_dist* *R_iface_addr*
2. ***R_dest_addr* *R_next_addr* *R_dist* *R_iface_addr***
3. „ „ „ „

Cada entrada na tabela consiste em *R_dest_addr*, *R_next_addr*, *R_dist*, e *R_iface_addr*. Essa entrada especifica que o nó identificado por *R_dest_addr* é

para esta outra rede e conseguido lendo os campos *Network Address* e *Netmask*. Estes campos são configurados no transmissor e podem direcionar para qualquer outra rede para o qual o transmissor funcione como *gateway*. Em geral se o transmissor for um *gateway* de *internet* o campo *Network Address* terá valor 0.0.0.0 e o campo *Netmask* terá valor 0.0.0.0.

3.6 Comentários Finais

Neste capítulo inicialmente foi apresentado as redes sem fio, suas classificações, padrões de mercado e seus modos de operação. Esta revisão foi extremamente necessária para se compreender o funcionamento das redes Wi-Fi e WiMAX, bem como suas capacidades e limitações, para em seguida ser apresentado as redes *Wi-Mesh*. Sobre redes *Wi-Mesh* foram apresentados o seu funcionamento, alguns protocolos proativos e reativos dentre os quais foi destacado o protocolo proativo OLSR. O OLSR se mostrou como o mais promissor de todos os comentados em questão e o qual foi detalhado o funcionamento. Como demonstrado, o mesmo possui várias implementações.

Uma dessas implementações será testada e colocada a prova no Capítulo 4 a seguir, onde será demonstrado o potencial do protocolo bem como serão testadas as principais funcionalidades que o destacaram dentre os demais protocolos apresentados.

4 MATERIAL E MÉTODOS

4.1 Escolha da implementação do protocolo

O OLSR foi desenvolvido pela INRIA e padronizado pelo IETF através RFC 3626 como protocolo experimental. Esta implementação do INRIA foi desenvolvido em conformidade com a RFC 3626 e foi projetado primeiramente para sistemas GNU/Linux recebendo o nome de OOLSR. O *download* do mesmo se encontra disponível na *internet* e pode ser baixado pela URL <http://hipercom.inria.fr/OOLSR/downloads.html>. A versão atual durante a escrita desse trabalho é a 0.99.15. Além da original implementação para GNU/Linux já se encontra disponível versões para NS2 e *Microsoft Windows*. Muitas outras implementações baseadas no OLSR podem ser encontradas⁵.

Além da implementação criada pelo própria INRIA, várias outras implementações do protocolo surgiram. A maioria deles desenvolvidos em universidades ou centros de pesquisa nas mais variadas localidades do planeta. Pelo menos cinco outras implementações de OLSR, baseadas na RFC 3626 que estão disponíveis publicamente merecem destaque:

- Uma implementação do NRL (*Naval Research Laboratory*), chamado de *nrlolsrd*, desenvolvido em C++, e essencialmente compatível com RFC 3626, com suporte ao IPv6. Esta implementação está disponível para GNU/Linux, para o simulador NS2 entre outros.
URL do projeto: <http://pf.itd.nrl.navy.mil/projects/olsr/>
- Uma implementação da *Niigata University*, chamado de *OLSR_Niigata*. Este sem maiores informações já que durante a escrita deste trabalho a URL do projeto se encontra desativada.

⁵ Exemplos dessas implementações estão disponíveis no próprio *site* da INRIA. Disponível em: <<http://hipercom.inria.fr/olsr/#OOLSR>>.

URL do projeto: <http://www.net.ie.niigata-u.ac.jp/~skame/olsr/>

- Uma implementação do LRI (*Laboratoire de Recherche en Informatique*), chamado de *qolyester*. Esta implementação desenvolvida em C++, compatível com RFC 3626, e com suporte ao IPv6 para GNU/Linux.

URL do projeto: <http://qolsr.lri.fr/>

- Uma implementação de GRC (Universidade Politécnica de Valência - Departamento de Engenharia de Computação). Esta implementação desenvolvida em conformidade com a RFC 3626, em C para Windows 2000.

URL do projeto: <http://www.grc.upv.es/software/index.html>

- Uma implementação da *Unik University*. Chamado de *olsrd*, desenvolvido em C, em conformidade com a RFC 3626. Esta implementação é uma reescrita completa do protocolo OLSR e por isso mesmo é até então uma das mais populares.

URL do projeto: <http://www.olsr.org/>

- Uma implementação da CRC (Centro de Pesquisa do Canadá de Comunicação). Desenvolvido para IPv6, e também compatível com RFC 3626.

URL do projeto:

http://www.crc.ca/en/html/manetsensor/home/research_area/routing

4.2 O OLSRD

A implementação desenvolvida pela *Unik University* com nome OLSRD é uma aplicação que utiliza-se do protocolo *Optimized Link State Routing*. Como tal, permite roteamento em malha de qualquer equipamento de rede. Ele roda em qualquer placa *Wi-Fi* que suporta modo *Ad Hoc* e, claro, em qualquer dispositivo

Ethernet. Atua na camada 2 (camada *internet*) do modelo de referência TCP/IP e é altamente portátil. Até agora, o mesmo possui implementações para:

- *Windows* (XP, *Vista* e *Windows 7*)
- *GNU/Linux* (*i386*, *arm*, *alpha*, *mips*, *xscale*)
- *OS X* (*powerpc*, *intel*, *xscale*, *iPhone*)
- *NetBSD*
- *FreeBSD*
- *OpenBSD*
- *Google phone* (*Android*, *G1*)
- *Laptop* de \$100 (do projeto OLPC)
- Entre outras

Está implementação do OLSRD é muito rápida e consome poucos ciclos de CPU, economizando energia, característica muito valioso pois o mesmo em geral é utilizado embutido em pequenos *hardwares* e dispositivos portáteis. O OLSRD é altamente escalável, existem redes *mesh* sem fio comunitárias ao redor do mundo rodando o OLSRD com milhares de nós. Conforme dados do *Project OLSRD* (2011), na Tabela 2, podemos observar alguns exemplos de redes *mesh* que rodam esta implementação do protocolo.

Tabela 2 Exemplos de redes *mesh* comunitárias com o OLSRD

<i>Rede</i>	<i>Qtde nós</i>	<i>URL Projeto</i>
Athenas Wireless Network	~ 2000 nós	http://wind.awmn.net/?page=nodes
Berlim FreiFunk.net	~ 600 nós	http://map.berlin.freifunk.net/
FunkFeuer.at	~ 400 nós	http://karte.graz.funkfeuer.at/

Está implementação do OLSRD é liberada sob uma licença BSD, o que facilita sua integração em projetos de grandes redes *mesh* comunitárias graças a essa licença ser muito liberal.

A implementação é um projeto *open source* e está em constante aperfeiçoamento, inclusive os próprios desenvolvedores encorajam novos programadores a se envolver no projeto para melhorar o protocolo. O objetivo é que o OLSRD seja uma implementação bem estruturada e bem codificada que deve ser fácil de manter, ampliar e portar para outras plataformas. Justamente por ser o mais popular e por ter uma documentação mais clara, o mesmo foi escolhido para os testes deste trabalho.

4.3 Apresentação do ambiente de testes

Conforme projeto inicial, a criação dos roteadores *Wi-Mesh* tem o propósito de serem utilizados em um projeto de cidade digital. Para tanto nos testes será demonstrado a configuração dos roteadores que funcionarão como *gateways* sem fio, interligando estações de trabalho de propósito geral a *internet* além das próprias estações de propósito geral.

Representando as estações de propósito geral serão utilizadas três *hosts*, um com sistema operacional *Debian GNU/Linux 6.0 squeeze*, outro com sistema operacional *Windows XP Professional* e outro com sistema operacional *Ubuntu Desktop 10.04*.

Representando o *roteador/gateway* será utilizado o *Debian GNU/Linux 6.0 squeeze*. A diferença básica entre os *hosts* de propósito geral e o *host* que funcionará como *gateway* é que o este *host* terá conectividade com a rede externa, ou seja ele é que encaminhará os pacotes da rede interna para a *internet* e da *internet* para a rede interna e terá conectividade ininterrupta. Esta função de *gateway* é habilitada durante a configuração do protocolo habilitando os pacotes de mensagens HNA e definindo a rota da rede externa. Para os testes realizados, foram utilizados 4 computadores todos equipados com placas e antenas para redes sem fio seguindo o seguinte layout da Figura 10.

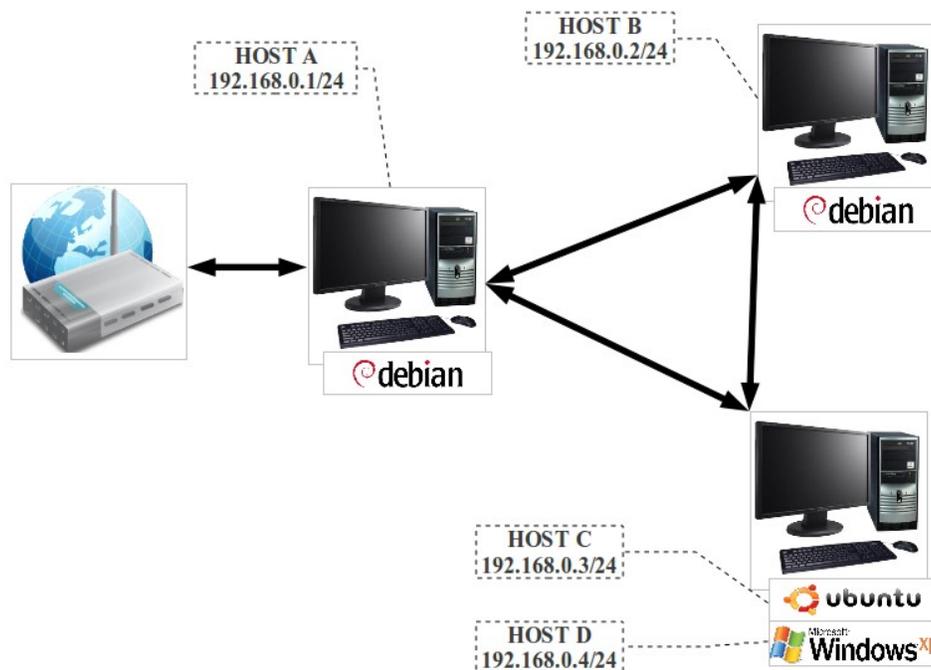


Figura 10 *Layout* inicial da rede experimental de testes

Ou seja as máquinas C (*Ubuntu Desktop 10.04*) e D (*Microsoft Windows XP*) nunca estarão ao mesmo tempo na rede, já que são a mesma máquina com dois sistemas configurados em *dual-boot*.

O sistema operacional escolhido para o roteador/*gateway* foi o GNU/Linux Debian 6.0 (6.0.2.1-i386) codinome “*squeeze*” e *kernel 2.6.32*. O *download* do mesmo pode ser feito pela URL <http://caesar.acc.umu.se/debian-cd/6.0.2.1/i386/iso-cd/debian-6.0.2.1-i386-netinst.iso> que instala apenas o sistema básico, consumindo pouco mais de 600MB de espaço em disco após instalado sendo assim ideal para a situação exposta.

As *interfaces* de rede sem fio são *RaLink RT2561/RT61 802.11g*. Os módulos de *drivers* dessas placas para o Debian 6.0 podem ser baixados via *apt-*

*get*⁶, o que facilita a instalação da mesma. Na seção 4.4 a seguir, o processo de instalação dos *drivers* dessa *interface* será explicado em detalhes. A implementação do protocolo OLSR a ser utilizada será a desenvolvida pela *Unik University* com nome OLSRD e que está disponível para *download* no *site* da *olsr.org* pela URL <http://www.olsr.org/releases/0.6/olsrd-0.6.0.tar.bz2>.

Além disso também foram instalados vários pacotes que se fizeram necessários já que a versão básica do *Debian* traz apenas pacotes básicos. Estes pacotes são detalhados também na seção 4.4 a seguir.

4.4 Instalação e configuração do protocolo OLSR no GNU/Linux

Uma vez que a distribuição *Debian GNU/Linux* 6.0 instalada corresponde a uma distribuição básica, primeiramente é preciso editar o arquivo `/etc/apt/sources.list` e acrescentar os seguintes repositórios oficiais.

```
deb http://ftp.debian.org/debian/ squeeze main contrib non-free
deb-src http://ftp.debian.org/debian/ squeeze main contrib non-free
```

Com estes repositórios, todas as dependências puderam ser instaladas sem problemas. Uma vez configurados é só atualizar a lista de repositórios com

6 O *apt-get* é um utilitário de gerenciamento de pacotes disponível para distribuições *debian like*. No início havia apenas pacotes binários do tipo `tar.gz`, `tar.bz2`, etc. Então os usuários/administradores tinham que descompactar e compilar cada programa usado em seu sistema *GNU/Linux*. Com a criação do *Debian*, surgiu a necessidade de um sistema de gerenciamento de pacotes instalados no sistema, surgindo o *dpkg* (pacote). Logo a *Red Hat* criou seu conhecido sistema *rpm*. Surgiu se então a necessidade de um sistema de gerenciamento de pacotes instalados no sistema. Uma forma rápida, prática e eficiente de instalar pacotes, gerenciar dependências automaticamente e cuidando de arquivos de configuração ao atualizar. O projeto *Debian*, novamente pioneiro, criou o APT ou *Advanced Packaging Tool*. Outras distribuições acabaram portando o sistema APT e incorporando em seus sistemas. O manual deste utilitário está disponível na URL: <http://www.debian.org/doc/manuals/apt-howto/>.

apt-get update.

```
# apt-get update
```

Após a atualização das lista de repositórios será necessário instalar o seguintes pacotes:

- Pacote para descompactar o arquivo que contém os fontes da implementação do protocolo.

```
# apt-get install bzip2
```

- Pacote que contém o binário necessário para compilação do código.

```
# apt-get install make
```

- Compiladores *gcc* e *g++* necessários para compilar o código.

```
# apt-get install gcc g++
```

- Pacote que contém um lista de pacotes considerados essenciais.

```
# apt-get install build-essential
```

- Pacotes *flex* e *bison* que são códigos projetadas para auxiliar no desenvolvimento de compiladores.

```
# apt-get install bison flex
```

- Pacote que instala algumas ferramentas a serem utilizadas na configuração do protocolo como o utilitário *iwconfig*, *iwlist*, etc

```
# apt-get install wireless-tools
```

Agora instalá se o módulo para placa de rede e carrega-o para o *kernel*. No caso as interfaces utilizadas para a realização dos testes são interfaces que utilizam o *chip RaLink RT2561/RT61 802.11g*.

```
# apt-get install firmware-ralink wireless-tools
```

```
# modprobe rt61pci
```

Uma vez instalados todas as dependências, o sistema está pronto para receber a instalação do protocolo OLSRD. Os passos abaixo descrevem como o processo de instalação foi feito. Também é possível instalar o protocolo *olsrd* da *Unik University* via *apt-get*, porém optou-se pela instalação através do código-fonte para instalar a última versão do protocolo.

- *Download* da última versão do protocolo OLSRD via *wget*.
wget -c http://www.olsr.org/releases/0.6/olsrd-0.6.0.tar.bz2
- Descompactação do pacote.
tar -jxvf olsrd-0.6.0.tar.bz2
- Compilação.
cd olsrd-0.6.0
make
make install

Após a instalação do OLSRD, o passo seguinte é editar o arquivo */etc/olsrd.conf* e configurá-lo. Uma cópia do arquivo original pode ser visualizada no ANEXO A. Aqui a configuração será bem mais simples. Com um editor de texto qualquer (*vi*, *nano*, *pico*) edita-se o arquivo da seguinte forma:

```

root@desktoplinux: /home/zilmar
DebugLevel 1
IpVersion 4
Hna4
{
0.0.0.0 0.0.0.0
}
OlsrPort 698
UseHysteresis no
TcRedundancy 2

Interface "wlan0"
{
    Mode "mesh"
    Ip4Broadcast 255.255.255.255
    HelloInterval 2.0
    HelloValidityTime 20.0
    TcInterval 5.0
    TcValidityTime 300.0
    MidInterval 5.0
    MidValidityTime 300.0
    HnaInterval 5.0
    HnaValidityTime 300.0
    Weight 0
}

```

Figura 11 Visão do arquivo de configuração do OLSRD

Conforme Frosi e Schaeffer (2010), os parâmetros do arquivo de configuração `olsrd.conf` são descritos da seguinte forma:

- *DebugLevel*: Este parâmetro seta o nível de *debug*. Em outras palavras significa a quantidade de mensagens a serem mostradas na saída padrão. Nível 0 o serviço rodará como *daemon*, nada será mostrado na saída padrão. Com nível 1 o serviço mostrará uma tela com o conjunto de nós vizinhos a 1 e 2 *hops*, e informações de topologia.
- *IpVersion*: Este parâmetro aponta qual é a versão do protocolo IP a ser usado. O protocolo OLSRD suporta as versões 4 e 6 do protocolo IP.
- *Hna4*: Este parâmetro define o conteúdo das mensagens HNA. Quando colocamos no parâmetro `0.0.0.0 0.0.0.0` estamos setando o *host* para ser o *gateway default* dos demais *hosts*. Da mesma forma poderia anunciar via mensagens HNA *links* com outras redes apenas adicionando seus

endereços e máscaras.

- *OlsrPort*: Indica qual porta será usada pelo protocolo. Por padrão a porta usada é a 698 e as comunicações de OLSR ocorrem fazendo uso do protocolo UDP.
- *UseHysteresis*: Este parâmetro ativa (*yes*) ou desativa (*no*) o *Link Histerisis*. Aqui o mesmo não foi utilizado pois essa implementação do protocolo já controla a qualidade dos *links* nos dois sentidos das vias de tráfego, sendo muito mais eficiente.
- *TcRedundancy*: Este parâmetro descreve a quantidade de informações que serão enviadas nas mensagens TC. O valor 2 para o parâmetro foi utilizado para que todo o conjunto de nós vizinhos fosse adicionado nas mensagens TC. Se o valor estivesse setado para 1, as mensagens TC, seriam enviadas apenas os nós que escolheram o remetente como MPR, e todos os MPR vizinhos. O valor 0 enviaria somente os vizinhos que escolheram o remetente da mensagem como MPR.
- *Interface*: Este parâmetro mostra qual *interfaces* que utilizará o protocolo, seguido de suas configurações.
- *Mode*: Este configura o modo de operação da *interface*. No caso funcionará em malha (*mesh*).
- *Ip4Broadcast*: Este parâmetro seta qual será o endereço *broadcast* do OLSR. No caso o endereço 255.255.255.255 é um endereço de *broadcast* mais alto do protocolo IP versão 4.
- *HelloInterval*: Este parâmetro define o intervalo entre o envio de mensagens HELLO.
- *HelloValidityTime*: Este configura a validade máxima das mensagens HELLO. Este é enviado em um dos campos da mensagem HELLO, e é interpretado por todos os nós que receberem a mensagem.
- *TcInterval*: Este parâmetro configura o intervalo entre o envio de

mensagens TC.

- *TcValidityTime*: Este parâmetro aponta a validade máxima da mensagens TC. Da mesma forma que nas mensagens HELLO, é interpretado por todos os nós que receberem as mensagens TC.
- *MidInterval*: Este parâmetro descreve o intervalo entre o envio de mensagens MID.
- *MidValidityTime*: Este define a validade das mensagens MID.
- *HnaInterval*: Este campo descreve o intervalo de envio de mensagens HNA.
- *HnaValidityTime*: Este campo define a validade das mensagens HNA. Mensagens HNA são enviadas apenas por nós que tenham uma entrada de configuração no parâmetro HNA.
- *Weight*: Esta opção aponta para o OLSR qual *link* deve ser escolhido quando múltiplos *links* são encontrados entre dois nós. *Links* com menor valor são escolhidos primeiro.

Seguindo na configuração do roteador *Wi-Mesh*, configura-se a *interface* de rede sem fio. No caso a mesma foi reconhecida como *wlan0*. Os passos também são relativamente simples:

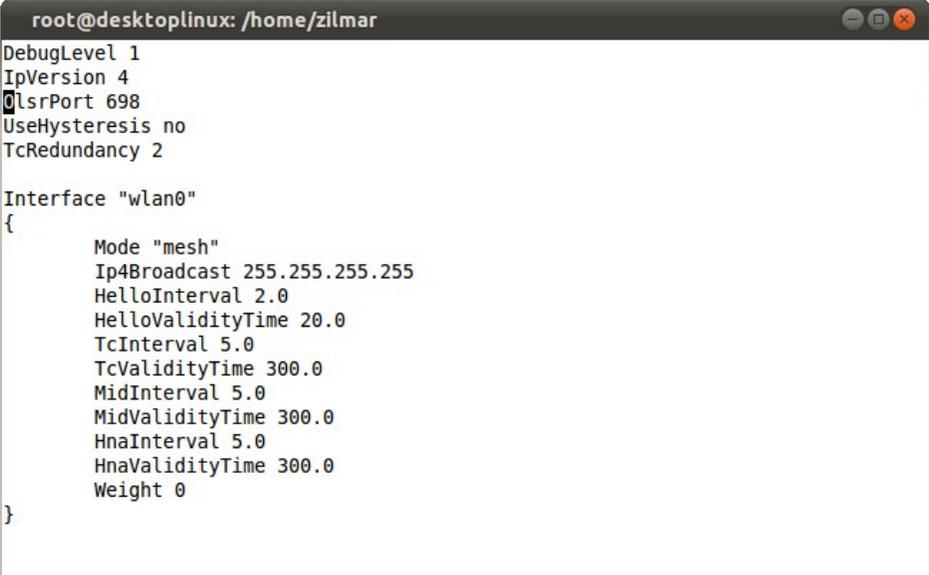
- Desativar a *interface* de rede.
ifconfig wlan0 down
- Criar o ESSID, o canal e setar o modo de operação da rede para *Ad Hoc*. No caso foi escolhido o ESSID “wimesh” e o canal 6.
iwconfig wlan0 essid wimesh
iwconfig wlan0 channel 6
iwconfig wlan0 mode ad-hoc
- Configurar o IP do *host* e levantar a *interface* de rede.
ifconfig wlan0 192.168.0.1/24 up

- Não há mais nada o que configurar a não ser levantar o serviço. O que pode ser feito executando do *daemon olsrd* no terminal.

```
# olsrd
```

4.5 Configuração de clientes GNU/Linux em modo *Ad Hoc*

Em uma rede *Wi-Mesh* todas as estações são roteadores em potencial, assim sendo não há muitas diferenças de configuração de uma estação de trabalho de propósito geral com GNU/Linux, da estação que funcionará como *gateway* para outra rede. Existem duas diferenças básicas. A primeira é o HNA que já foi comentado anteriormente, pois uma vez configurado o mesmo fará com que o *host* funcione na rede *mesh* se identificando como *gateway* para outra rede e este apresentará seus serviços através de mensagens HNA. A outra diferença é que deveremos ativar o NAT nas estações que funcionarão como *gateways*, o que não será necessários nas estações clientes. O arquivo de configuração nos clientes ficaria similar ao apresentado na Figura 12.



```
root@desktoplinux: /home/zilmar
DebugLevel 1
IpVersion 4
OlsrPort 698
UseHysteresis no
TcRedundancy 2

Interface "wlan0"
{
    Mode "mesh"
    Ip4Broadcast 255.255.255.255
    HelloInterval 2.0
    HelloValidityTime 20.0
    TcInterval 5.0
    TcValidityTime 300.0
    MidInterval 5.0
    MidValidityTime 300.0
    HnaInterval 5.0
    HnaValidityTime 300.0
    Weight 0
}
```

Figura 12 Visão do arquivo de configuração do OLSRD nos clientes

Observa-se que a única diferença com relação aos *hosts* que funcionarão como *gateways* e a ausência do parâmetro *Hna4*. Já as configurações da *interfaces* sem fio nos clientes a única alteração é no endereço IP. Por exemplo:

```
# ifconfig wlan0 down
# iwconfig wlan0 essid wimesh
# iwconfig wlan0 channel 6
# iwconfig wlan0 mode ad-hoc
# ifconfig wlan0 192.168.0.2/24 up
```

Não havendo mais nada o que configurar é só levantar o serviço.

```
# olsrd
```

4.6 Configuração de clientes MS/Windows em modo *Ad Hoc*

Apesar do trabalho se propor a montar roteadores *Wi-Mesh* com GNU/Linux, não poderia se deixar de testar o funcionamento do protocolo no *Microsoft Windows*, isso porque a grande maioria dos usuários domésticos utilizam o Windows em seus computadores pessoais. O objetivo é utilizar o protocolo em um ambiente de uma cidade digital, obviamente inundada por diversos sistemas operacionais diferentes e, em sua grande maioria, sistemas *Microsoft Windows*. O teste no *Windows* foi realizado em um *notebook* utilizado como estação de trabalho de propósito geral equipado com *Windows XP Professional*. O *download* da versão do protocolo OLSRD para *Windows* está disponível na URL <http://olsr.org/releases/0.5/olsrd-0.5.6-r3-pre-cac1df64dcd5-setup.exe>.

Uma vez que o *download* foi concluído o protocolo pode ser instalado

clicando duas vezes sobre o executável e sempre clicando no botão *NEXT* até a conclusão. Um atalho chamado *OLSR Swith* será criado na área de trabalho.

Para configurar a estação *Windows* basta configurar o IP da rede e o DNS na *interface* sem fio, nesse caso foi utilizado o IP 192.168.0.4/24 conforme pode ser observado na Figura 13.

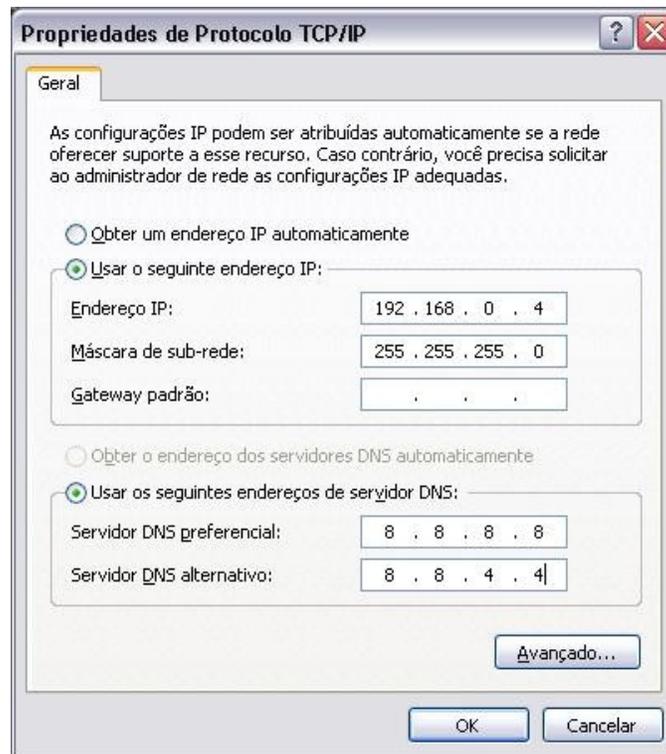


Figura 13 Configuração de endereços *Windows*

Não é necessário configurar o *gateway* uma vez que o próprio OLSR encontrará as estações HNA responsáveis por encaminhar os pacotes para outras redes. Uma vez configurado o IP na *interface* sem fio basta conectar na rede conforme pode ser observado na Figura 14 e após executar o ícone *OLSR Swith*. A Figura 15 mostra o aplicativo *OLSR Swith* para configuração do protocolo no *Microsoft Windows*.

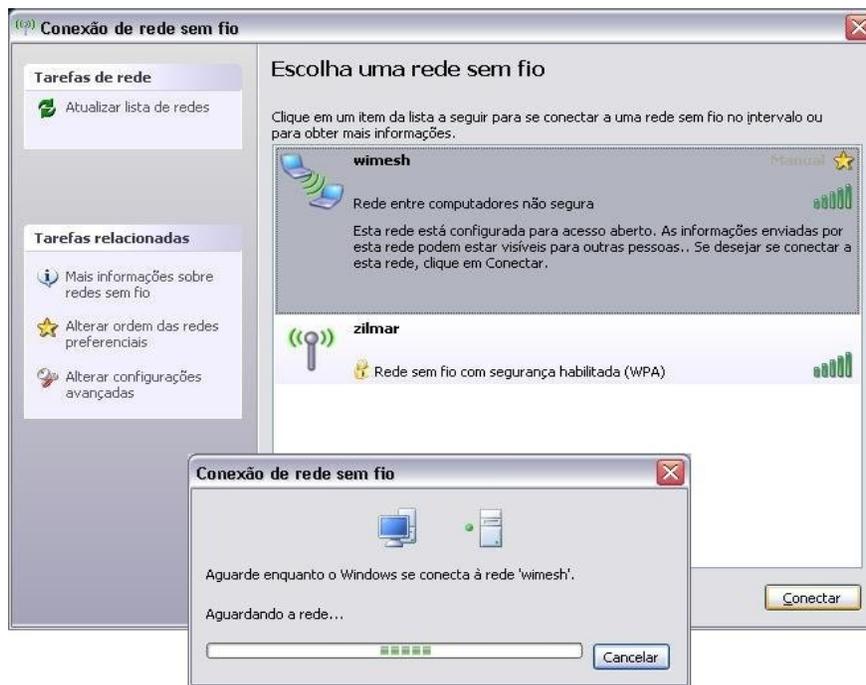


Figura 14 Conectando na rede *wimesh*

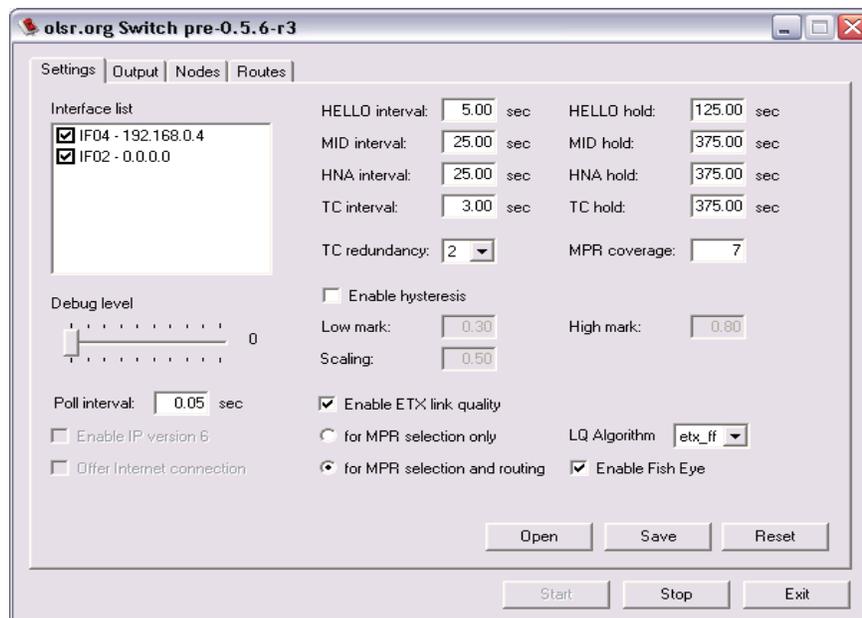


Figura 15 Screenshot da tela de configuração do aplicativo OLSR *Switch*

5 RESULTADOS E DISCUSSÃO

Como apresentado no capítulo anterior, temos um ambiente de testes configurado com quatro *hosts* distribuídos conforme a Tabela 3 a seguir. As estações C e D não aparecem ao mesmo tempo nos testes por estarem configuradas na mesma máquina em *dual-boot*.

Tabela 3 Configurações dos *hosts* dentro do *layout* da rede

Nº <i>Host</i>	Sistema Operacional	IP	Gateway de Internet
A	<i>Debian GNU/Linux 6.0</i>	192.168.0.1/24	SIM
B	<i>Debian GNU/Linux 6.0</i>	192.168.0.2/24	NÃO
C	<i>Ubuntu GNU/Linux 10.10</i>	192.168.0.3/24	NÃO
D	<i>Microsoft Windows XP</i>	192.168.0.4/24	NÃO

Para mostrar os resultados serão apresentados três cenários diferentes onde ocorreram os testes. Cada cenário apresenta um *layout* de rede diferente, ou seja situações diferentes em que a rede foi exposta para ver o funcionamento do protocolo e como o mesmo reage nessas situações, inclusive com *hosts* móveis se movimentando na rede.

5.1 Cenário 1

Neste primeiro cenários os *hosts* A, B e C são inicializados seguindo o seguinte *layout* de rede da Figura 16. As estações são então carregadas e então o *daemon olsrd* é executado. O resultado pode ser visto nas Figuras 17,18 e 19 que mostram o que cada um dos 3 *hosts* encontrou na rede.

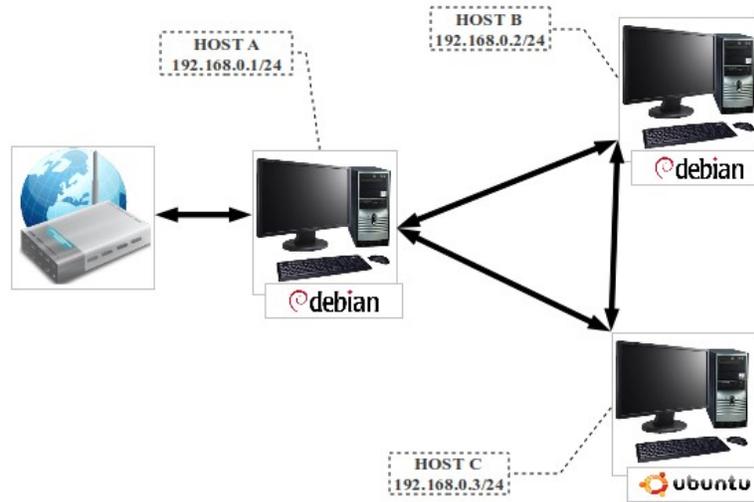


Figura 16 *Layout* de rede apresentado para teste no cenário 1

Na Figura 17 vemos a *screenshot* do *host B*, e podemos observar que a mesma mostra com quais outros *hosts* o *host B* tem um *link*, ou seja quem são seus vizinhos a um *hop* de distância e a dois *hops* de distância.

```

root@desktoplinux: /home/zilmar
root@desktoplinux: /home/zilmar x root@desktoplinux: /home/zilmar x
--- 22:59:26.076983 ----- LINKS
IP address      hyst      LQ      ETX
192.168.0.1    0.000    1.000/1.000    1.000
192.168.0.3    0.000    0.713/1.000    1.400

--- 22:59:26.077025 ----- TWO-HOP NEIGHBORS
IP addr (2-hop) IP addr (1-hop) Total cost
192.168.0.3     192.168.0.1   2.274
192.168.0.1     192.168.0.3   2.063

--- 22:59:26.077050 ----- TOPOLOGY
Source IP addr  Dest IP addr    LQ      ETX
192.168.0.1    192.168.0.2    1.000/1.000    1.000
192.168.0.1    192.168.0.3    0.784/1.000    1.274
192.168.0.2    192.168.0.1    1.000/1.000    1.000
192.168.0.2    192.168.0.3    0.713/1.000    1.400
192.168.0.3    192.168.0.1    1.000/1.000    1.000
192.168.0.3    192.168.0.2    1.000/0.940    1.063

```

Figura 17: *Screenshot* do *host B*, mostrando as rotas para seus vizinhos.

É possível observar que os *hosts* A e C estão a um *hop* de distância, porém ao mesmo tempo eles também possuem rotas a dois *hops* de distância. Observe que pode-se chegar ao *host* C saltando pelo *host* A e vice-versa.

```

root@desktoplinux: /home/zilmar
root@desktoplinux: /home/zilmar
--- 22:59:06.933055 ----- LINKS
IP address      hyst      LQ      ETX
192.168.0.2     0.000    1.000/1.000  1.000
192.168.0.3     0.000    1.000/1.000  1.000

--- 22:59:06.933094 ----- TWO-HOP NEIGHBORS
IP addr (2-hop) IP addr (1-hop) Total cost
192.168.0.3     192.168.0.2   2.063
192.168.0.2     192.168.0.3   2.063

--- 22:59:06.933117 ----- TOPOLOGY
Source IP addr  Dest IP addr      LQ      ETX
192.168.0.1     192.168.0.2     1.000/1.000  1.000
192.168.0.1     192.168.0.3     1.000/1.000  1.000
192.168.0.2     192.168.0.1     1.000/1.000  1.000
192.168.0.2     192.168.0.3     0.796/0.568  2.209
192.168.0.3     192.168.0.1     1.000/1.000  1.000
192.168.0.3     192.168.0.2     1.000/0.940  1.063

```

Figura 18 Screenshot do *host* A, mostrando as rotas para seus vizinhos.

Na Figura 18 é possível ver a mesma situação da rede porém visto agora do *host* A. Observe que ele mantém rotas a um *hop* de distância com os *hosts* B e C ao mesmo tempo que ele possui rotas a 2 *hops* de distância com as mesmos *hosts*. Ou seja o *host* A pode trocar mensagens com o *host* B com 1 *hop* ou com 2 *hops* passando primeiramente pelo *host* C e vice-versa.

Por fim nessa primeira situação vemos na Figura 19 a visão do *host* C, que possui rotas a 1 *hop* de distância e a 2 *hops* de distância com os *hosts* A e B.

```

Arquivo  Editar  Ver  Terminal  Abas  Ajuda
*** olsr.org - 0.6.0 (2011-08-23 10:32:57 on ubuntu) ***

--- 10:57:24.870882 ----- LINKS

IP address      hyst      LQ      ETX
192.168.0.2     0.000    0.427/0.442  5.278
192.168.0.1     0.000    0.462/0.442  4.876

--- 10:57:24.870977 ----- TWO-HOP NEIGHBORS

IP addr (2-hop) IP addr (1-hop) Total cost
192.168.0.2     192.168.0.1  5.876
192.168.0.1     192.168.0.2  6.278

--- 10:57:24.871035 ----- TOPOLOGY

Source IP addr  Dest IP addr      LQ      ETX
192.168.0.1    192.168.0.2      1.000/1.000  1.000
192.168.0.1    192.168.0.3      0.442/0.419  5.378
192.168.0.2    192.168.0.1      1.000/1.000  1.000
192.168.0.2    192.168.0.3      0.442/0.380  5.932
192.168.0.3    192.168.0.1      0.462/0.442  4.876
192.168.0.3    192.168.0.2      0.427/0.442  5.278

```

Figura 19 Screenshot do *host C*, mostrando as rotas para seus vizinhos

5.2 Cenário 2

Neste segundo cenário os *hosts A*, *B* e *D* são inicializados seguindo o *layout* de rede da Figura 20 a seguir.

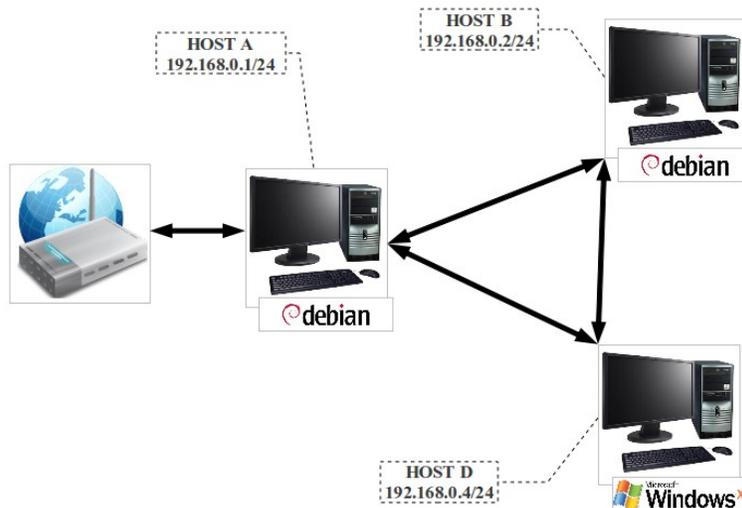


Figura 20 *Layout* de rede apresentado para teste no cenário 2

As estações são então carregadas e então o *daemon olsrd* é executado. O resultado pode ser visto nas Figuras 21, 22, 23 e 24 a seguir.

Na Figura 21 vê-se o *screenshot* do *host B* e mais uma vez a mesma mostra com quais outros *hosts*, o *host B* tem um *link*, ou seja quem são seus vizinhos a um *hop* de distância (no caso são os *hosts A* e *D*) e a dois *hops* de distância.

```

root@desktoplinux: /home/zilmar
root@desktoplinux: /home/zilmar x root@desktoplinux: /home/zilmar x
--- 23:14:40.700215 ----- LINKS
IP address      hyst      LQ      ETX
192.168.0.1     0.000    1.000/1.000  1.000
192.168.0.4     0.000    0.812/0.944  1.303

--- 23:14:40.700252 ----- TWO-HOP NEIGHBORS
IP addr (2-hop) IP addr (1-hop) Total cost
192.168.0.4     192.168.0.1   2.000
192.168.0.1     192.168.0.4   2.365

--- 23:14:40.700276 ----- TOPOLOGY
Source IP addr  Dest IP addr      LQ      ETX
192.168.0.1    192.168.0.2      1.000/1.000  1.000
192.168.0.1    192.168.0.4      1.000/1.000  1.000
192.168.0.2    192.168.0.1      1.000/1.000  1.000
192.168.0.2    192.168.0.4      0.812/0.944  1.303
192.168.0.4    192.168.0.1      0.940/1.000  1.063
192.168.0.4    192.168.0.2      0.886/0.812  1.390

```

Figura 21 *Screenshot* do *host B*, mostrando as rotas para seus vizinhos

Assim como no cenário anterior é possível observar que os *hosts A* e *D* estão a um *hop* de distância, ao mesmo tempo que também possuem rotas a dois *hops* de distância. Ou seja o *host B* pode-se chegar ao *host D* saltando pelo *host A* e vice-versa.

Na Figura 22 é possível ver a mesma situação, só que vista agora do *host A*. Ele mantém rotas a um *hop* de distância com os *hosts B* e *D* ao mesmo tempo que ele possui rotas a 2 *hops* de distância com as mesmos *hosts*.

```

root@desktoplinux: /home/zilmar
root@desktoplinux: /home/zilmar
--- 23:14:49.702158 ----- LINKS
IP address      hyst      LQ      ETX
192.168.0.2     0.000    1.000/1.000  1.000
192.168.0.4     0.000    1.000/1.000  1.000

--- 23:14:49.702193 ----- TWO-HOP NEIGHBORS
IP addr (2-hop) IP addr (1-hop) Total cost
192.168.0.4     192.168.0.2   2.203
192.168.0.2     192.168.0.4   2.390

--- 23:14:49.702216 ----- TOPOLOGY
Source IP addr  Dest IP addr      LQ      ETX
192.168.0.1    192.168.0.2      1.000/1.000  1.000
192.168.0.1    192.168.0.4      1.000/1.000  1.000
192.168.0.2    192.168.0.1      1.000/1.000  1.000
192.168.0.2    192.168.0.4      0.812/0.944  1.303
192.168.0.4    192.168.0.1      0.937/1.000  1.066
192.168.0.4    192.168.0.2      0.944/0.812  1.303

```

Figura 22 Screenshot do *host A*, mostrando as rotas para seus vizinhos

Observa-se agora na Figura 23 a visão do *host D*, que é uma estação *Microsoft Windows* e que possui rotas a 1 *hop* de distância e a 2 *hops* de distância com os *hosts B e A*.

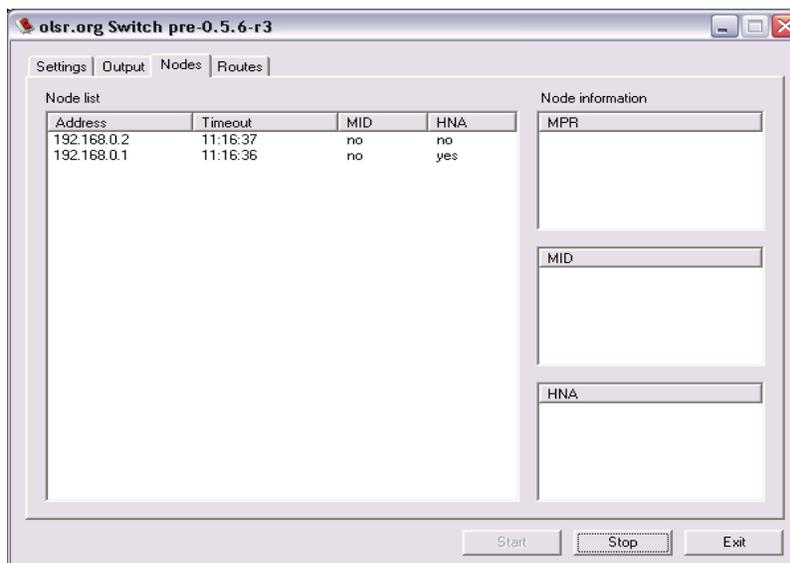


Figura 23 Screenshot do *host D*, mostrando os hosts vizinhos

Na Figura 23, pode-se identificar de forma mais clara que o *host A* é a estação HNA, ou seja é um *gateway* de ligação para outra rede. Na Figura 24 a seguir vemos no mesmo *host D* a tabela de roteamento.

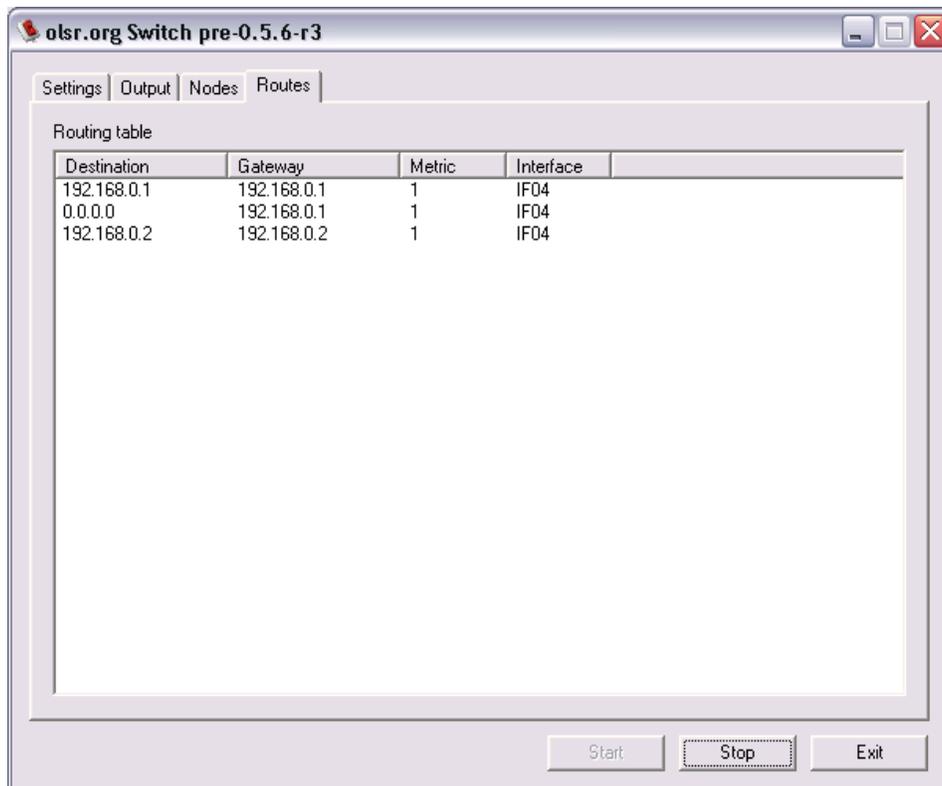


Figura 24 Screenshot do *host D*, mostrando as rotas para seus vizinhos

5.3 Cenário 3

Neste terceiro cenário os *hosts A*, *B* e *C* são inicializados seguindo o *layout* de rede da Figura 25 a seguir.

As estações são então carregadas e então o *daemon olsrd* é executado. O resultado pode ser vistos nas Figuras 26, 27 e 28 que mostram o que cada um

dos 3 *hosts* encontrou na rede.

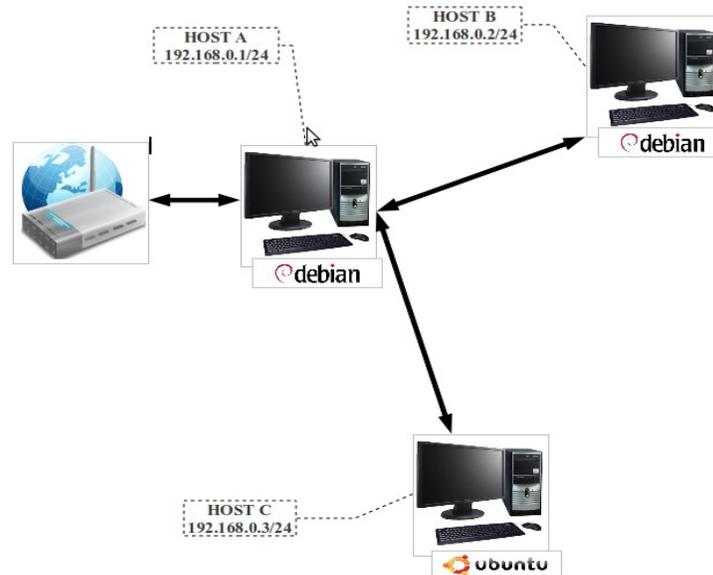


Figura 25 *Layout* de rede apresentado para teste no cenário 2

No estudo deste cenário onde estão as mesmos *hosts* do cenário 1, porém com a diferença que e a movimentação do *host C* para longe do *host B*, assim sendo o mesmo perde a conectividade direta com o *host B*. A ideia aqui é analisar como a rede vai reagir a partir dessa situação.

Na Figura 26 a seguir pode-se observar a visão do *olsrd* a partir do *host B*. Observa-se que o mesmo detectou os *hosts A* e *C*, porém o *host A* está um *hop* de distância enquanto o *host C* está o 2 *hops* de distância. Ou seja o *host B* consegue se comunicar diretamente com o *host A*, e para se comunicar com o *host C* ele precisa passar pelo *host A* para só ai alcançar o *host C*.

Comparando então as Figuras 17 e 26 ambas visões do *host B*, porém em situações diferentes, observamos que o OLSR fez automaticamente uma adaptação nas rotas para que o *host B* não perdesse conectividade com o *host C*. Está é a prova maior do funcionamento do OLSR, fazendo adaptações na rede em malha para manter a conectividade entre os *hosts*.

```

root@desktoplinux: /home/zilmar
root@desktoplinux: /ho... x root@desktoplinux: /ho... x root@desktoplinux: /ho... x
*** olsr.org - 0.6.0 (2011-08-23 03:36:22 on debian) ***

--- 22:37:21.707985 ----- LINKS

IP address      hyst      LQ      ETX
192.168.0.1     0.000    1.000/1.000  1.000

--- 22:37:21.708021 ----- TWO-HOP NEIGHBORS

IP addr (2-hop) IP addr (1-hop) Total cost
192.168.0.3     192.168.0.1   2.138

--- 22:37:21.708039 ----- TOPOLOGY

Source IP addr  Dest IP addr      LQ      ETX
192.168.0.1    192.168.0.2     1.000/1.000  1.000
192.168.0.1    192.168.0.3     0.878/1.000  1.138
192.168.0.2    192.168.0.1     1.000/1.000  1.000
192.168.0.3    192.168.0.1     1.000/0.886  1.128

```

Figura 26 Screenshot do *host B*, mostrando os *hosts* vizinhos

Na Figura 27 a seguir que mostra a visão a partir do *host A*, pode-se observar o *olsrd* detectou os *hosts B* e *C*. Os dois estão a um *hop* de distância do *host A*. Neste caso não há *hosts* a dois *hops* de distância do *host A*. Isso significa que o *host A* não possui rotas alternativas para os demais *hosts* e nesse caso fica claro que o *host A* funcionará como um roteador intermediário para a comunicação dos *hosts B* e *C*.

Uma diferença básica nas rotas do *host A* podem ser observadas comparando as Figuras 18 e 27. Como os *hosts B* e *C* se afastaram, as rotas a 2 *hops* de distância (Figura 17) que o *host A* tinha para os *hosts B* e *C* foram

removidas (Figura 27), provando mais uma vez que o protocolo OLSR faz adaptações assim que ele faz a análise do estado da rede enviando as mensagens HELLO.

```

root@desktoplinux: /home/zilmar
root@desktoplinux: /ho... ✖ root@desktoplinux: /ho... ✖ root@desktoplinux: /ho... ✖
*** olsr.org - 0.6.0 (2011-08-23 01:36:57 on debian) ***

--- 15:57:29.953531 ----- LINKS

IP address      hyst      LQ      ETX
192.168.0.3     0.000    0.878/1.000  1.138
192.168.0.2     0.000    1.000/1.000  1.000

--- 15:57:29.953572 ----- TWO-HOP NEIGHBORS

IP addr (2-hop) IP addr (1-hop) Total cost

--- 15:57:29.953586 ----- TOPOLOGY

Source IP addr  Dest IP addr      LQ      ETX
192.168.0.1     192.168.0.2      1.000/1.000  1.000
192.168.0.1     192.168.0.3      0.878/1.000  1.138
192.168.0.2     192.168.0.1      1.000/1.000  1.000
192.168.0.3     192.168.0.1      1.000/0.886  1.128

```

Figura 27 Screenshot do *host A*, mostrando os *hosts* vizinhos

Na Figura 28 observa-se a visão do *olsrd* a partir do *host C* para este novo cenário. Observa-se que o mesmo detectou os *hosts A* e *B*, porém o *host A* está 1 *hop* de distância enquanto o *host B* está o 2 *hops* de distância. Ou seja o *host C* consegue se comunicar diretamente com o *host A*, e para se comunicar com o *host B* ele precisa passar pelo *host A* usando-o como roteador intermediário para só ai alcançar o *host B*.

Comparando então as Figuras 19 e 28 que mostram a *visão* do *host C* em dois cenários diferentes porém com as mesmas *hosts* na rede, observamos mais uma vez a adaptabilidade do protocolo, ou seja a sua capacidade de reorganizar a rede quando há uma movimentação dos *hosts*.

```

Arquivo  Editar  Ver  Terminal  Abas  Ajuda
*** olsr.org - 0.6.0 (2011-08-23 10:32:57 on ubuntu) ***

--- 16:06:59.650131 ----- LINKS

IP address      hyst      LQ      ETX
192.168.0.1     0.000    0.109/0.113  80.079

--- 16:06:59.650191 ----- TWO-HOP NEIGHBORS

IP addr (2-hop) IP addr (1-hop) Total cost
192.168.0.2     192.168.0.1   INFINITE

--- 16:06:59.650231 ----- TOPOLOGY

Source IP addr  Dest IP addr      LQ      ETX
192.168.0.1    192.168.0.2     1.000/0.164  INFINITE

```

Figura 28 Screenshot do *host C*, mostrando os *hosts* vizinhos

5.4 Observação da movimentação dos pacotes de mensagens na rede

Durante todo e qualquer processo de transmissão em redes há a movimentação de pacotes de mensagens. No caso de uma rede *Wi-Mesh* com *OLSR* as principais mensagens são as com pacotes HELLO, HNA e TC. O objetivo desta seção é mostrar as pacotes com estas mensagens, de forma a visualizar a movimentação destes. A captura desses pacotes foram realizadas no *host A* com a utilização do *tcpdump*, salvando o resultado em um arquivo e posteriormente analisando o arquivo gerado com o aplicativo *wireshark*.

O *tcpdump* e o *wireshark* são dois analisadores de protocolos muito conhecidos de gerentes e administradores de redes, pois são duas das principais ferramentas de redes utilizadas para este fim.

No host A o *tcpdump* foi executado da seguinte forma:

```
# tcpdump -i wlan0 port 698 -s0 -w network-scan.pcap
```

- *-i wlan0* : Especificação da *interface* de rede a ser monitorada;
- *port 698* : Porta a ser monitorada. A 698 é a porta padrão do OLSR;
- *-s0* : Configuração para salvar todo o pacote;
- *-w network-scan.pcap* : Salvar o resultado em arquivo com nome *network-scan.pcap* para posterior análise utilizando com o *wireshark* .

Após ser gerado o arquivo com alguns pacotes, o mesmo foi analisado com o *wireshark*, de onde puderam ser observados os seguintes pacotes de mensagens, mostrados nas Figuras 29, 30 e 31.

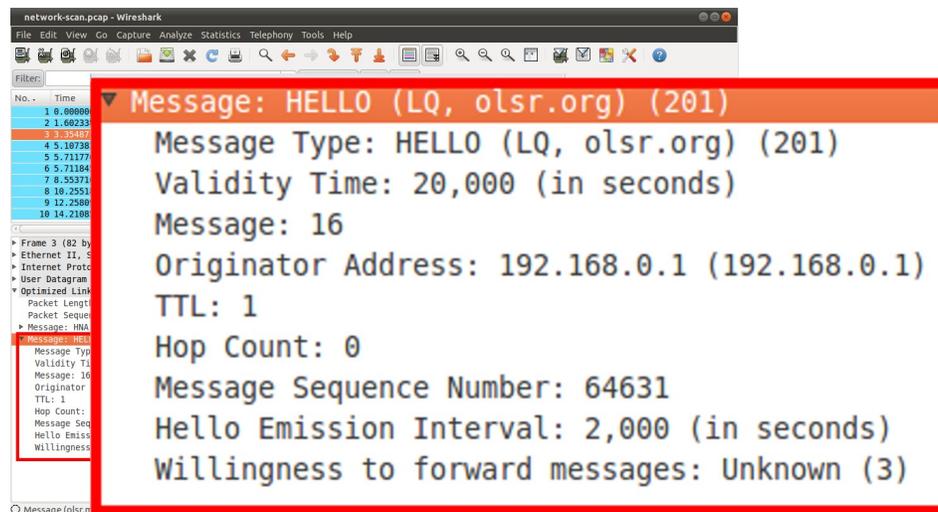


Figura 29 Screenshot do *wireshark*, pacote de mensagem HELLO

Na Figura 29, pode-se observar um dos pacotes HELLO capturados pelo *tcpdump* e mostrados aqui pelo *wireshark*. Pode-se observar também o uso do UDP para a transmissão, já que o mesmo é o pacote da camada de transporte utilizado pelo OLSR. Comparando a parte destacada da Figura 29 com a Figura

6, nota-se o uso dos campos do pacote HELLO.

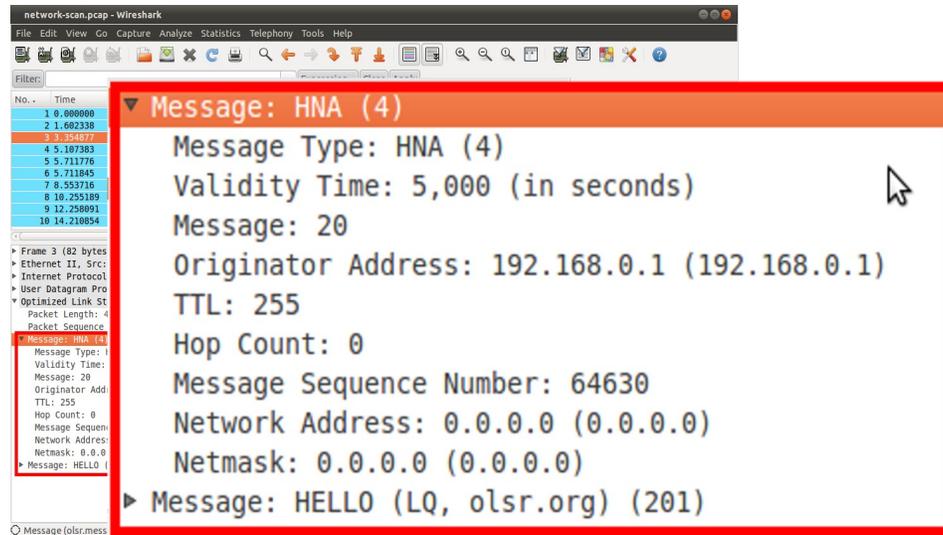


Figura 30 Screenshot do wireshark, pacote de mensagem HNA

Na Figura 30, observa-se um dos pacotes HNA capturados pelo *tcpdump* e mostrados aqui pelo *wireshark*. Comparando a parte destacada da Figura 30 com a Figura 9, observa-se o uso dos campos do pacote HNA.

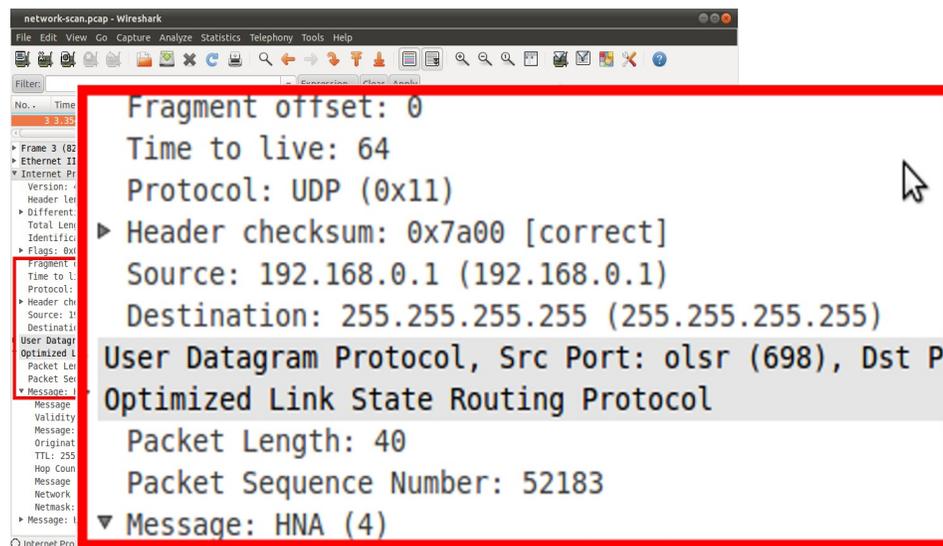


Figura 31 Screenshot do wireshark, pacote HNA enviado por broadcast

O mais interessante é analisar que a mensagem foi originada do *host A* e enviada por *broadcast* na rede conforme pode ser observado na Figura 31, anunciando que o *host A* é um *gateway* para outra rede. No caso observa-se na Figura 30 o campo *Network Address* veremos que o mesmo aponta para a rede 0.0.0.0 com máscara 0.0.0.0, deixando claro que o *host A* é o *gateway* de *internet* da rede.

Finalmente temos na Figura 32, um pacote do tipo TC capturado pelo *tcpdump* e visualizado pelo *wireshark*.

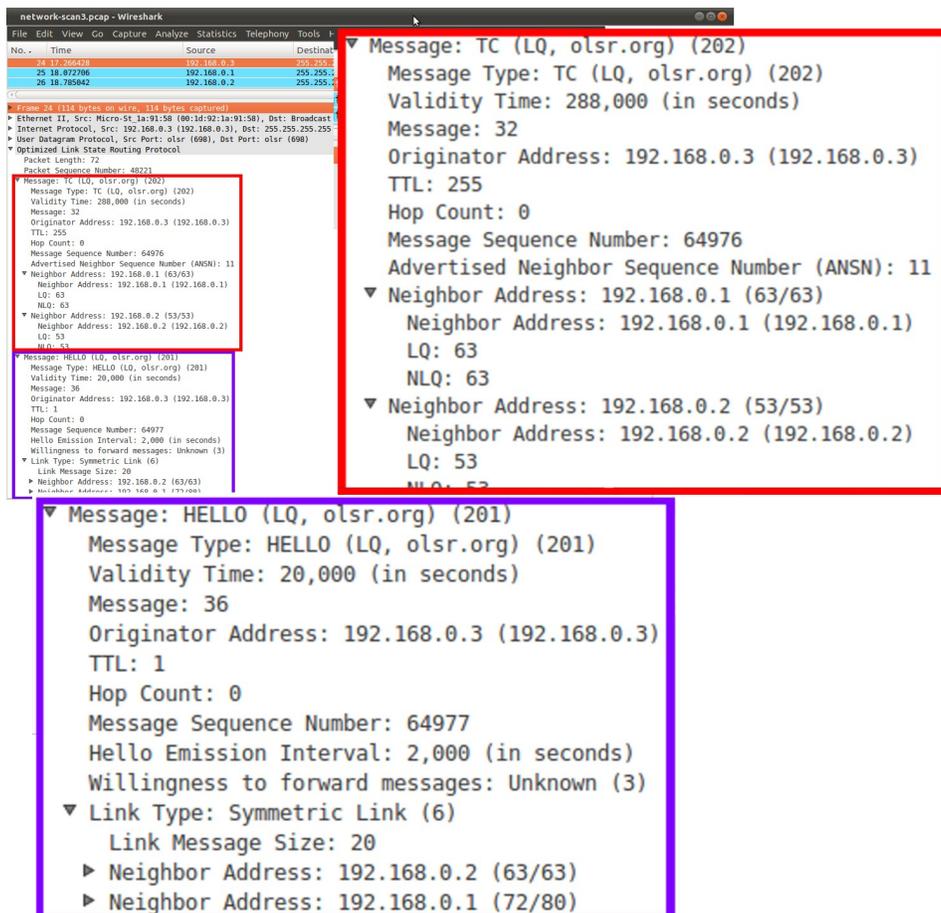


Figura 32 Screenshot do wireshark, pacote de mensagem TC e HELLO

Comparando a parte destacada da Figura 32 com a Figura 8, é possível notar os vários campos do pacote TC. Nesse pacote o mais interessante é observar que o *host C* foi o originador do pacote e o mesmo pacote indicando quem são seus nós vizinhos (*host A* e *C*), através dos campos *Neighbor Address* que contém os endereços IPs desses vizinhos. Na Figura 32 também foi destacado o pacote de mensagem HELLO.

5.5 Análise e resultados

A principal vantagem das redes *Wi-Mesh Ad Hoc* é a de ser capaz de se auto configurar. Como pode ser observado nos testes aplicados nos três cenários expostos, o protocolo de roteamento sempre organiza a rede montando um malha entre os seus nós gerando rotas para todos os destinos da rede e tornando a rede mais inteligente. O protocolo busca deixar a topologia da rede adequada para que o nó se comunique com outro buscando o menor ou o melhor caminho. Ainda há muito o que melhorar, como por exemplo, questões de segurança que não foram abordados nesse trabalho e que fica como uma proposta de trabalho futuro.

Com as análises realizadas pode-se observar uma rede que têm tem potencial para se popularizar, uma vez que o esquema de roteamento parece bem mais eficaz. Mesmo em ambientes de redes muito densas como em MANETs, que em geral são disponibilizadas em ambientes que necessitem de um roteamento diferenciado. O protocolo OLSR corresponde bem a grande demanda da grandes redes móveis, no que diz respeito a encontrar rotas para que os *hosts* possam se comunicar com eficiência. Os resultados obtidos no ambiente de testes que o protocolo OLSR foi exposto, comprovaram a grande capacidade e confiabilidade do protocolo.

6 CONCLUSÃO

Mobilidade e capacidade de automaticamente redefinir suas rotas para os demais nós é uma característica crucial quando necessitamos colocar um *host* em uma grande rede móvel sem fios. Características como esta, de transformar cada *host* da rede em um potencial roteador com certeza fornece a essa rede uma capacidade de auto organização que converge esses tipos de rede a um novo estágio de conectividade, que vai muito além dos modos de operação de redes já conhecidos e difundidos.

Com os testes apresentados nesse trabalho, percebemos que em ambientes sem fio densos como no caso da MANETs e que necessitam de várias alternativas de roteamento como é o caso da necessidade das cidades digitais, rede mesh é uma alternativa interessante. Mostrou-se também que a capacidade dos protocolos de redes *Wi-Mesh* de sempre alocar o *host* dentro de uma rede localizando os nós vizinhos, tornam a rede *mesh* mais inteligente e eficiente. Pode-se perceber também com os resultados obtidos com os testes, que a rede emprega um esquema de roteamento onde, quanto mais nós, mais eficiente a rede se torna, pois mais rotas para outros nós estarão disponíveis. Com o apresentado no trabalho, percebe-se que em redes neste formato, os problemas de escalabilidade não são tão graves. Considerado que hoje este é um dos maiores desafios encontrados em redes sem fio dos atuais padrões IEEE 802.11a/b/g, também conhecidos no jargão da informática como *Wi-Fi*, diminuir a gravidade deste problema é certamente uma grande vantagem.

Dentre os protocolos reativos e proativos mencionados, o OLSR foi testado e colocado a prova, e com isso conseguimos chegar a um resultado satisfatório. Os testes mostraram que o OLSR emprega um roteamento eficiente, inclusive com algumas melhorias como o uso de MPRs, que o torna uma implementação muito mais otimizada. Apesar de o conceito não ser novo, haja

vista que a própria *internet* já utiliza um esquema de roteamento parecido, verificamos que as redes em malha ainda estão apenas começando a engatinhar, mas possuem grande potencial para se popularizar. Cada vez mais percebe-se as vantagens que essas redes podem trazer. Os estudos nos levam a perceber que nos dias de hoje, o que todos querem é mobilidade com conectividade ininterrupta, em alta velocidade e a um custo extremamente baixo. É aí que as redes em malha tendem a se destacar das demais.

Conclui-se que realmente é possível utilizando soluções com *software livre* como o caso do GNU/Linux em conjunto com o protocolo *OLSR* a criação de roteadores altamente robustos, confiáveis, personalizáveis, gerenciáveis e a um custo baixo. São incontáveis a quantidade de outras funcionalidades e serviços que podem ser adicionadas a este roteador, já que existem inúmeros outros serviços disponíveis para GNU/Linux, como *proxies*, DHCPs, NMSs, DNSs, HTTPs, CMSs, LMS, etc, só para citar alguns. É perfeitamente possível a um custo de licenciamento zero e a um custo de hardware muito baixo, disponibilizar dentro de um ambiente de rede das cidades digitais, serviços viáveis tanto financeiramente como operacionalmente.

REFERÊNCIAS

AGHA, K.A.;LASSOUS, I.G.;PUJOLLE, G. Challenges in ad hoc networking: Fourth Annual Mediterranean Ad Hoc. Springer: Springer Ifip, 2006. 419 p.

BRODSKY, I. Wireless Computing: A Manager's Guide to Wireless Networking. New York: John Wiley & Sons Ltda, 1997. 305 p.

CLAUSEN, T.; JACQUET, P. Request for Comments 3626: Optimized Link State Routing Protocol (OLSR). IETF, Outubro 2003. Disponível em: <<http://www.ietf.org/rfc/rfc3626.txt>>. Acesso em: 30 mai. 2011.

CLAUSEN, T.; JACQUET, P.; LAOUITI, A.; MUHLETHALER, P.; QAYYUM, A.; VIENNOT, L.; Projet HIPERCOM: High PERFORMANCE COMMUNICATIONS: Optimized Link State Routing Protocol for Ad Hoc Networks. Disponível em: <<http://hipercom.inria.fr/olsr/inmic2001.ps>>. Acesso em: 15 Mar. 2011.

FIGUEIREDO, F. L. Fundamentos da tecnologia WiMAX. Disponível em: <http://barao.mycpqd.com/file.upload/sas1437_tecnologia_wimax_port_v02.pdf>. Acesso em: 28 Mai. 2011.

FROSI, B.G.; SCHAEFFER, C.A.L. Experimentação do Protocolo OLSR. Disponível em: <www.upf.br/computacao/images/stories/TCs/201001/Bruno_Frosi_b.pdf>. Acesso em: 02 Jul. 2011.

Guia das Cidades Digitais. Dicionário de Termos Técnicos. Disponível em: <<http://www.guiadascidadesdigitais.com.br/site/dicionario/W>>. Acesso em: 31 Mai. 2011.

Guia das Cidades Digitais: Gestão moderna e serviços para a população. Disponível em: <<http://www.guiadascidadesdigitais.com.br/site/pagina/gesto-moderna-e-servios-para-a-populao>>. Acesso em: 28 Nov. 2009.

JARDIM, F.M. Treinamento avançado em REDES WIRELESS: Voip, GSM, *Wi-Fi*, Skipe, Bluetooch, WLAN Access Point, Antenas e Gateway. São Paulo: Digerati, 2007. 128 p.

JOHNSON, D.; Hu, Y.; MALTZ, D. Request for Comments 4728: The Dynamic Source Routing Protocol (DSR)for Mobile Ad Hoc Networks for Ipv4. IETF, Fevereiro 2007. Disponível em: <<http://www.ietf.org/rfc/rfc4728.txt>>. Acesso em: 30 mai. 2011.

JOSGRILBERG, F. B. Guia das Cidades Digitais: Para além da Cidade Digital. Disponível em: <<http://www.guiadascidadesdigitais.com.br/site/pagina/para-alm-da-cidade-digital>>. Acesso em: 28 Nov. 2009.

MAIA, L. P. Arquitetura de Redes de Computadores. Rio de Janeiro: LTC, 2009. 240 p.

Projet HIPERCOM: High PERFORMANCE COMMUNICATIONS: OLSR - Optimized Link State Routing Protocol. Disponível em: <<http://hipercom.inria.fr/olsr>>. Acesso em: 31 Mai. 2011.

Projet HIPERCOM: High PERFORMANCE COMMUNICATIONS: Demonstration of MPR Floating. Disponível em: <<http://hipercom.inria.fr/olsr/mpr-flooding.html>>. Acesso em: Acesso em: 31 Mai. 2011.

Project OLSRD – An Adhoc Wireless Mesh Routing Daemon. Disponível em: <<http://www.olsr.org/>>. Acesso em: Acesso em: 24 Ago. 2011.

ROCHA. M. Guia das Cidades Digitais: Os muitos benefícios das Cidades Digitais. Disponível em: <<http://www.guiadascidadesdigitais.com.br/site/pagina/os-muitos-benefcios-das-cidades-digitais>>. Acesso em: 28 Nov. 2009.

ROSS, J. *Wi-Fi*: Instale, configure e use redes wireless. Rio de Janeiro: Alta Books, 2003. 245 p.

ROYER, E. B.; PERKINS, C. Request for Comments 3561: Ad hoc On-Demand Distance Vector (AODV) Routing. IETF, Julho 2003. Disponível em: <<http://www.ietf.org/rfc/rfc3561.txt>>. Acesso em: 31 mai. 2011.

SCHRODER, C. Redes Linux: Livro de Receitas. Rio de Janeiro: Alta Books, 2009. 557 p.

SOUSA, L. B. Redes de Computadores: Dados, Voz e Imagem. 7ª Edição. São Paulo: Érica, 2004. 482 p.

VEDANA, V. A Internet Como um Serviço Público Municipal. Disponível em: <http://bd.camara.gov.br/bd/bitstream/handle/bdcamara/1122/internet_servico_vedana.pdf?sequence=3> Acesso em: 29 Mai. 2011.

APÊNDICE A - Projeto RMP/Cidade Digital

O projeto RMP/Cidade Digital, é um projeto que visa o estudo, análise e testes de tecnologias, tanto na área de *hardwares* como *softwares*, para a implantação de uma rede municipal de pesquisa, a ser disponibilizada dentro da infraestrutura de uma cidade digital. Os estudos são realizados de maneira a testar ferramentas a serem utilizadas para serviços diversos dentro da rede, e que reduzam o custo de implantação e manutenção, viabilizando o projeto.

São estudadas ferramentas que possam disponibilizar serviços voltados a agregar conhecimento à comunidade, como blogs, wikis, CMS, LMS, bem como serviços públicos em diversas áreas como bibliotecas virtuais, jornais e revistas eletrônicas, de forma que a mesma possa disponibilizar conteúdos de qualidade ao invés de simplesmente “*internet* de graça”, e que seja acessível a toda comunidade.

No que diz respeito aos softwares o projeto prioriza o uso de soluções com *software* livre (*free software*), afim de diminuir despesas com licenças de uso, serem ferramentas robustas e eficientes.

O objetivo do RMP/Cidade Digital, é que com esta rede a comunidade principalmente a docente/discente possa ter acesso a jornais e revistas eletrônicas, realizar pesquisas escolares, frequentar cursos diversos on-line, resolver rapidamente assuntos que dependem de pesquisas na *internet*, utilizando uma infraestrutura com portais, blogs, gerenciadores de conteúdo, onde educadores e pesquisadores poderão disponibilizar conteúdos de interesse da comunidade acadêmica, estudantil e de todos em geral, através de sites voltados para conteúdos técnicos e científicos, e de forma a facilitar a pesquisa para estudantes das mais diversas áreas, construindo conteúdo de forma colaborativa.

Numa cidade digital, espera-se que toda a comunidade tenha acesso aos recursos oferecidos, podendo se conectar a rede a qualquer instante. Conforme pode ser observado na Figura 33 a seguir, no projeto RMP/Cidade Digital, serão disponibilizados dezenas de roteadores *wireless* (pontos de acessos), em pontos estratégicos da cidade. Assim sendo todos os roteadores comunicarão entre si em uma rede *Wi-Mesh* em modo *Ad Hoc*, e através desses roteadores os nós clientes também configurados em modo *Ad Hoc* obterão conectividade a toda infraestrutura da RMP/Cidade Digital, obtendo assim acesso a todos os serviços oferecidos dentro da rede e também a *internet*.

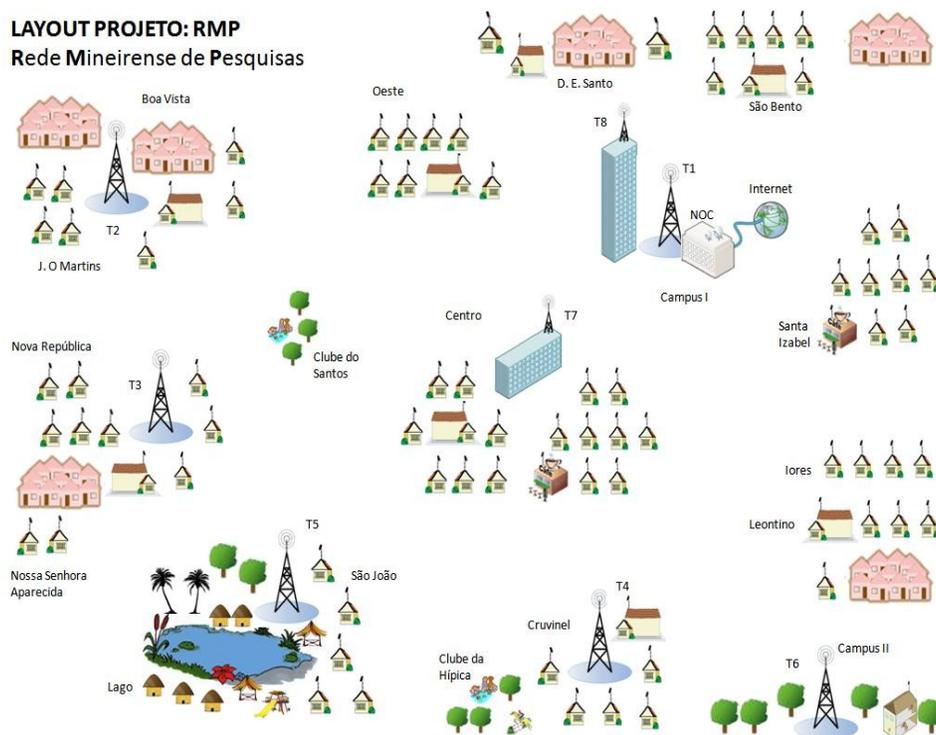


Figura 33 *Layout* da RMP/Cidade Digital
Fonte: RMP/Cidade Digital

Como dito, estes roteadores serão responsáveis pela conectividade e

funcionarão também como *proxies* transparentes, e terão recursos de um *firewall* realizando algumas filtragens de pacotes. Os roteadores terão duas *interfaces* de rede. Uma *interface* na qual será plugada uma antena do tipo omnidirecional 360° para conectividade entre o roteador e todos os outros roteadores bem como entre o roteador e com o nós clientes deste. Outra *interface* de rede na qual será ligada uma antena do tipo direcional para conectividade direta entre o roteador e o NOC da RMP/Cidade Digital. As duas *interfaces* de rede estarão conectadas em modo *Ad Hoc*. O *layout* da rede está representado pela Figura 33.

ANEXO A – Arquivo olsrd.conf Default

```
#
# OLSR.org routing daemon config file
# This file contains the usual options for an ETX based
# stationary network without fisheye
# (for other options see olsrd.conf.default.full)
#
# Lines starting with a # are discarded
#

##### ATTENTION for IPv6 users #####
# Because of limitations in the parser IPv6 addresses must NOT
# begin with a ":", so please add a "0" as a prefix.

#####
### Basic configuration ###
#####

# keep this settings at the beginning of your first configuration file

# Debug level (0-9)
# If set to 0 the daemon runs in the background, unless "NoFork" is set to true
# (Default is 1)

# DebugLevel 1

# IP version to use (4 or 6)
```

```
# (Default is 4)

# IpVersion 4

#####
### OLSRd agent configuration ###
#####

# this parameters control the settings of the routing agent which are not
# related to the OLSR protocol and it's extensions

# FIBMetric controls the metric value of the host-routes OLSRd sets.
# - "flat" means that the metric value is always 2. This is the preferred value
# because it helps the linux kernel routing to clean up older routes
# - "correct" use the hopcount as the metric value.
# - "approx" use the hopcount as the metric value too, but does only update the
# hopcount if the nexthop changes too
# (Default is "flat")

# FIBMetric "flat"

#####
### Linux specific OLSRd extensions ###
#####

# these parameters are only working on linux at the moment, but might become
# useful on BSD in the future

# SrcIpRoutes tells OLSRd to set the Src flag of host routes to the originator-ip
```

```
# of the node. In addition to this an additional localhost device is created
# to make sure the returning traffic can be received.
# (Default is "no")

# SrcIpRoutes no

# Specify the proto tag to be used for routes olsr inserts into kernel
# currently only implemented for linux
# valid values under linux are 1 .. 254
# 1 gets remapped by olsrd to 0 UNSPECIFIED (1 is reserved for ICMP
redirects)
# 2 KERNEL routes (not very wise to use)
# 3 BOOT (should in fact not be used by routing daemons)
# 4 STATIC
# 8 .. 15 various routing daemons (gated, zebra, bird, & co)
# (defaults to 0 which gets replaced by an OS-specific default value
# under linux 3 (BOOT) (for backward compatibility)

# RtProto 0

# Activates (in IPv6 mode) the automatic use of NIIT
# (see README-Olsr-Extensions)
# (default is "yes")

# UseNiit yes

# Activates the smartgateway ipip tunnel feature.
```

See README-Olsr-Extensions for a description of smartgateways.

(default is "yes")

SmartGateway yes

Allows the selection of a smartgateway with NAT (only for IPv4)

(default is "yes")

SmartGatewayAllowNAT yes

Defines what kind of Uplink this node will publish as a

smartgateway. The existence of the uplink is detected by

a route to 0.0.0.0/0, ::ffff:0:0/96 and/or 2000::/3.

possible values are "none", "ipv4", "ipv6", "both"

(default is "both")

SmartGatewayUplink "both"

Specifies if the local ipv4 uplink use NAT

(default is "yes")

SmartGatewayUplinkNAT yes

Specifies the speed of the uplink in kilobit/s.

First parameter is upstream, second parameter is downstream

(default is 128/1024)

```

# SmartGatewaySpeed 128 1024

# Specifies the EXTERNAL ipv6 prefix of the uplink. A prefix
# length of more than 64 is not allowed.
# (default is 0::/0

# SmartGatewayPrefix 0::/0

#####
### OLSR protocol settings ###
#####

# HNA (Host network association) allows the OLSR to announce
# additional IPs or IP subnets to the net that are reachable
# through this node.
# Syntax for HNA4 is "network-address network-mask"
# Syntax for HNA6 is "network-address prefix-length"
# (default is no HNA)
Hna4
{
# Internet gateway
# 0.0.0.0 0.0.0.0
# specific small networks reachable through this node
# 15.15.0.0 255.255.255.0
}
Hna6
{
# Internet gateway

```

```
# 0::          0
# specific small networks reachable through this node
# fec0:2200:106:0:0:0:0:0 48
}

#####
### OLSR protocol extensions ###
#####

# Link quality algorithm (only for lq level 2)
# (see README-Olsr-Extensions)
# - "etx_float", a floating point ETX with exponential aging
# - "etx_fpm", same as ext_float, but with integer arithmetic
# - "etx_ff" (ETX freifunk), an etx variant which use all OLSR
# traffic (instead of only hellos) for ETX calculation
# - "etx_ffeth", an incompatible variant of etx_ff that allows
# ethernet links with ETX 0.1.
# (defaults to "etx_ff")

# LinkQualityAlgorithm "etx_ff"

# Fisheye mechanism for TCs (0 meansoff, 1 means on)
# (default is 1)

LinkQualityFishEye 0

#####
```

```

### Example plugin configurations ###
#####

# Olsrd plugins to load
# This must be the absolute path to the file
# or the loader will use the following scheme:
# - Try the paths in the LD_LIBRARY_PATH
# environment variable.
# - The list of libraries cached in /etc/ld.so.cache
# - /lib, followed by /usr/lib
#
# the examples in this list are for linux, so check if the plugin is
# available if you use windows/BSD.
# each plugin should have a README file in it's lib subfolder

# LoadPlugin "olsrd_txtinfo.dll"
LoadPlugin "olsrd_txtinfo.so.0.1"
{
    # port number the txtinfo plugin will be listening, default 2006
# PIParam "port" "81"
    # ip address that can access the plugin, use "0.0.0.0"
    # to allow everyone
    PIParam "Accept" "127.0.0.1"
}

#####

### OLSRD default interface configuration ###
#####

# the default interface section can have the same values as the following

```

```
# interface configuration. It will allow you so set common options for all
# interfaces.
```

```
InterfaceDefaults {
    # Ip4Broadcast    255.255.255.255
}
```

```
#####
```

```
### OLSRd Interfaces configuration ###
```

```
#####
```

```
# multiple interfaces can be specified for a single configuration block
# multiple configuration blocks can be specified
```

```
# WARNING, don't forget to insert your interface names here !
```

```
Interface "<OLSRd-Interface1>" "<OLSRd-Interface2>"
```

```
{
    # Interface Mode is used to prevent unnecessary
    # packet forwarding on switched ethernet interfaces
    # valid Modes are "mesh" and "ether"
    # (default is "mesh")
```

```
    # Mode "mesh"
```

```
}
```