

UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
MESTRADO EM ENGENHARIA DE TELECOMUNICAÇÕES

Rafael De Tommaso do Valle

**MeshAdmin: uma Plataforma Integrada para Gerência de
Redes em Malha Sem Fio**

Niterói
2011

Rafael De Tommaso do Valle

MeshAdmin: uma Plataforma Integrada para Gerência de Redes em Malha Sem Fio

Dissertação apresentada ao Curso de Pós-Graduação “Stricto Sensu” em Engenharia de Telecomunicações da Universidade Federal Fluminense como requisito parcial para obtenção do Grau de Mestre. Área de concentração: Sistemas de Telecomunicações.

Orientadora: Profa. Débora Christina Muchaluat-Saade, D.Sc.

Niterói
2011

Rafael De Tommaso do Valle

MeshAdmin: uma Plataforma Integrada para Gerência de Redes em Malha Sem Fio

Dissertação apresentada ao Curso de Pós-Graduação “Stricto Sensu” em Engenharia de Telecomunicações da Universidade Federal Fluminense como requisito parcial para obtenção do Grau de Mestre. Área de concentração: Sistemas de Telecomunicações.

Aprovado em Junho de 2011

BANCA EXAMINADORA

Profª. Débora Christina Muchaluat-Saade, D.Sc.
Orientadora
UFF

Prof. Célio Vinicius Neves de Albuquerque, Ph.D.
UFF

Profª. Noemi de La Rocque Rodriguez, D.Sc.
PUC-Rio

Niterói
2011

Aos meus pais, que tanto se esforçaram para que eu
chegasse até aqui.

AGRADECIMENTOS

A todos os meus familiares, em especial aos meus pais, que sempre me apoiaram, em todos os sentidos, para dar prosseguimento a meus estudos.

Aos professores do Laboratório Mídiacom, Débora, Célio, Schara, Carrano e Igor, que contribuíram com esse trabalho com críticas e sugestões, além de todo o esforço que empregaram em melhorar a qualidade do laboratório. Em especial à Professora Débora, sempre dedicada e atenciosa, fundamental na minha trajetória acadêmica.

Aos amigos do MídiaCom, que fizeram do ambiente de trabalho o mais agradável. Em especial a Álvaro, Clayton, Diego, Joacir, Joel e Júlia que contribuíram para o desenvolvimento desse trabalho.

À Fábria, que esteve ao meu lado e me apoiou nos bons e maus momentos ao longo desta jornada.

Às instituições que apoiaram este trabalho, primeiramente à UFF, que tem sido minha segunda casa nos últimos 7 anos. Também a TBE, CAPES, FAPERJ e CNPq que possibilitaram a realização deste trabalho através do fomento à pesquisa.

RESUMO

Redes em malha sem fio (redes mesh) têm surgido como uma alternativa de baixo custo para promover acesso banda larga, entre outras aplicações. Essas redes são compostas por roteadores sem fio e clientes que utilizam tipicamente o padrão IEEE 802.11 no modo ad hoc e formam uma malha onde a informação é transmitida da origem ao destino através de múltiplos saltos utilizando enlaces sem fio.

Existem diversas técnicas e ferramentas propostas na literatura para gerência de redes em malha sem fio. Porém, essas técnicas e ferramentas normalmente focam em um conjunto de requisitos específicos, fazendo-se necessária a utilização de mais de uma delas em conjunto. Este trabalho discute os requisitos de uma plataforma de gerência integrada para redes em malha sem fio, que engloba diversas funcionalidades.

Nesta dissertação também é apresentada MeshAdmin, uma plataforma de gerência que satisfaz a um grupo dos requisitos apresentados. Para avaliar a quantidade de tráfego de monitoramento inserido pela ferramenta e a capacidade da mesma em identificar falhas, são apresentados experimentos realizados em um *testbed* real de rede mesh.

Palavras Chave:

Redes em Malha Sem Fio, Redes Mesh, Gerência de Redes em Malha Sem Fio, Plataforma de Gerência, MeshAdmin

ABSTRACT

Wireless mesh networks (WMNs) are emerging as a low-cost alternative for wideband access. Those networks are composed by wireless routers and clients using the IEEE 802.11 standard in ad-hoc mode. WMNs provide a wireless backbone where information is transmitted using the wireless links through multihop communication.

There are several techniques and tools proposed in the literature for wireless mesh network management. Each of those techniques and tools usually includes just a few functionalities leading to the use of more than one together. This work discusses a set of requirements of an integrated platform for wireless mesh network management.

This work also presents MeshAdmin, a management platform that fulfills a subset of the presented requirements. In order to evaluate MeshAdmin monitoring traffic overhead and its capability of detecting network faults, experiments done in a real mesh testbed are presented.

Keywords:

Wireless Mesh Networks, Wireless Mesh Network Management, Management Tool, MeshAdmin

LISTA DE ACRÔNIMOS

AODV:	Ad hoc On-Demand Distance Vector Routing
AMI:	Acute Myocardial Infarction
AP:	Access Point
API:	Application Programming Interface
AToMS:	AMI Teleconsultation and Monitoring System
ETX:	Expected Transmission Count
ICMP:	Internet Control Message Protocol
IEEE:	Institute of Electrical and Electronics Engineers
IP:	Internet Protocol
LAN:	Local Area Network
MIB:	Management Information Base
ML:	Minimum Losses
MTV:	Mesh Topology Viewer
MTV:	Model Template View
MVC:	Model View Controller
NTP:	Network Time Protocol
OLSR:	Optimized Link State Routing
POE:	Power Over Ethernet
RADIUS:	Remote Authentication Dial In User Service
RAM:	Random Access Memory
ReMesh:	Grupo de Pesquisa em Redes Mesh
ReMoTE:	Rede de Monitoramento para Linhas de Transmissão de Energia
RSSI:	Received Signal Strength Indicator
RTT:	Round-Trip Time
SMS:	Short Message Service
SNMP:	Simple Network Management Protocol
SSID:	Service Set Identifier
SSH:	Secure Shell
SVG:	Scalable Vector Graphics
TBE:	Transmissores Brasileiros de Energia
TCP:	Transmission Control Protocol
TC:	Topology Controller
TTL:	Time To Live
UDP:	User Datagram Protocol

WAN: Wide Area Network

WLAN: Wireless Local Area Network

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO	8
1.1 Objetivos da Dissertação	10
1.2 Organização da Dissertação	10
CAPÍTULO 2 - TRABALHOS RELACIONADOS	12
2.1 Projeto ReMesh	12
2.2 Projeto ReMoTE	13
2.3 <i>Framework</i> Hierárquico para Monitoramento de Redes Mesh	16
2.4 MeshMon	18
2.5 Mesh-Mon	18
2.6 MeshFlow	19
2.7 Abaré	21
2.8 Maya	21
2.9 SCUBA	22
2.10 Mesh Topology Viewer	23
2.11 Ferramentas para controle de acesso de usuários	25
2.12 Comparação dos trabalhos relacionados	25
CAPÍTULO 3 - LEVANTAMENTO DE REQUISITOS PARA A PLATAFORMA INTEGRADA	27
3.1 Coleta de Dados	29
3.2 Armazenamento de Dados	30
3.3 Monitoramento	31

3.4	Controle de Usuários	32
3.5	Configuração da Rede	33
CAPÍTULO 4 - A PLATAFORMA MESHADMIN		36
4.1	Painel de Configuração da Ferramenta	37
4.2	Módulo de Coleta	39
4.2.1	Coleta de Informação dos Nós	39
4.2.2	Coleta de Informação de Enlaces	41
4.3	Módulo de Alerta	42
4.4	Módulo de Armazenamento de Dados	43
4.5	Módulo de Exibição	45
4.5.1	Visualização da Topologia	45
4.5.2	Monitoramento dos Nós	46
4.5.3	Exibição de Alertas	49
4.6	Detalhes da Implementação	50
CAPÍTULO 5 - AVALIAÇÃO DA PLATAFORMA MESHADMIN		55
5.1	Avaliação da Plataforma	55
5.1.1	Desempenho do Módulo de Coleta	55
5.1.2	Funcionamento do Mecanismo de Alertas	58
5.2	Comparação com Trabalhos Relacionados	60
5.3	Limitações da Implementação Atual	61
CAPÍTULO 6 - CONCLUSÃO		62
6.1	Contribuições da Dissertação	63
6.2	Trabalhos Futuros	63
REFERÊNCIAS BIBLIOGRÁFICAS		65

LISTA DE FIGURAS

Figura 1: Topologia típica de uma rede em malha sem fio.	8
Figura 2: Topologia da rede interna instalada no Instituto de Computação da UFF. . .	13
Figura 3: Topologia da rede externa instalada no campus da Praia Vermelha da UFF e vizinhança.	14
Figura 4: Primeira versão do Kit mesh desenvolvido pelo projeto ReMoTE, instalado em uma linha de transmissão em SC.	15
Figura 5: Topologia da rede instalada pelo Projeto ReMoTE ao longo de uma linha de transmissão de energia.	15
Figura 6: Segunda versão do Kit mesh desenvolvido pelo projeto ReMoTE, em instalação no campus da Praia Vermelha na UFF.	16
Figura 7: Visualização de topologia utilizando a ferramenta SCUBA.	23
Figura 8: Captura de tela da ferramenta MTV.	24
Figura 9: Esquema de funcionamento da plataforma MeshAdmin.	37
Figura 10: Captura de tela da interface de configuração.	38
Figura 11: Organização do banco de dados de MeshAdmin.	44
Figura 12: Tela inicial da ferramenta.	46
Figura 13: Captura de tela com a informação de um determinado nó.	47
Figura 14: Definição de parâmetros para criação do gráfico.	49
Figura 15: Exemplo de gráfico gerado pela ferramenta.	49
Figura 16: Captura de tela do frame de medidas ativas.	49
Figura 17: Captura de tela com o resultado de uma medida de vazão TCP com Iperf. .	50
Figura 18: Cenários dos testes.	56
Figura 19: Gráfico com a quantidade de bytes de monitoramento trafegados na rede. .	57

Figura 20: Gráfico do tempo médio para execução das rotinas de monitoramento. . .	57
Figura 21: Rede de testes antes da inserção de falhas.	58
Figura 22: Rede de testes após a inserção de falhas.	59
Figura 23: Arquitetura SNMP [Stallings 1998].	69
Figura 24: Exemplo de troca de mensagens no SNMPv1 [Stallings 1998].	70

LISTA DE TABELAS

Tabela 1: Comparação entre as ferramentas apresentadas.	26
Tabela 2: Resumo dos requisitos para uma plataforma integrada de gerência de redes em malha sem fio.	35
Tabela 3: MIBs e objetos utilizados pelo módulo de monitoramento dos nós.	40
Tabela 4: Mensagens e níveis de alerta.	43
Tabela 5: Comparação das ferramentas apresentadas com MeshAdmin.	60

CAPÍTULO 1 - INTRODUÇÃO

Redes em malha sem fio (redes mesh) têm surgido como uma alternativa de baixo custo para promover acesso banda larga [Muchaluat-Saade et al. 2007], entre outras aplicações. Essas redes são compostas por roteadores sem fio e clientes que tipicamente utilizam o padrão IEEE 802.11 [802.11 2007] no modo ad hoc e formam uma malha onde a informação é transmitida da origem ao destino através de múltiplos saltos utilizando enlaces sem fio. Os clientes se associam à rede através de um dos roteadores mesh, que encaminham suas mensagens ao longo da rede até a saída para a Internet. Em uma rede em malha, podem existir um ou mais roteadores mesh conectando a malha à Internet, esses roteadores são chamados de *gateways*. A Figura 1 ilustra a topologia típica de uma rede mesh.

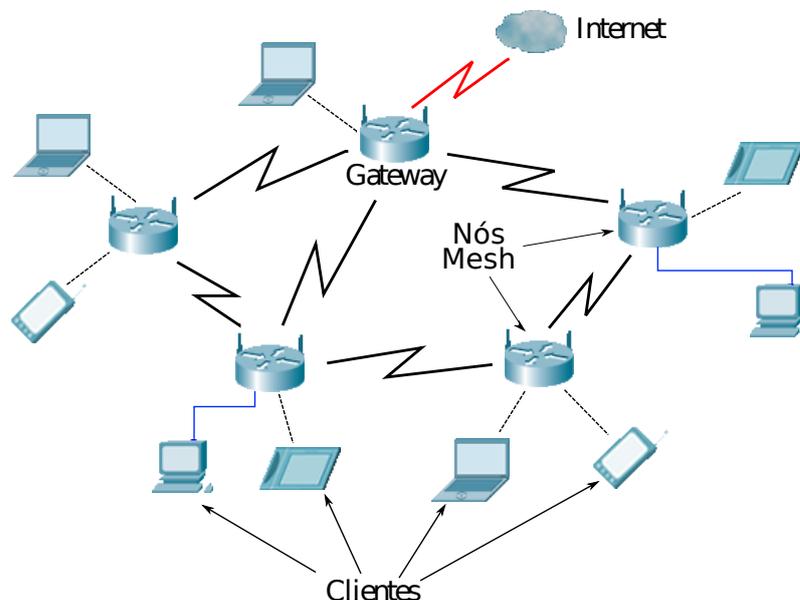


Figura 1: Topologia típica de uma rede em malha sem fio.

Sua utilização é vantajosa principalmente onde não há disponibilidade de infraestrutura cabeada e a utilização de outras tecnologias sem fio, como comunicação via satélite, tem custo de implantação muito elevado. Nessas redes, os clientes obtêm acesso via interface sem fio ou de forma cabeada. Além disso, redes em malha são tolerantes a falhas, pois elas disponibilizam diferentes rotas entre a origem e o destino de uma mensagem [Passos et al. 2006].

Redes mesh utilizam protocolos de roteamento ad hoc e sua topologia é dinâmica. O pro-

protocolo de roteamento descobre o melhor caminho entre os nós da rede sem necessidade de configuração prévia de rotas. Logo, o processo de adição de um nó se faz de maneira simples, sendo apenas necessário habilitar o protocolo de roteamento utilizado no roteador que se deseja incluir na rede em malha.

Nos últimos anos, essa tecnologia tem sido estudada por fabricantes de equipamentos de rede sem fio, operadoras de telecomunicações e instituições acadêmicas [Campista et al. 2008]. Entre suas possíveis aplicações, podemos citar sua utilização no fornecimento de acesso banda larga dentro dos *campi* universitários e vizinhanças e como infraestrutura de rede na construção de cidades digitais, que tem sido um mecanismo dos governos para prover inclusão digital. Outra aplicação é sua utilização como infraestrutura de comunicação para sistemas de telemedicina [Silva et al. 2010]. O projeto AToMS (*AMI Teleconsultation and Monitoring System*) [Teixeira et al. 2009], por exemplo, propõe uma solução para o diagnóstico e o tratamento de infarto agudo do miocárdio através de atendimento remoto, que necessita de uma infraestrutura de comunicação sem fio entre emergencistas e cardiologistas. Essa infraestrutura de comunicação pode ser implementada através de uma rede em malha sem fio metropolitana.

Redes mesh requerem menos gastos com infraestrutura, quando comparadas às soluções de acesso cabeadas, por utilizar apenas roteadores sem fio como infraestrutura de acesso e, ao mesmo tempo, provendo conectividade a uma área bem maior que as redes sem fio tradicionais, que utilizam pontos de acesso (APs – *access points*) no modo infraestruturado.

No entanto, para viabilizar a utilização dessa tecnologia em larga escala é necessária uma plataforma de gerência que facilite a operação e manutenção da rede instalada.

Gerência de redes mesh é uma tarefa mais complexa que gerência de redes cabeadas devido, principalmente, à limitação de recursos na rede e à grande variabilidade de qualidade nos enlaces sem fio [Duarte et al. 2007]. As soluções para monitoramento de redes cabeadas não apresentam desempenho satisfatório quando utilizadas em redes em malha sem fio [Sailhan et al. 2007]. Ainda, há grande diferença na gerência de redes mesh em comparação à gerência de redes sem fio infraestruturadas, principalmente devido à maior confiabilidade e capacidade do *backbone* cabeado em relação ao *backbone* mesh [Nanda e Kotz 2008].

Soma-se a isso o fato de que, atualmente, existem poucas soluções para gerenciar redes mesh disponíveis comercialmente, sendo geralmente ferramentas desenvolvidas exclusivamente para soluções mesh proprietárias. Essas soluções proprietárias têm custos muito elevados, o que pode inviabilizar sua utilização em projetos que possuem recursos limitados, como projetos de inclusão digital e projetos de cidades digitais.

Na literatura, existem algumas técnicas e ferramentas desenvolvidas para auxiliar a gerência de redes em malha sem fio. Normalmente, cada uma delas individualmente não é suficiente para gerenciar uma rede em malha, tornando-se necessária a utilização de outras ferramentas em conjunto. Isso causa certa dispersão das informações, pois cada ferramenta tem sua própria base de dados e/ou interface, e sobreposição de funcionalidades, quando diferentes ferramentas tratam de alguns parâmetros em comum. Pode até ser interessante comparar informações de duas fontes diferentes, porém, essa sobreposição de dados significa maior consumo de memória e CPU nos roteadores e maior tráfego de informação na rede. Considerando que usualmente o hardware de roteadores sem fio tem recursos limitados e os enlaces sem fio oferecem taxa de transmissão muito variável, é desejável que se utilize o mínimo de recursos possível.

1.1 OBJETIVOS DA DISSERTAÇÃO

Este trabalho discute diversas técnicas e ferramentas para gerência de redes em malha sem fio a fim de levantar os aspectos mais relevantes para se obter um diagnóstico claro e preciso do funcionamento de uma rede mesh.

Neste contexto, esta dissertação tem como objetivo definir os requisitos de uma plataforma integrada de gerência de rede em malha sem fio, que unifique as informações coletadas através dos roteadores em uma única base de dados e possibilite a visualização das mesmas através de uma única interface. Os requisitos são definidos de forma que o administrador da rede realize o mínimo de interação possível com a plataforma e não necessite acessar outra ferramenta de gerência para executar as tarefas de administração da rede.

Ainda, esta dissertação tem como objetivo apresentar a ferramenta MeshAdmin, uma plataforma integrada para gerência de redes em malha sem fio que cumpre boa parte dos requisitos apresentados e futuramente os contemplará por completo.

1.2 ORGANIZAÇÃO DA DISSERTAÇÃO

No Capítulo 2, são descritos projetos que utilizam redes mesh como infraestrutura de comunicação sem fio, que se beneficiariam de uma plataforma de gerência integrada. Ainda nesse capítulo, são apresentados os principais trabalhos e ferramentas relacionados à gerência de redes em malha sem fio encontrados na literatura.

No Capítulo 3, é feito o levantamento dos requisitos necessários para o desenvolvimento de uma plataforma integrada de gerência para redes em malha sem fio.

Já no Capítulo 4, é apresentada a arquitetura e implementação da ferramenta MeshAdmin, mostrando as funcionalidades disponíveis na ferramenta.

No Capítulo 5, a plataforma é avaliada em relação à quantidade de tráfego de monitoramento inserido na rede e quanto ao funcionamento do mecanismo de alerta. Além disso, é feita uma comparação com os trabalhos relacionados e são pontuadas as limitações da plataforma.

Finalmente, o Capítulo 6 destaca as contribuições desta dissertação e as considerações finais deste trabalho. Também são apresentadas sugestões para trabalhos futuros.

CAPÍTULO 2 - TRABALHOS RELACIONADOS

Este capítulo apresenta os Projetos ReMesh e ReMoTE, ilustrando cenários práticos em que a proposta desta dissertação pode ser utilizada. Além disso, apresenta diversas ferramentas e outras propostas encontradas na literatura para gerência de redes em malha sem fio.

2.1 PROJETO REMESH

Inspirado em projetos piloto de redes mesh ao redor do mundo, o Projeto ReMesh [Muchaluat-Saade et al. 2007] propôs a implantação de uma rede de acesso do tipo mesh para usuários universitários que residem nas proximidades de suas universidades.

O Projeto ReMesh desenvolveu uma solução mesh de baixo custo utilizando o roteador WRT54G da Linksys, utilizando como *firmware* uma personalização do sistema operacional para sistemas embarcados OpenWrt [OpenWrt 2011]. Cada nó mesh utiliza ainda uma antena – omnidirecional ou direcional, dependendo da localização do ponto – de alto ganho conectada ao roteador utilizando-se cabos coaxiais de baixa perda. Para alimentação dos nós, é utilizado um módulo POE (*Power Over Ethernet*).

Para a comunicação *ad hoc* entre os roteadores, foi utilizado o protocolo de roteamento OLSR-ML [Passos et al. 2006], uma adaptação do protocolo OLSR [Clausen et al. 2003], que utiliza a métrica de roteamento ML (*Minimum Loss*) que seleciona as rotas com menor probabilidade de perda de pacotes entre origem e destino. Para o controle de acesso da rede foi utilizado uma solução aberta de *captive portal* chamada WiFiDog [Lenczner 2005], que fazia autenticação e coletava dados dos usuários conectados à rede. Para monitoramento da rede, duas ferramentas foram desenvolvidas, uma para visualização em tempo real da topologia da rede, chamada de *Mesh Topology Viewer* (MTV) [Valle et al. 2008, Valle e Muchaluat-Saade 2007], descrita na Seção 2.10, e uma ferramenta para coleta de dados sobre os nós da rede. Essa última é responsável por coletar de cada nó da rede as seguintes informações: atraso, vazão, perda de pacotes e número de saltos (todas em relação ao *gateway* da rede), uso de CPU, número de processos ativos, memória disponível e utilizada e bytes trafegados nas interfaces LAN e WLAN. Para obter o diagnóstico da rede em caso de falhas, os administradores acessavam cada ferramenta para buscar informações relevantes. A equipe do projeto sentia a necessidade de

integração das informações das várias ferramentas, o que agilizaria o acesso às mesmas, facilitando o trabalho dos administradores da rede.

Ao longo do projeto foram instaladas duas redes em malha utilizando a solução desenvolvida pelo projeto ReMesh. Uma em um ambiente interno, localizada no Instituto de Computação da UFF. Outra externa, instalada no campus da Praia Vermelha da UFF e vizinhanças, permitindo o acesso via rede mesh pelas comunidades situadas ao redor dos diversos campi da universidade. As Figuras 2 e 3 mostram a topologia das redes interna e externa, respectivamente, geradas pela ferramenta de visualização da topologia. A solução desenvolvida pelo projeto ReMesh foi utilizada em outras redes instaladas em Curitiba, Bélem e Brasília.

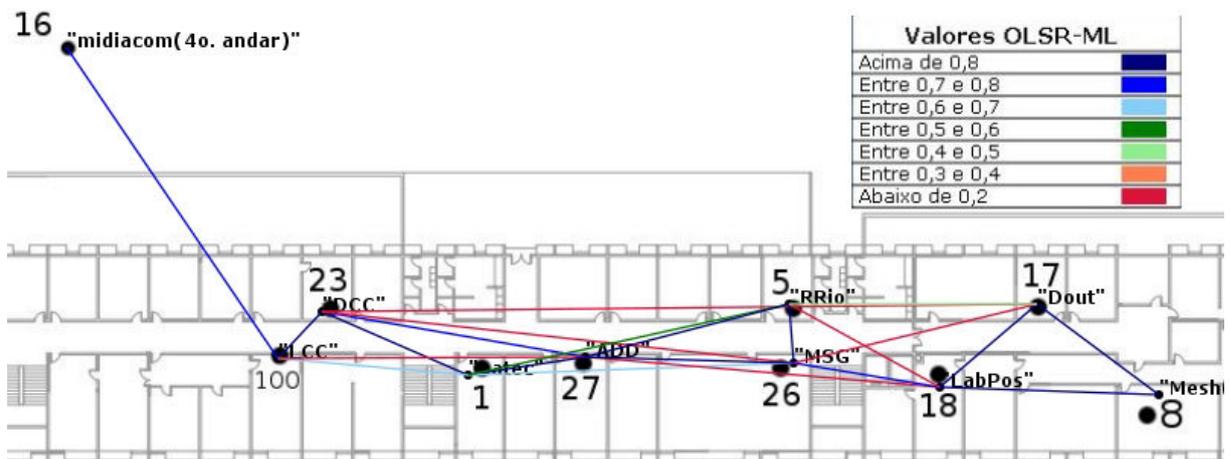


Figura 2: Topologia da rede interna instalada no Instituto de Computação da UFF.

2.2 PROJETO REMOTE

O projeto ReMoTE (Rede para Monitoramento de Linhas de Transmissão de Energia) [Valle et al. 2009, Gerk et al. 2009] propõe a utilização de rede em malha sem fio para o controle e supervisão de linhas de transmissão de energia. O principal objetivo do projeto é oferecer uma solução de comunicação para sistemas de transmissão de energia onde não haja outra alternativa para infraestrutura de comunicação (de dados, voz ou vídeo), como em linhas de transmissão localizadas em áreas rurais.

Geralmente, linhas de transmissão de energia se situam em locais onde não há infraestrutura de comunicação adequada, principalmente em países com extensas áreas de baixa densidade populacional, como o Brasil. Por isso, os transmissores de energia elétrica investem quantias substanciais em sistemas de comunicação, como enlaces microondas e fibras ópticas, para o controle, supervisão e também para prover conectividade entre operadores ao longo da linha de

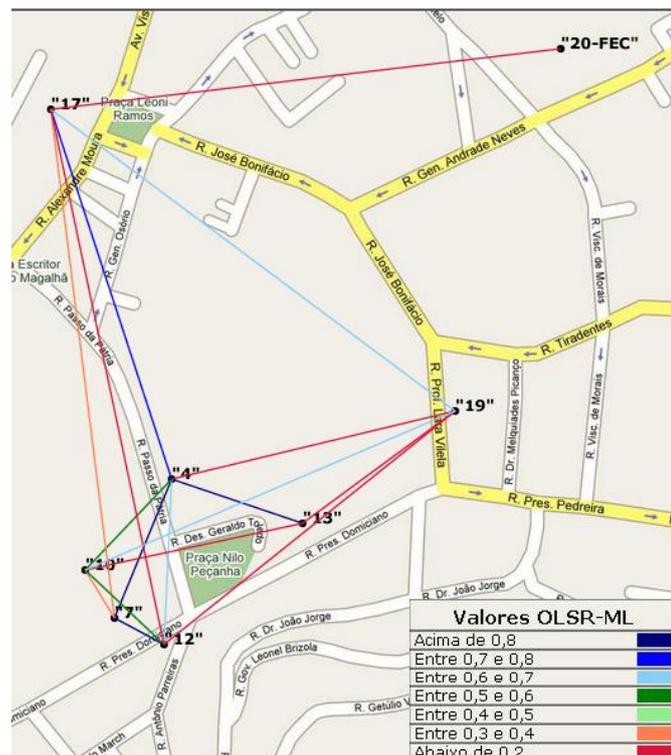


Figura 3: Topologia da rede externa instalada no campus da Praia Vermelha da UFF e vizinhança.

transmissão de energia.

A empresa TBE (Transmissores Brasileiros de Energia), em parceria com o Laboratório MídiaCom da Universidade Federal Fluminense (UFF), tem trabalhado no desenvolvimento de uma solução para infraestrutura de comunicação para supervisão e controle de linhas de transmissão de energia utilizando redes em malha sem fio.

No escopo desse projeto, foi desenvolvida uma solução mesh de baixo custo para ser instalada no topo de torres de transmissão de energia. Essa solução foi desenvolvida de modo que fosse compatível com hardware de “prateleira” como Linksys WRT54g e Ubiquiti Bullet, que são facilmente encontrados no mercado e possuem custo reduzido, porém possuem recursos limitados no que diz respeito a capacidade de processamento, memória RAM e espaço em disco. O *firmware* desenvolvido para esses roteadores é uma personalização do OpenWrt. Cada nó mesh utiliza 2 antenas direcionais, cada uma apontando para um sentido da linha de transmissão para aumentar a capacidade de alcance do sinal sem fio. Como no alto das torres de transmissão de energia não há fonte de alimentação, os nós são alimentados por baterias que, por sua vez, são carregadas por painéis solares. Para chavear esses equipamentos interconectados no sistema de alimentação do roteador, é utilizado um controlador de carga, que evita descarga da bateria abaixo dos níveis aceitáveis e também evita que o roteador receba níveis

indesejados de tensão.

A Figura 4 mostra a solução de um kit mesh, que está instalada na linha de transmissão que liga a Usina Hidrelétrica de Machadinho no Rio Grande do Sul à Subestação de Campos Novos em Santa Catarina. Esta linha possui mais de 40 km de extensão e são utilizados 41 pontos para conectar a rede de ponta a ponta. A Figura 5 ilustra a topologia da rede instalada.

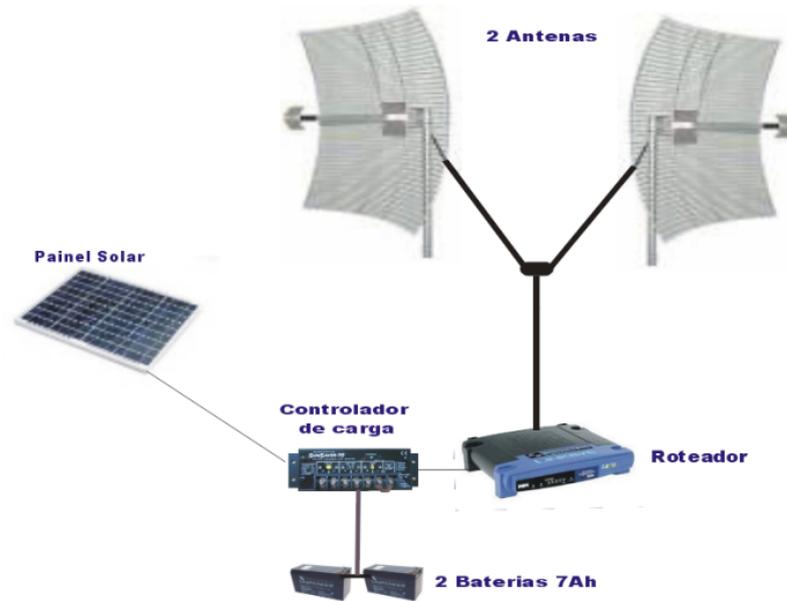


Figura 4: Primeira versão do Kit mesh desenvolvido pelo projeto ReMoTE, instalado em uma linha de transmissão em SC.



Figura 5: Topologia da rede instalada pelo Projeto ReMoTE ao longo de uma linha de transmissão de energia.

O projeto desenvolveu uma segunda versão deste kit promovendo algumas melhorias. Um microcontrolador foi adicionado ao kit. Sua função é substituir o controlador de carga, atuando de forma mais eficiente na gestão dos recursos de energia, e fornecer uma interface de comunicação para sensores. Os sensores podem ser utilizados na própria gerência do kit, como sensores de tensão, corrente e temperatura, como no monitoramento das linhas de transmissão,

como sensores de fumaça (para detecção de incêndios na proximidade da linha), imagem (para acompanhar o crescimento de vegetação nas proximidades das torres) ou acelerômetro (para medir o quanto se movimenta o topo das torres). A nova solução utiliza dois rádios, um para comunicação entre os roteadores, formando um *backbone* sem fio, utilizando o padrão IEEE 802.11a (frequência de 5GHz), e outro para acesso, trabalhando com frequência de 2.4GHz, padrão IEEE 802.11b/g, que pode ser utilizado por técnicos na linha de transmissão e por equipamentos que façam o monitoramento da linha.

A Figura 6 mostra a segunda versão do kit que está sendo instalada em um *testbed* no campus da Praia Vermelha da UFF.

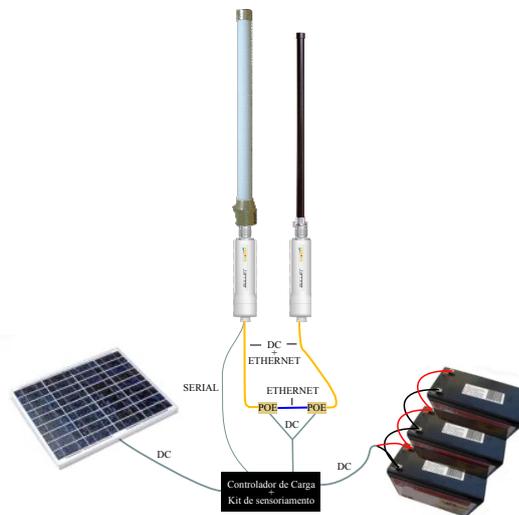


Figura 6: Segunda versão do Kit mesh desenvolvido pelo projeto ReMoTE, em instalação no campus da Praia Vermelha na UFF.

O projeto ReMoTE tem necessidade de utilizar uma plataforma de gerência que permita a manutenção e operação da rede de forma facilitada, já que o acesso aos roteadores mesh é bastante dificultado, por estarem instalados sobre torres de transmissão e pelo fato de a rede em malha possuir um número considerável de roteadores mesh. A proposta apresentada nesta dissertação pode ser usada para gerência das redes em malha do Projeto ReMoTE.

As próximas subseções apresentam diversas propostas encontradas na literatura para gerência de redes em malha sem fio.

2.3 FRAMEWORK HIERÁRQUICO PARA MONITORAMENTO DE REDES MESH

Sailham et al. [Sailhan et al. 2007] tratam do monitoramento de redes em malha sem fio e o dividem em 2 etapas: etapa de medição, na qual o desempenho dos diversos parâmetros

dos elementos da rede são avaliados; e etapa de coleta de dados, na qual os dados medidos são armazenados com o propósito de se obter informações sobre o estado da rede como um todo.

O trabalho propõe uma divisão dos nós em grupos hierárquicos ou *clusters*. Essa divisão é feita com o único propósito de distribuir as informações de monitoramento. Cada grupo possui um *cluster head*, um nó eleito para coordenar e publicar as informações referentes à topologia dos *clusters* e *cluster elements* que estão sob sua supervisão e dos *cluster heads* vizinhos. Sua intenção é reduzir a quantidade de mensagens de monitoramento difundidas pela rede. Ainda, em sua arquitetura de monitoramento, é definido um *Topology Controller* (TC), que consiste em um *cluster head* que é responsável por gerar e manter uma visão local de seus *clusters* e sua conexão com os *cluster heads* vizinhos. Assim, os TCs operam em conjunto para construir a topologia da rede, isto é, a topologia como um todo ou uma determinada subárvore ou nível da rede. A topologia completa consiste na agregação das informações coletadas por todos os TCs de uma determinada topologia. A arquitetura proposta também provê suporte a consultas em tempo real relacionadas à topologia da rede.

Os autores focam o trabalho no monitoramento de uma rede em malha sem fio baseada no padrão IEEE 802.11, utilizando o protocolo de roteamento OLSR (Optimized Link State Routing) [Clausen et al. 2003] com um ou mais *gateways*, fazendo a conexão da rede com a Internet. Para tanto, os autores atribuem aos nós quatro serviços:

- *Medida*: Coleta de informações sobre a topologia da rede obtidas pelo OLSR;
- *Diretório*: Armazena as informações coletadas de cada nó e mapeia os nós de acordo com a estrutura hierárquica definida;
- *Notificação de Eventos*: É utilizada para a comunicação existente no monitoramento e agrupamento dos nós;
- *Agrupamento*: É responsável pela formação dos *clusters*.

O artigo apresenta um *framework* interessante para o monitoramento de redes em malha, principalmente no que diz respeito à coleta e armazenamento de informações. Porém, considerando que o protocolo OLSR é proativo e cada nó mantém uma visão completa da topologia da rede, as informações de topologia podem ser obtidas apenas através de consultas ao protocolo em cada nó. Portanto, essas informações podem ser obtidas através de consulta aos *gateways* da rede, não havendo necessidade da divisão dos nós em grupos menores para o monitoramento da rede.

2.4 MESHMON

Raghavendra et al. apresentam em [Raghavendra et al. 2009] o MeshMon, um *framework* multinível para monitoramento de redes em malha sem fio. Basicamente, o *framework* controla a granularidade das informações coletadas baseada em eventos que ocorrem na rede e dessa forma reduz a quantidade de banda utilizada para o monitoramento da rede.

O trabalho define o conceito de nível de métricas, onde cada nível coleta informações mais detalhadas que o nível inferior. O objetivo do *framework* é diagnosticar os problemas da rede na camada mais baixa possível, ou seja, com o menor nível de detalhe necessário. Caso não seja possível, as camadas superiores são ativadas, sucessivamente, até que o problema seja identificado.

MeshMon possui duas hierarquias de métrica diferentes, uma relacionada ao *backbone* mesh e outra relacionada à conexão de clientes à malha. Quando um parâmetro da hierarquia relacionada à malha ultrapassa seu limite, a árvore de decisão correspondente à malha é ativada. No caso do problema ser detectado na camada de conexão dos clientes, ambas as árvores são ativas, já que o problema pode estar em qualquer uma das camadas.

O artigo avalia a ferramenta em relação à redução de banda para o monitoramento e às falhas detectadas. Seus autores relatam que conseguem detectar 86,6% das falhas, reduzindo o *overhead* de tráfego de monitoramento em 66%. Porém, ainda assim, para um *testbed* de 15 nós mesh, foram gerados 134MB de dados de monitoramento em 4 horas, o que ainda é uma quantidade de informação muito grande trafegando em um *backbone* limitado como o de uma malha sem fio.

2.5 MESH-MON

Nanda e Kotz [Nanda e Kotz 2008] propõem o Mesh-Mon, um sistema descentralizado de monitoramento e gerência de redes mesh multirrádio independente do protocolo de roteamento. Seu principal objetivo é auxiliar o administrador da rede na detecção de falhas e na observação do desempenho da rede.

O administrador da rede se conecta a um dos nós da rede e roda um software em sua estação que recebe relatórios de cada nó e cliente mesh. Esses relatórios podem conter alertas referentes a falhas ou anomalias detectadas. O administrador então se encarrega da análise dos relatórios e toma as medidas necessárias para solucionar os problemas.

Os autores definem três princípios básicos para operação de Mesh-Mon:

1. Cada nó e cliente mesh deve monitorar a si próprio;
2. Cada nó mesh deve monitorar seus vizinhos a k -saltos de distância e manter uma representação detalhada de sua rede local e uma representação superficial da rede como um todo;
3. Cada nó deve fazer parte de uma rede hierárquica, sobreposta à rede de dados, para a propagação das informações de monitoramento.

O sistema é composto ainda por dois softwares: “Mesh-Mon”, instalado nos nós da rede mesh, que gerencia a coleta, comunicação e análise das informações obtidas; e “Mesh-Mon-Ami” (MMA), instalado nos clientes da rede mesh, responsável por auxiliar a disseminação das informações de gerência em áreas desconectadas da rede. Esses softwares possuem três componentes básicos: um sistema para coleta de informação local, que é responsável por medir, coletar e armazenar informações nas três camadas mais baixas (física, enlace e rede) tanto nos nós mesh quanto nos clientes; um sistema para compartilhamento e armazenamento de informação, que é responsável por manter uma visão detalhada das suas informações e de seus vizinhos e alguma informação sobre a rede como um todo e disseminar essa informação, através de transmissão *broadcast*, que será armazenada localmente por seus vizinhos; e um sistema para análise e detecção de falhas, que através das informações coletadas tenta detectar alguma anomalia na rede.

Mesh-Mon é uma boa solução para uma rede em malha sem fio multirrádio. Contudo é necessário que os roteadores mesh possuam certa robustez em relação ao processamento e uma quantidade razoável de espaço em disco para armazenar as informações coletadas. Isso impede sua utilização em equipamentos de baixo custo, como os utilizados nas redes em malha sem fio construídas pelos projetos ReMesh [Muchaluat-Saade et al. 2007] e ReMoTE [Valle et al. 2009].

2.6 MESHFLOW

MeshFlow é definido por Huang et al. [Huang et al. 2007] como um *framework* para monitoramento de redes em malha sem fio baseado no padrão internacional da Cisco para monitoramento de tráfego, o NetFlow [Netflow 2011]. O *framework* é composto de diversos componentes: definição da estrutura de um registro do *MeshFlow*, criação de registro, manutenção de registro, difusão de registro, agregação de registros e análise.

Um registro MeshFlow é um tipo especial de pacote que contém as principais propriedades em comum dos dados trafegados no roteador mesh. Um registro MeshFlow possui os seguintes campos: endereços de origem e destino, endereço do próximo salto, quantidade de bytes, número de pacotes, protocolos de transporte e somatório do tempo de transmissão.

Os registros são criados e armazenados na memória de cada roteador. O tamanho do espaço reservado para armazenar os registros MeshFlow é variável, dependendo principalmente da disponibilidade do equipamento. Quando um pacote passa por um roteador, suas informações de fluxo são capturadas e armazenadas em um registro. Caso haja mais de um pacote com os mesmos endereços de origem, destino e de próximo salto, e o mesmo protocolo de transporte, as informações são consolidadas em um único registro MeshFlow, incrementando o número de pacotes e o tempo de transmissão.

Quando um registro é criado, ele recebe uma marca de tempo indicando o início do registro. Um mecanismo para a verificação do tempo de vida do fluxo é então ativado. Depois de um período predeterminado, os registros são então exportados para um coletor. Os autores não definem um método para a transmissão dos registros MeshFlow, porém a alternativa mais simples seria através do próprio *backbone* mesh, acarretando um *overhead* de tráfego.

Uma vez que os registros de cada nó da rede estão no coletor, é possível mapear o tráfego de toda a rede. Através da análise desses registros, é possível monitorar aplicações, serviços, desempenho, banda utilizada, além de identificar possíveis ações maliciosas e ataques à rede.

O mecanismo apresentado pelos autores é de grande valia para o monitoramento de usuários, nós e aplicações, além de garantir alguma segurança. Para que se tenha um noção mais exata das implicações em relação aos *overheads*, tanto de consumo de memória e CPU dos roteadores, é necessário que sejam realizados experimentos para que seja possível determinar o quão robusto deve ser o equipamento utilizado, a quantidade de bytes trafegados na rede, e com isso averiguar a viabilidade da transmissão dos registros pela própria rede.

Outra ressalva ao mecanismo apresentado é em relação à agregação de registro. Em uma rede em malha, que tipicamente possui uma topologia dinâmica, um mesmo fluxo pode percorrer caminhos diferentes da sua origem ao seu destino. Considerando que o mecanismo só agrega registros que possuem um mesmo endereço de próximo salto, um determinado fluxo que percorrer diferentes caminhos será dividido em registros diferentes. Para uma análise mais precisa, seria interessante agregar registros com endereços de próximo salto diferentes, acrescentando esses endereços no registro, para que se possa identificar os diferentes caminhos utilizados por um mesmo fluxo.

2.7 ABARÉ

Abaré [Pinheiro et al. 2010] é um *framework* para implantação, monitoramento e gerenciamento coordenado e autônomo para redes em malha sem fio desenvolvido para equipamentos que operam no padrão IEEE 802.11 convencionais com alteração de *firmware* para distribuições Linux embarcadas (como OpenWrt ou DD-WRT [DD-WRT 2011]) e fornece uma base teórica para criação de ferramentas de gerência para redes em malha sem fio.

Foi desenvolvido de forma modular e possui em três camadas: Administração, responsável pela interação com o administrador da rede, para mudanças de *firmware*, configuração de sistema e gerência do sistema; Núcleo, a parte central do sistema responsável pela parte lógica e armazenamento de informações, possui funções para obtenção e gestão das informações dos componentes do *framework*; e Camada do Roteador que fornece acesso ao sistema operacional dos nós e adiciona funções autônomicas ao *framework*.

Abaré é um *framework* bem completo no que diz respeito à especificação das diversas fases de gerência de redes em malha sem fio. Porém, muitas funções propostas no *framework* são executadas no roteador o que pode causar um esgotamento de recursos. Na implementação apresentada em [Pinheiro et al. 2010] não foram feitas avaliações neste sentido.

2.8 MAYA

Maya [Manzano et al. 2008] é uma ferramenta para gerência e configuração dos elementos de redes em malha sem fio. Sua arquitetura é baseada na cooperação de três diferentes componentes: uma unidade de gerência, a rede distribuída e os clientes da rede. O primeiro consiste em um ou mais servidores de gerência que abrigam a interface web e a base de dados da ferramenta. A rede distribuída é composta por roteadores mesh, que se comunicam em modo ad hoc através de múltiplos saltos, utilizando o protocolo de roteamento AODV (*Ad hoc On-Demand Distance Vector Routing*) [Perkins et al. 2003]. O servidor de gerência utiliza mensagens UDP (*User Datagram Protocol*) e conexões SSH (*Secure Shell*) para configurar esses roteadores que, por sua vez, fornecem acesso aos clientes sem fio.

Sua implementação é dividida em duas partes: uma interface gráfica e um módulo de configuração. A interface gráfica de Maya é integrada ao OpenWrt de forma que a partir da interface da ferramenta seja possível acessar os roteadores da rede através do console de administração gráfico do próprio OpenWrt. O módulo de configuração é responsável por garantir que os parâmetros modificados na interface gráfica sejam atualizados em cada roteador mesh.

Maya possibilita ao administrador da rede habilitar ou desabilitar nós da rede, o que não desativa a interface de rede sem fio ou desliga o nó, mas apenas bloqueia todo o tráfego de dados passante, exceto o tráfego de controle. Ainda, a ferramenta utiliza um algoritmo para que os parâmetros dos nós mais distantes (em número de saltos) do servidor da rede sejam alterados primeiro. Isso porque a alteração de parâmetros como SSID (*Service Set Identifier*) e frequência de operação da rede quando feita primeiro nos nós mais próximos do servidor deixará os roteadores seguintes inacessíveis via múltiplos saltos.

Maya apresenta uma solução interessante para configuração de parâmetros de redes em malha sem fio. Porém, o algoritmo para alteração automática de parâmetros na rede pode não funcionar como esperado em redes com mudanças frequentes de topologia. A ferramenta calcula a distância em saltos de um nó ao servidor utilizando pacotes ICMP *echo* e verifica seu TTL (*Time To Live*). Se a topologia da rede for alterada, o caminho de um nó ao servidor pode ser alterado. Caso isso ocorra durante o procedimento de atualização dos parâmetros, um determinado nó pode ficar inacessível caso algum nó ao longo do caminho tenha seus parâmetros alterados.

2.9 SCUBA

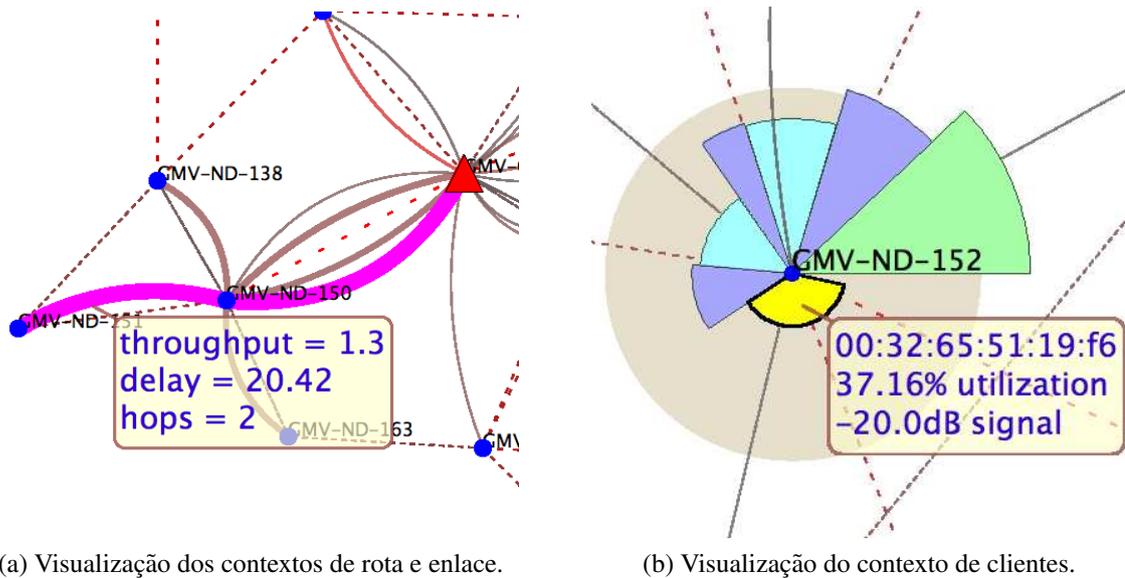
Jardosh et al. [Jardosh et al. 2008] apresentam o SCUBA, uma ferramenta para monitoramento em tempo real de redes em malha sem fio que visa facilitar o diagnóstico de problemas nessas redes. A ferramenta divide as métricas de desempenho em múltiplas camadas, chamadas de contexto, com a finalidade de reduzir as informações apresentadas na tela e de mostrar uma visão geral de um determinado contexto de métricas. Desta forma, o administrador da rede pode focar em uma determinada região, aumentando progressivamente a resolução, navegando por diferentes contextos e assim obter informações mais específicas. A ferramenta possui três contextos de visualização diferentes: contexto de rotas, de enlaces e de clientes.

O contexto de rotas representa a camada mais alta de contextos e mostra apenas os múltiplos saltos entre os nós e seus respectivos *gateways*. Esse contexto utiliza como métricas a vazão de fluxos TCP (*Transmission Control Protocol*) através rotas criadas em cada roteador e o RTT (*Round-Trip Time*) de pacotes UDP nas mesmas rotas.

O contexto de enlaces adiciona mais uma métrica para o diagnóstico de problemas na rede, a métrica ETX (*Expected Transmission Count*) [Couto et al. 2003], que se refere ao número total de transmissões necessárias para um pacote ser encaminhado com sucesso em um enlace sem fio.

O contexto de clientes representa a camada mais baixa de contexto e mostra os seguintes parâmetros: número de clientes associados por roteador, percentual de utilização do canal por cliente, RSSI (*Received Signal Strength Indicator*) de quadros recebidos dos clientes e o nível de interferência externa.

A Figura 7 mostra os diferentes contextos da ferramenta.



(a) Visualização dos contextos de rota e enlace.

(b) Visualização do contexto de clientes.

Figura 7: Visualização de topologia utilizando a ferramenta SCUBA.

SCUBA é uma ferramenta bem completa para a visualização da topologia e identificação de falhas em uma rede em malha sem fio. A ferramenta possui ainda dois modos de visualização: plano (bidimensional) e hiperbólico. Uma limitação é que a ferramenta não captura informações sobre os nós da rede. Dessa forma, o diagnóstico de problemas causados por falhas em algum recurso dos nós da rede, como o esgotamento de memória em um determinado roteador, por exemplo, fica comprometido.

2.10 MESH TOPOLOGY VIEWER

Mesh Topology Viewer (MTV) [Valle et al. 2008, Valle e Muchaluat-Saade 2007] é uma ferramenta para visualização da topologia de redes mesh em tempo real. A ferramenta mostra os nós e a qualidade de cada enlace da rede. Também há uma opção para informação adicional sobre os nós da rede.

O administrador da rede define alguns parâmetros de configuração da visualização da rede, como figura de fundo da topologia, usualmente um mapa ou planta do local de instalação da rede mesh, e *gateway* da rede; e para a configuração dos nós, como IP da interface WLAN,

posição dos nós no mapa e identificação do nó.

A ferramenta MTV busca no *gateway* definido pelo arquivo de configuração as informações sobre a qualidade de cada enlace. Essas informações são obtidas passivamente através do protocolo de roteamento OLSR. Foram desenvolvidas duas versões da ferramenta, uma para a versão padrão do OLSR, utilizando a métrica ETX e outra para o OLSR-ML [Passos et al. 2006], uma customização do OLSR desenvolvida pelo projeto ReMesh [Muchaluat-Saade et al. 2007] que utiliza a métrica *Minimum Losses* (ML).

Após obter as informações do arquivo de configuração e do *gateway*, a ferramenta gera um arquivo no formato SVG (*Scalable Vector Graphics*) [SVG 2003], que pode ser acessado através de um servidor web.

A Figura 8 mostra uma captura de tela da ferramenta.

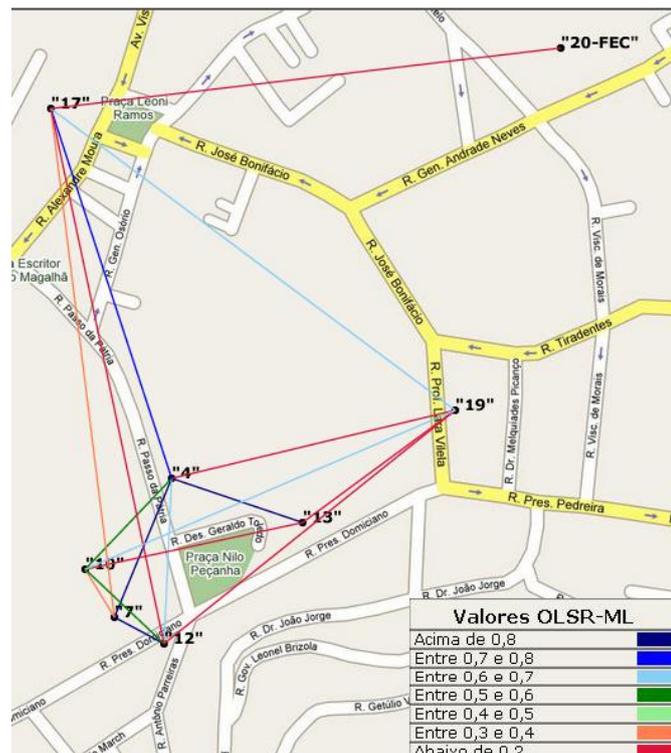


Figura 8: Captura de tela da ferramenta MTV.

Uma das vantagens da MTV é não inserir tráfego adicional para monitoramento da rede, pois cada roteador que utiliza o protocolo de roteamento OLSR tem uma visão completa da topologia. Desta forma, só é necessário acessar o *gateway* da rede. Porém, apenas a visualização da topologia não é suficiente para o monitoramento e identificação de problemas em uma rede em malha.

2.11 FERRAMENTAS PARA CONTROLE DE ACESSO DE USUÁRIOS

O WiFiDog [Lenczner 2005] é uma ferramenta aberta de *captive portal* que possibilita autenticação de usuários e monitoramento centralizado de uma rede sem fio. Outra ferramenta utilizada para este fim é CoovaChilli [CoovaChilli 2011]. O CoovaChilli também é uma ferramenta de controle de acesso de usuários para redes sem fio para *captive portal* e acesso via 802.1X, baseado no projeto já terminado ChilliSpot. Uma das vantagens do CoovaChilli é a possibilidade da utilização de um servidor RADIUS [Rigney et al. 2000] para autenticação dos usuários. Assim, como o WiFiDog, o CoovaChilli também fornece diversas estatísticas sobre os usuários da rede.

O WiFiDog já foi implementado como solução para controle de usuários em redes em malha sem fio no Projeto ReMesh [Duarte et al. 2007]. Apesar de ser desenvolvido para redes sem fio infraestruturadas, a partir de algumas modificações, foi possível utilizar o software no contexto de uma rede em malha. Porém, com o WiFiDog não é possível dar suporte a múltiplos *gateways*. Isso porque a autenticação é feita por *gateway*. Por isso, se o usuário trocar de *gateway* durante a conexão, terá que se autenticar novamente.

Ambas as ferramentas, além de realizar a autenticação dos usuários, também coletam diversas informações sobre os mesmos, como tráfego gerado por cada usuário, tempo de conexão de cada usuário, número de usuários conectados na rede em cada período, entre outras informações. Essas informações são úteis para o monitoramento do uso da rede.

2.12 COMPARAÇÃO DOS TRABALHOS RELACIONADOS

Este capítulo mostrou os trabalhos mais relevantes encontrados na literatura relacionada à gerência de redes em malha sem fio. A Tabela 1 compara as ferramentas e técnicas discutidas neste capítulo.

Os seguintes itens são avaliados na tabela: Coleta de Dados, que é dividida em três subcategorias: Dados de Usuários, referentes ao monitoramento do tráfego de usuários da rede, Dados dos Nós, referentes ao monitoramento dos recursos dos nós da rede, Dados da Rede, referentes ao monitoramento das informações de tráfego e qualidade dos enlaces da rede; Visualização de Topologia; e Monitoramento de Falhas, dividido em duas subcategorias: Detecção, capacidade da ferramenta de detectar alguma falha passivamente, e Notificação, capacidade da ferramenta gerar um alerta após detectar uma falha na rede; Configuração de Nós, que diz respeito à capacidade da ferramenta configurar parâmetros dos nós mesh.

É importante observar que a maioria das ferramentas estudadas não fornece um mecanismo para coleta de informações sobre os nós da rede. Muitos problemas em redes em malha, principalmente quando são utilizados equipamentos de baixo custo, são ocasionados por esgotamento de recursos de um roteador, seja espaço em disco, memória disponível ou uso de CPU. A importância desse requisito será discutida mais detalhadamente no Capítulo 3.

Tabela 1: Comparação entre as ferramentas apresentadas.

	Coleta de Dados			Visualização da Topologia	Monitoramento de falhas		Config. de Nós
	Usuários	Nós	Rede		Detecção	Notificação	
[Sailham et al.]			x		x	x	
MeshMon			x		x	x	
Mesh-Mon	x		x				
MeshFlow	x		x		x		
Abaré		x	x		x	x	x
Maya							x
SCUBA			x	x	x		
MTV			x	x	x		
WiFiDog	x						
CoovaChilli	x						

CAPÍTULO 3 - LEVANTAMENTO DE REQUISITOS PARA A PLATAFORMA INTEGRADA

Este capítulo toma por base os trabalhos relacionados no Capítulo 2, aponta os principais desafios na gerência de redes em malha sem fio e especifica os principais requisitos de uma plataforma integrada para gerência dessas redes.

Como dito anteriormente, gerência de redes em malha é uma tarefa mais complexa que gerência de redes cabeadas ou redes sem fio infraestruturadas [Duarte et al. 2007, Sailhan et al. 2007, Nanda e Kotz 2008]. Duarte et al. [Duarte et al. 2007] apontam os principais desafios em se monitorar uma rede em malha sem fio:

- Limitação de recursos dos equipamentos: Em uma solução de baixo custo, o hardware impõe sérias limitações em relação à capacidade de armazenamento local, memória RAM e capacidade de processamento. Essa limitação de recursos pode dificultar o monitoramento da rede.
- *Backbone* formado por enlaces sem fio: Tipicamente, os nós de uma rede em malha sem fio se comunicam apenas através do meio aéreo. Dessa forma, as mensagens de controle precisam ser enviadas na mesma banda em que trafegam os dados da rede. Por isso, o overhead adicionado pelas mensagens de controle deve ser o menor possível a fim de se utilizar o mínimo possível de banda, dado que enlaces sem fio possuem capacidade limitada.
- Grande variação da qualidade dos enlaces: Os diferentes tipos de interferência a qual estão sujeitos enlaces sem fio resultam em grande variabilidade da qualidade dos enlaces da rede em malha. Essas variações causam alterações nas rotas e podem, inclusive, interromper uma conexão corrente. Com isso, a entrega de informações de monitoramento pode ser afetada.
- Nós fora do alcance físico dos administradores (topo de edifícios, torres, postes etc): Usualmente os nós de uma rede em malha são instalados em locais de difícil acesso e nem sempre é possível uma interação física direta do administrador com o hardware.

A partir destes desafios, os autores sugerem alguns pontos que devem ser considerados no desenvolvimento de uma ferramenta de gerência de redes em malha sem fio:

- **Interação mínima dos administradores:** Uma ferramenta deve simplificar e reduzir ao máximo a necessidade de interação do administrador na execução de alguma tarefa ou busca por alguma informação na rede.
- **Confiabilidade:** No processo de configuração remota de um nó, a alteração de alguns parâmetros pode isolar o nó do resto da rede, como mudança no canal de operação ou SSID da rede. É desejável que os nós possam voltar a sua configuração inicial, caso haja alguma falha no processo de configuração e a conectividade seja perdida.
- **Utilização de disco mínima:** Como os recursos dos elementos da rede são limitados, é importante que a ferramenta não ocupe muito espaço em disco, seja pela instalação de bibliotecas e programas ou pelo armazenamento local de informações de monitoramento.
- **Tempo de execução limitado:** As tarefas referentes a monitoramento não podem interferir no objetivo principal de uma rede em malha sem fio que é oferecer acesso aos clientes da rede. Dessa forma, a ferramenta de monitoramento deve ser simples, consumir pouca memória e processamento. Assim, é possível utilizar equipamentos de baixo custo como solução para redes em malha sem fio.
- **Resistência a falhas:** Alguns nós podem ficar inacessíveis, seja por erros no processo de configuração, seja por falha em algum enlace ou até mesmo falta de energia para alimentar os roteadores. A ferramenta de gerência deve ser capaz de tratar esses problemas de falta de conectividade.

Além dos requisitos descritos acima, existem também requisitos relacionados ao desenvolvimento da plataforma. Já foi relatada a necessidade da ferramenta utilizar o mínimo de recursos nos roteadores. Também é interessante que o administrador da rede possa acessar a plataforma de qualquer estação de trabalho e de qualquer sistema operacional. Dessa forma, é desejável que a plataforma seja instalada em um servidor web, sendo acessada facilmente a partir de qualquer navegador pelo administrador da rede.

No que diz respeito às funcionalidades de uma plataforma integrada de gerência, são definidos os seguintes requisitos, descritos com maior detalhe nas próximas seções:

- coletar e armazenar estatísticas da rede;

- auxiliar no monitoramento da rede, permitindo a visualização da topologia da rede, informando a qualidade dos enlaces sem fio e facilitando a verificação de falhas na rede;
- efetuar o controle de usuários que acessam a rede; e
- facilitar a configuração dos nós da rede.

3.1 COLETA DE DADOS

Uma plataforma de gerência deve ser capaz de coletar dados referentes aos nós e enlaces da rede. Desta forma, será possível observar o comportamento da rede como um todo e com o auxílio das técnicas de monitoramento será possível identificar e reparar falhas.

Cada nó possui um conjunto de parâmetros que necessitam ser observados para que seja possível identificar falhas em seu funcionamento, tais como:

- Uptime (tempo desde a última inicialização);
- Uso de CPU;
- Memória utilizada/disponível;
- Bytes trafegados nas interfaces LAN, WLAN e WAN (apenas para os *gateways*);
- Espaço em disco;
- Informações do rádio, tais como taxa de transmissão, canal de operação, SSID e potência de transmissão;
- Outros parâmetros específicos, como por exemplo, nível de bateria do nó, para roteadores alimentados por algum tipo de energia alternativa, como a solar por exemplo.

A qualidade dos enlaces da rede também é um importante parâmetro a ser verificado, pois, com essa informação, pode-se identificar problemas na instalação dos pontos mesh. Na literatura relacionada, são encontradas diversas referências a métricas para redes em malha sem fio que indicam a qualidade dos enlaces [Campista et al. 2008].

A coleta de informações deve ser feita de forma que tráfego de dados na rede não seja afetado, por isso as mensagens trocadas entre a rede e a plataforma de gerência devem ter tamanho reduzido.

É interessante podermos observar a qualidade dos enlaces sem fio em tempo real, o que exige uma consulta periódica aos elementos da rede. Para que essas consultas não gerem uma sobrecarga muito grande de tráfego de monitoramento, é desejável que essas informações sejam obtidas de forma passiva, ou seja, sem injetar tráfego de monitoramento da rede. Isso é possível em redes em malha sem fio que utilizam protocolos de roteamento proativos, por exemplo o OLSR [Clausen et al. 2003]. Neste caso, o próprio protocolo já dissemina informações sobre a topologia completa da rede. Desta forma, a ferramenta pode consultar diretamente as informações de topologia, sem injetar tráfego adicional na rede, consultando um ou mais *gateways* da rede.

3.2 ARMAZENAMENTO DE DADOS

Uma plataforma deve, ainda, guardar os dados coletados por determinado período, que deve ser especificado pelo administrador da rede. Por exemplo, em uma rede de produção pode não ser necessário manter informações de tráfego da rede por muito tempo, pois, normalmente, a quantidade de dados coletados é muito grande. Obviamente, serão necessários servidores com maior capacidade de armazenamento – e por consequência mais custosos – de informações de gerência à medida que o tempo de armazenamento de dados aumenta. Em uma rede de testes, a quantidade de dados armazenada tende a ser menor, podendo ser armazenada por períodos maiores, caso haja necessidade.

Não é interessante que estes dados sejam armazenados nos roteadores, primeiro devido a sua limitação de recursos já discutida anteriormente e, segundo, caso um nó fique inacessível, não seria possível observar as informações coletadas sobre o mesmo para se obter um diagnóstico. Logo, é necessário que haja um servidor contendo um banco de dados a fim de armazenar as informações coletadas pela plataforma de gerência.

Porém, por um período de tempo reduzido, pode haver necessidade de armazenar as informações coletadas nos nós, para então serem transferidas para a base de dados. Cada nó pode armazenar uma determinada quantidade de informações. Porém, o tempo e quantidade de dados armazenados nos nós devem ser definidos levando-se em consideração o espaço em memória disponível e a quantidade de tráfego que a transferência dessas informações irá gerar na rede. Isto implica um mecanismo inteligente de transferência de dados do nó para o servidor, como o proposto em [Huang et al. 2007].

Esse mecanismo não deve interferir no tráfego de dados dos usuários. Caso seja utilizado este tipo de abordagem, é interessante que se definam classes de tráfego distintas a fim de que

o tráfego de dados coletados não interfira no tráfego dos usuários e vice-versa. Dessa forma, o diagnóstico da rede não será comprometido.

3.3 MONITORAMENTO

Os dados coletados por uma plataforma de gerência devem ser constantemente monitorados para que seja possível a identificação e solução de problemas na rede. As ferramentas de monitoramento devem ser projetadas visando a facilitar ao máximo o trabalho do administrador da rede em se obter o diagnóstico da mesma.

Uma característica essencial para uma plataforma em termos de monitoramento é a possibilidade de visualização da topologia em tempo real [Valle et al. 2008, Jardosh et al. 2008]. A plataforma deve ser capaz de informar ao administrador a qualidade dos enlaces - de acordo com a métrica que lhe for mais conveniente - para que ele possa identificar possíveis falhas.

Muitas vezes as condições do ambiente interferem no desempenho da rede, seja por questões naturais, como ação de chuva e ventos, ou por ação humana, como vandalismo ou construção de obstáculos interferindo em um determinado enlace. Por isso, é interessante que a visão da topologia informe a posição geográfica dos nós facilitando a identificação de um possível problema relacionado ao ambiente em que se encontra o nó, correlacionando os dados de desempenho da rede com dados externos (dados climáticos, por exemplo).

Outra característica importante é a possibilidade de visualizar as informações coletadas dos elementos da rede. Com isso, em caso de alguma anormalidade, o administrador pode consultar séries históricas armazenadas na base de dados e identificar algum padrão de falha, como por exemplo aumento do consumo de memória, esgotamento de espaço em disco, taxa excessiva de perda de pacotes em uma interface etc.

As facilidades de visualização de topologia ou de visualização dos dados coletados podem ser classificadas como monitoramento passivo da rede, porque o administrador deve acessar a plataforma para obter as informações desejadas.

Além do monitoramento passivo, uma plataforma deve ser capaz de enviar ao administrador de rede alertas sobre o (mau) funcionamento da rede, que definimos como monitoramento ativo. A plataforma deve observar os parâmetros coletados e compará-los com limiares definidos pelo administrador da rede e/ou valores anteriormente coletados. Caso haja alguma anormalidade, a plataforma pode gerar uma mensagem de alerta que pode ser enviada de diferentes maneiras (email, SMS, twitter etc) para o administrador.

3.4 CONTROLE DE USUÁRIOS

É esperado que uma ferramenta de gerência de rede sem fio efetue o controle de usuários que acessam a rede, normalmente através do registro dos mesmos em uma base de dados. Essa base de dados deve estar instalada, preferencialmente, fora da rede de acesso sem fio. Ainda é esperado que ela forneça informações sobre os usuários registrados e sobre o perfil de acesso dos mesmos. Essas informações podem ser divididas em dois grupos, informações sobre a conexão do usuário associado e sobre os fluxos gerados por cada usuário.

Sobre a conexão dos usuários, é desejável informar:

- Lista de usuários conectados;
- Nó mesh a qual cada usuário está conectado;
- Endereço IP da conexão do usuário;
- Número de fluxos por usuário;
- Data e hora de início da conexão de cada usuário;
- Data e hora de fim da conexão (se já estiver terminada) de cada usuário;
- Quantidade de bytes trafegados por cada usuário.

Também devem ser coletadas as informações sobre os fluxos gerados por cada usuário, tais como:

- Porta de origem;
- IP de destino;
- Porta de destino;
- Protocolo de transporte;
- Números de pacotes enviados;
- Números de pacotes recebidos;
- Bytes enviados pelo fluxo;
- Bytes recebidos pelo fluxo;

- *Gateway* utilizado pelo fluxo.

A análise de fluxos é importante para o aspecto de segurança, pois através dela é possível identificar tráfego malicioso na rede e que usuário originou ou recebeu esse tráfego. Ainda, essas informações são importantes para identificar pontos de saturação da rede, ou seja, determinados nós que estejam sendo acessados por um número considerável de usuários. A partir desses dados é possível replanejar a rede, remanejando ou adicionando nós de forma a melhorar a qualidade da mesma [Huang et al. 2007].

Ainda, através da análise de fluxos, é possível se obter informações sobre o desempenho da rede. Os fluxos podem ser agregados por enlace a fim de se inferir a banda disponível em cada enlace, por exemplo. Além disso, é possível extrair informações relativas ao roteamento, como quantos saltos ou qual a rota mais utilizada por cada cliente ou nó até o *gateway* da rede.

Além desses requisitos que são atendidos pela maior parte das ferramentas de controle de usuários para rede sem fio infraestruturada, temos ainda requisitos que são específicos de redes em malha sem fio e que são desejáveis para uma plataforma integrada de gerência. Eles são descritos a seguir.

Suporte a Múltiplos Saltos: A comunicação em redes em malha se dá através de múltiplos saltos. É necessário que um usuário, após se registrar na rede, possa se autenticar a partir de qualquer nó.

Suporte a Múltiplos Gateways: Em uma rede em malha sem fio, pode existir mais de um *gateway* (roteador sem fio que conecta a rede à Internet) [Duarte et al. 2008]. Isso implica que a ferramenta de autenticação deve possuir uma base integrada, que seja consultada por todos os *gateways* no momento da autenticação. Dessa forma, o usuário que se registrar através de um determinado *gateway* poderá se conectar à Internet através de qualquer outro.

Autenticação em múltiplas interfaces: Uma das vantagens de redes em malha sem fio é a possibilidade do usuário se conectar à rede através da interface cabeada ou através da interface sem fio. Por isso, é necessário que a ferramenta seja capaz de autenticar usuários em múltiplas interfaces.

3.5 CONFIGURAÇÃO DA REDE

Outro aspecto importante de uma plataforma de gerência da rede é a possibilidade de configuração dos parâmetros da rede. A maior parte dos roteadores sem fio já possuem uma interface onde é possível configurar diversos tipos de parâmetros dos mesmos. Porém, parâmetros

como SSID e canal de operação da rede afetam a rede em malha inteira e para que não houvesse perda de conectividade deveriam ser modificados instantaneamente [Manzano et al. 2008]. Se um roteador da rede tem esses parâmetros alterados, ele não terá mais conectividade com os roteadores que tiverem seus parâmetros ainda inalterados. Dessa forma, alguns roteadores podem ficar inacessíveis pela ferramenta de configuração. A seguir são discutidos três métodos para configuração desses parâmetros críticos para a conectividade da rede.

O primeiro é uma adaptação da proposta do *framework* Abaré [Pinheiro et al. 2010]. Ele possibilita ao administrador da rede carregar uma versão do firmware que será utilizado, aplicando algumas configurações iniciais, via interface cabeada. Porém isso é feito para cada roteador individualmente. Seria interessante que essas configurações iniciais pudessem ser divididas em globais – aplicadas para todos os roteadores da rede – e específicas – aplicadas apenas em um nó –, poupando esforço do administrador de rede.

Dessa forma, os roteadores ainda seriam atualizados via interface cabeada e não haveria o problema de conectividade. Porém, a capacidade de nós configurados de uma vez irá depender da infraestrutura cabeada disponível para a conexão dos roteadores, como switches e cabos de rede. Ainda, este método é limitado a alteração de parâmetros antes da instalação física dos nós, não permitindo que o administrador da rede faça modificações com a rede já instalada.

O segundo método é utilizado pela ferramenta Maya [Manzano et al. 2008]. Seus autores propõem um mecanismo para atualização dos parâmetros a partir dos nós a mais saltos de distância do servidor de gerência. Dessa forma, não haveria problemas de desconectividade durante a modificação de parâmetros. Entretanto, em redes com topologia dinâmica como redes em malha, a ordem dos nós em relação a distância do servidor pode ser alterada e com isso algum nó pode ficar inacessível no momento de sua atualização de parâmetros.

O terceiro método, proposto nesta dissertação, não foi encontrado na literatura relacionada à gerência de redes em malha. Ele consiste na atualização dos parâmetros de forma sincronizada. Uma mensagem de atualização dos parâmetros é enviada aos roteadores com o horário em que essa atualização deve ser feita. Dessa forma, os roteadores teriam seus parâmetros modificados de maneira que não perdessem a conectividade.

Para que esse método funcione é necessário que os roteadores estejam sincronizados, o que pode ser feito utilizando-se o protocolo NTP (*Network Time Protocol*) [Mills 1991]. Ainda, deve ser garantida a entrega da mensagem de atualização para todos os nós selecionados para atualização.

O primeiro método é a forma mais segura para configuração dos parâmetros da rede e pode

ser utilizado em conjunto com o segundo ou o terceiro método, que permitem ao administrador da rede alteração dos parâmetros mesmo depois da rede instalada.

Este capítulo apresentou os principais requisitos necessários para o desenvolvimento de uma plataforma integrada de gerência de redes em malha sem fio. A Tabela 2 mostra um resumo dos requisitos apresentados. Esses requisitos foram definidos baseados no estudo da literatura relacionada e nas necessidades encontradas pelas equipes dos projetos ReMesh e ReMoTE em gerenciar as redes mantidas por esses projetos. O próximo capítulo apresenta MeshAdmin, a plataforma de gerência integrada para redes em malha sem fio proposta nesta dissertação.

Tabela 2: Resumo dos requisitos para uma plataforma integrada de gerência de redes em malha sem fio.

Requisito	Descrição
COLETA DE DADOS	Coletar dados referentes aos nós e enlaces da rede inserindo o mínimo de <i>overhead</i> de tráfego possível.
ARMAZENAMENTO DE DADOS	Armazenamento das informações coletadas em bases de dados externas à rede.
MONITORAMENTO	Possibilitar a visualização da topologia em tempo real e realizar notificação de falhas.
CONTROLE DE USUÁRIOS	Monitorar os fluxos oriundos de cada usuário.
CONFIGURAÇÃO DA REDE	Permitir a configuração automática de parâmetros dos nós da rede.

CAPÍTULO 4 - A PLATAFORMA MESHADMIN

Baseada nos requisitos discutidos, esta dissertação apresenta MeshAdmin, uma plataforma integrada para gerência de redes em malha sem fio [Valle e Muchaluat-Saade 2011]. MeshAdmin foi desenvolvida de forma modular, de modo que seja possível adicionar novas funcionalidades à ferramenta. A plataforma é composta dos seguintes módulos:

- Painel de Configuração da Ferramenta
- Módulo de Coleta
 - Informação de Nós
 - Informação de Enlaces
- Módulo de Armazenamento de Dados
- Módulo de Alerta
- Módulo de Exibição
 - Visualização da Topologia
 - Informações dos Nós
 - Visualização de Alertas

A Figura 9 mostra como esses módulos se relacionam. A plataforma MeshAdmin possui um Painel de Configuração que atualiza, no Módulo de Armazenamento, informações referentes à configuração das redes que serão monitoradas e configura alguns parâmetros do Módulo de Coleta, como interfaces e discos que serão monitorados em cada nó. O Módulo de Armazenamento, além das informações de configuração, também recebe informações do Módulo de Coleta, que reúne informações dos nós e enlaces da rede, e do Módulo de Alerta, a fim de que os registros (*logs*) gerados por este módulo possam ser acessados no futuro. O Módulo de Visualização consulta o Módulo de Armazenamento para obter informações sobre as redes que serão monitoradas, recebe informações do Módulo de Coleta a fim de atualizar as informações de topologia em tempo real e recebe as mensagens do Módulo de Alerta, que serão exibidas na tela. Cada um desses módulos será definido com detalhes nas próximas seções.

- Mecanismo de alimentação do roteador (através de energia solar ou POE).

A Figura 10 mostra a interface de configuração e o formulário para a adição de nós.

The screenshot displays the MeshAdmin interface. On the left, the 'Administration' menu is visible, with the 'Disks' option under 'Nodemonitoring' highlighted with a red box and a red arrow pointing to the configuration form. The form on the right is for configuring a node named 'Lab Pos IC 2'. It contains the following fields:

- Node id:** Lab Pos IC 2
- Ip:** 10.151.38.1
- Latitude:** -22.9048871935
- Longitude:** -43.1310750933
- Disks:** /tmp
- Ifaces:** eth0.0, eth0.1, wlan0
- Network:** Rede Interna
- Power:** Yes

At the bottom of the form, there are buttons for 'Delete', 'Save and add another', 'Save and continue editing', and 'Save'.

Figura 10: Captura de tela da interface de configuração.

Já no grupo *nodeMonitoring*, é possível adicionar discos e interfaces que serão monitorados pela ferramenta. No processo de adição de cada nó serão selecionados discos e interfaces contidos nesta base. Para cada disco adicionado, é passado apenas seu caminho no roteador, já para as interfaces são passadas o tipo de interface (WAN, LAN, WLAN) e o nome da interface no roteador (eth0, por exemplo).

No grupo *topologyMonitoring*, é possível adicionar a métrica utilizada pelo módulo de visualização da topologia. A métrica adicionada deve ser a mesma utilizada pelo protocolo de roteamento, que na implementação é responsável por enviar as informações referentes à topologia da rede. O administrador insere o nome da métrica, uma descrição e intervalos numéricos associados a cores. As cores serão utilizadas para desenhar os enlaces sem fio na visão da topologia, conforme a qualidade do enlace, representada pelo valor da métrica de roteamento. Por fim, o administrador deve associar a métrica a uma rede monitorada pela plataforma. Com essa funcionalidade, é possível utilizar a plataforma MeshAdmin com qualquer métrica de roteamento.

4.2 MÓDULO DE COLETA

4.2.1 COLETA DE INFORMAÇÃO DOS NÓS

Utilizando o protocolo SNMP (*Simple Network Management Protocol*) [Stallings 1998], um processo agente é instalado nos nós da rede e um processo gerente é integrado à ferramenta MeshAdmin. O processo gerente faz requisições em busca das informações que auxiliem o administrador no diagnóstico de problemas nos nós. Mas informações sobre o protocolo SNMP são apresentadas no Apêndice A.

Como o hardware utilizado pelos roteadores pode ser de baixo custo e possui recursos limitados, havia necessidade de se utilizar uma implementação compacta do agente SNMP nos roteadores, de modo que não comprometesse o uso de memória e CPU dos mesmos.

Foram testadas duas implementações do SNMP desenvolvidas para o OpenWrt: Net-SNMP¹ e Mini SNMP Daemon². A primeira praticamente esgotou os recursos de memória. Já a segunda obteve um bom desempenho, não comprometendo os recursos de um roteador. Por isso, o Mini SNMP é usado nos roteadores mesh para comunicação com o servidor MeshAdmin. O Mini SNMP Daemon implementa as versões 1 e 2 do protocolo SNMP. Já a plataforma MeshAdmin utiliza a versão 2 do protocolo para coleta de informações. Entretanto, o Mini SNMP atende a um número limitado de MIBs, ainda que novas MIBs possam ser implementadas com alterações em seu código. Na Tabela 3 são listadas as MIBs utilizadas e suas respectivas variáveis.

Além das variáveis definidas pelas MIBs descritas na Tabela 3, que são implementadas pelo agente SNMP escolhido, também há a necessidade de captar outras informações que não estão definidas em MIBs, como parâmetros referentes ao rádio dos roteadores e parâmetros do sistema de alimentação de energia. Para coletar essas informações, o agente SNMP foi modificado para que o gerente pudesse fazer a coleta desses parâmetros utilizando o protocolo, ao invés de utilizar outros mecanismos de coleta que pudessem não ser tão eficientes, como por exemplo *scripts* de coleta.

Em relação ao rádio, as seguintes variáveis são coletadas:

- Nível de sinal para cada vizinho;
- Taxa de transmissão para cada vizinho; e

¹<http://www.net-snmp.org/>

²<http://freshmeat.net/projects/minisnmpd>

Tabela 3: MIBs e objetos utilizados pelo módulo de monitoramento dos nós.

MIB	Objeto	Descrição
HOST-RESOURCES-MIB	hrSystemUptime	O tempo decorrido desde que o elemento foi ligado.
UCD-SNMP-MIB	memTotalReal	Quantidade total de memória física no elemento.
	memAvailReal	Quantidade disponível de memória física no elemento.
	dskPath	Caminho onde o disco está montado.
	dskTotal	Tamanho total do disco ou partição.
	dskAvail	Espaço disponível em disco.
	dskUsed	Espaço utilizado em disco.
	laLoad	Média da carga de CPU nos últimos 1, 5 e 15 minutos (cada valor é recebido em uma linha).
IF-MIB	ifNumber	Número de interfaces de rede encontradas no elemento.
	ifIndex	Índice numérico único atribuído a uma interface
	ifDescr	Nome da interface de rede (por exemplo: eth0, wlan0 etc).
	ifOperStatus	Status da interface (Ativa ou Inativa).
	ifInOctets	O número total de octetos recebidos pela interface.
	ifInUcastPkts	Número de pacotes <i>unicast</i> recebidos na interface.
	ifInDiscards	Número de pacotes recebidos que foram descartados mesmo sem erro detectado.
	ifInErrors	Número de pacotes recebidos que contêm algum erro.
	ifOutOctets	Número de octetos transmitidos pela interface.
	ifOutUcastPkts	Número de pacotes <i>unicast</i> transmitidos pela interface.
	ifOutDiscards	Número de pacotes que seriam transmitidos mas foram descartados mesmo sem erro detectado.
	ifOutErrors	Número de pacotes que não puderam ser transmitidos por conter algum erro.

- Potência de transmissão do rádio.

Em relação ao sistema de alimentação, as seguintes variáveis são coletadas:

- Tensão no painel solar;
- Corrente gerada pelo painel solar;
- Tensão nas baterias;
- Tensão no roteador;
- Corrente que alimenta o roteador;
- Temperatura interna da caixa que armazena o kit;
- Temperatura externa; e
- Luminosidade.

4.2.2 COLETA DE INFORMAÇÃO DE ENLACES

O Módulo de Coleta de Informação de Enlaces é responsável por obter os endereços de origem e destino de todos os enlaces ativos na rede e a respectiva qualidade, indicada pela métrica de roteamento, de cada um desses enlaces.

A plataforma de gerência se comunica periodicamente com o *gateway* da rede em busca das informações sobre a qualidade dos enlaces da rede. A periodicidade varia de acordo com as necessidades do administrador da rede e pode ser configurada através do Painel de Configuração.

Essas informações são obtidas através de consultas ao protocolo de roteamento OLSR-ML [Passos et al. 2006], uma adaptação do OLSR [Clausen et al. 2003], que utiliza a métrica de roteamento ML (*Minimum Losses*). O OLSR é um protocolo de roteamento proativo, desenvolvido para redes sem fio em modo *ad hoc*, baseado em estado de enlace. Em redes que utilizam OLSR, cada nó mantém uma visão completa da topologia da rede, ou seja, cada nó possui em sua tabela de roteamento todos os nós atingíveis – nós com quem o nó em questão tem conectividade – e a qualidade de cada enlace da malha conectada. Por isso, é possível obter as informações de qualidade dos enlaces consultando apenas um nó da rede. Na implementação de MeshAdmin, os *gateways* da rede são consultados. Dessa forma, para obtenção dessas informações, não há inserção de nenhum tráfego adicional na malha sem fio.

Para obter os valores de cada enlace, é instalado um *plugin* desenvolvido para o OLSR chamado *Dot Draw Plugin*, que escuta por requisições em uma determinada porta (configurada no próprio arquivo de configuração do OLSR). O módulo de visualização por sua vez, faz requisições aos *gateways* na porta definida. O *plugin* fornece as seguintes informações sobre cada enlace: endereços IP de origem e destino e qualidade do enlace.

Na implementação atual de MeshAdmin, desenvolvida para o protocolo de roteamento OLSR, é utilizada a métrica de roteamento ML (*Minimum Losses*) que mede a qualidade do enlace através da probabilidade de sucesso da transmissão de um pacote. Porém, o administrador pode configurar o protocolo OLSR com a métrica que desejar - alterando seu arquivo de configuração nos roteadores da rede - e configurar a métrica escolhida na plataforma através do Painel de Configuração.

Os dados coletados são repassados ao módulo de armazenamento e ao módulo de visualização da topologia. Dependendo das necessidades do administrador, a periodicidade com que as informações são coletadas pode ser configurada para cada módulo.

Usando a configuração *default*, os dados são repassados ao módulo de armazenamento de hora em hora enquanto que para o módulo de visualização de topologia são repassados de 10 em 10 segundos. Essa diferença se deve à necessidade da exibição da topologia em tempo real pelo módulo de visualização. Logo, faz-se necessária a coleta de informações com periodicidade alta. Já para alimentação do Módulo de Armazenamento, essa alta periodicidade pode não ser tão interessante, pois será necessário armazenar uma grande quantidade de informação na base de dados. Para que não haja um esgotamento dos recursos de armazenamento, é interessante que se utilizem períodos de tempo maiores.

4.3 MÓDULO DE ALERTA

O Módulo de Alerta tem como objetivo notificar o administrador sobre eventuais problemas na rede no momento da coleta de informações. Com isso, o administrador pode ser alertado prontamente sobre alguma falha na rede que necessite de reparo imediato sem precisar observar as informações de cada nó.

Esses alertas estão divididos em 3 níveis: Crítico, Aviso e Informação. O nível Crítico é responsável por identificar falhas nos nós e enlaces, como nós desligados ou queda de enlaces. O nível Aviso alerta sobre falhas na configuração da ferramenta, como nós que estão na tabela de roteamento do *gateway*, mas não estão configurados na ferramenta ou alguma falha na coleta de dados. Já o nível Informação deve fornecer informações sobre a execução dos procedimentos da ferramenta, como a coleta de informações dos elementos ou aparecimento de novo enlace na topologia. Considerando que a topologia de redes em malha é dinâmica, é interessante informar ao administrador o surgimento de novos enlaces. A Tabela 4 mostra as mensagens exibidas em cada nível de alerta.

Tabela 4: Mensagens e níveis de alerta.

Nível	Mensagem	Função
Crítico	<i>Node not reached.</i>	Informa quando um nó configurado na plataforma não foi alcançado pela rotina de coleta de dados.
	<i>Connection refused by Gateway.</i>	Informa quando a ferramenta não consegue acessar o <i>gateway</i> para obter as informações de topologia.
	<i>Metric is not configured.</i>	Informa que a métrica de qualidade não foi configurada.
Aviso	<i>Interface is down.</i>	Informa quando uma interface está inativa, acompanhado do nó e da interface.
	<i>Interface set in router but not set in MeshAdmin.</i>	Informa quando existe uma interface no roteador que não foi configurada para ser monitorada na ferramenta. Informa o nó e a interface.
	<i>Disk set in router but not set in MeshAdmin.</i>	Informa quando existe um disco no roteador que não foi configurado para ser monitorado na ferramenta. Informa o nó e o disco.
	<i>Nodes found in the route table but not configured.</i>	Informa o aparecimento de nós na tabela de roteamento do <i>gateway</i> que não estão configurados na ferramenta.
Info	<i>New link.</i>	Informa o aparecimento de um novo enlace, acompanhado dos nós interligados por ele.
	<i>Data collection round complete.</i>	Informa quando foi realizada a coleta de dados e os nós ativos.

4.4 MÓDULO DE ARMAZENAMENTO DE DADOS

As informações obtidas pelo Módulo de Coleta são reunidas no Módulo de Armazenamento de MeshAdmin. Ainda, este módulo recebe as informações adicionadas pelo administrador da rede através do Painel de Configuração e armazena as mensagens geradas pelo Módulo de Alerta.

O Módulo de Armazenamento é composto basicamente por um banco de dados relacional implementado com Postgres³, estruturado de forma que seja possível armazenar as informações coletadas e fornecê-las para os diversos módulos da ferramenta. A organização do banco de dados de MeshAdmin é exibida na Figura 11.

³<http://www.postgresql.org>

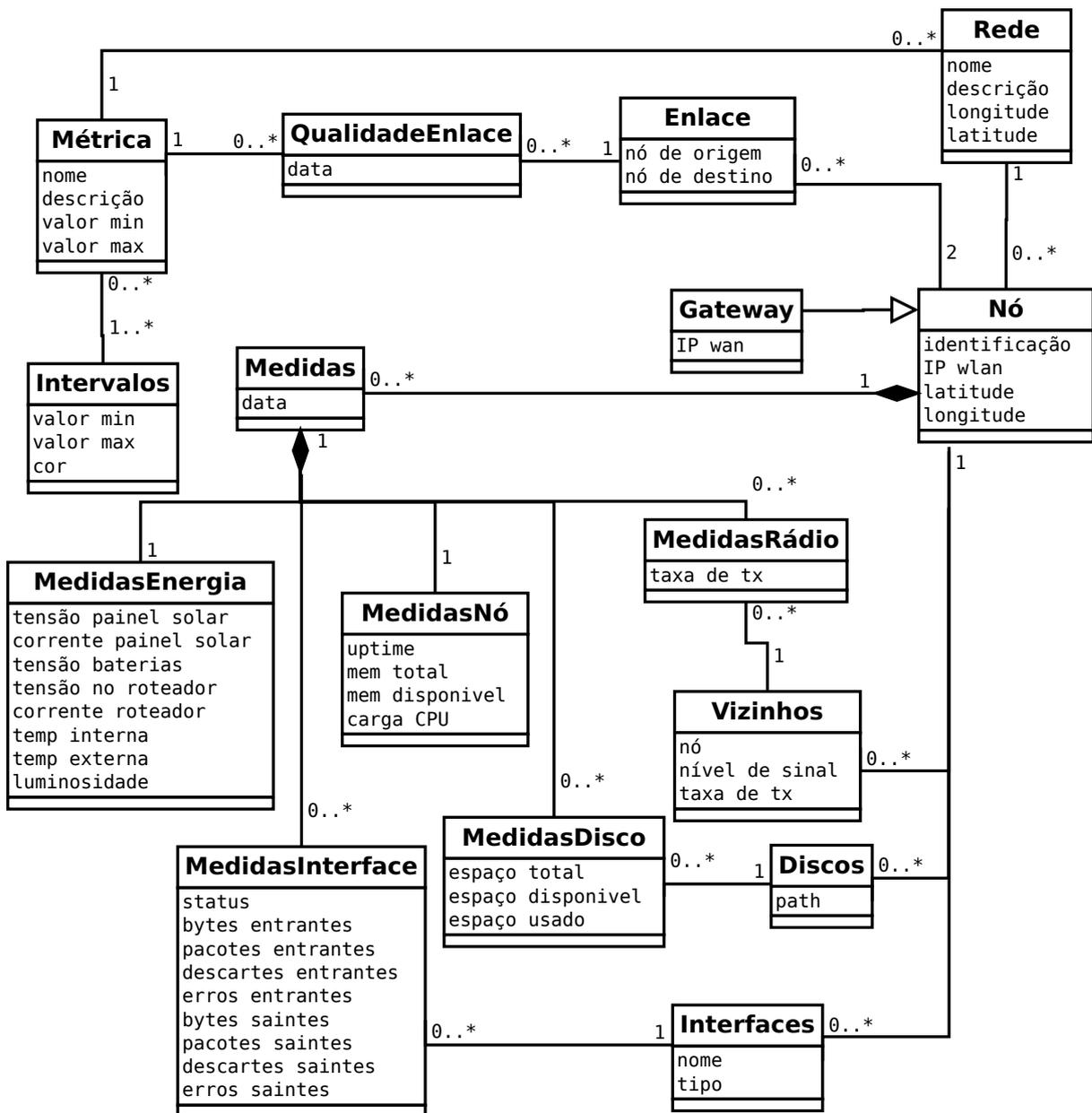


Figura 11: Organização do banco de dados de MeshAdmin.

A base de dados da plataforma armazena as informações sobre os nós e redes que são monitorados. Para cada nó monitorado são armazenadas sua identificação, seu endereço IP da interface sem fio e suas coordenadas geográficas. Caso esse nó seja um *gateway*, também é armazenado o endereço IP de sua interface WAN. Cada nó deve pertencer a uma rede, que por sua vez, possui um nome, uma descrição e coordenadas geográficas.

Os nós mesh possuem um conjunto de recursos que serão monitorados pela plataforma. Eles são divididos em:

- Medidas do Nó: armazena o *uptime*, memória RAM total e disponível e carga de CPU;

- Medidas de Interface: para cada interface, armazena os bytes, pacotes, descartes e erros na transmissão e recepção;
- Medidas de Disco: para cada disco, armazena o espaço total, disponível e utilizado;
- Medidas de Energia: armazena todos os parâmetros coletados pelo Módulo de Coleta dos Nós referente à parte de energia;
- Medidas do Rádio: armazena a taxa de transmissão do rádio e para cada vizinho ou cliente, sua taxa de transmissão e nível de sinal;

Finalmente, os enlaces são formados por dois nós, um de origem e outro de destino. Cada enlace possui uma qualidade de acordo com uma métrica que também é armazenada na base de dados. A métrica possui um conjunto de intervalos numéricos que são associados a uma cor. A cada rede deve ser associada uma métrica utilizada pelo protocolo de roteamento.

4.5 MÓDULO DE EXIBIÇÃO

Para facilitar o monitoramento remoto, MeshAdmin oferece uma interface web, desenvolvida utilizando o *framework* de desenvolvimento em Python para web chamado Django [Django 2011].

A tela inicial da ferramenta MeshAdmin possui 3 *frames*: visualização da topologia, informação de redes e nós e mensagens de alerta. Além disso, a tela inicial possui um *link* para o painel de configuração da ferramenta. A tela inicial e a interface de configuração só podem acessadas pelos administradores após autenticação na ferramenta. A Figura 12 mostra uma captura da tela inicial de MeshAdmin.

4.5.1 VISUALIZAÇÃO DA TOPOLOGIA

Um dos principais requisitos que devem ser atendidos por uma plataforma de gerência é a possibilidade de visualização da topologia da rede em tempo real. O módulo de visualização de MeshAdmin tem como principal objetivo mostrar a topologia da rede em tempo real, com os nós da rede posicionados geograficamente em um mapa e os enlaces da rede coloridos de acordo com sua qualidade (vide Figura 12).

As informações sobre cada enlace são coletadas pelo Módulo de Coleta. Então, elas são processadas por um programa em Python que recebe os endereços IPs de origem e destino e

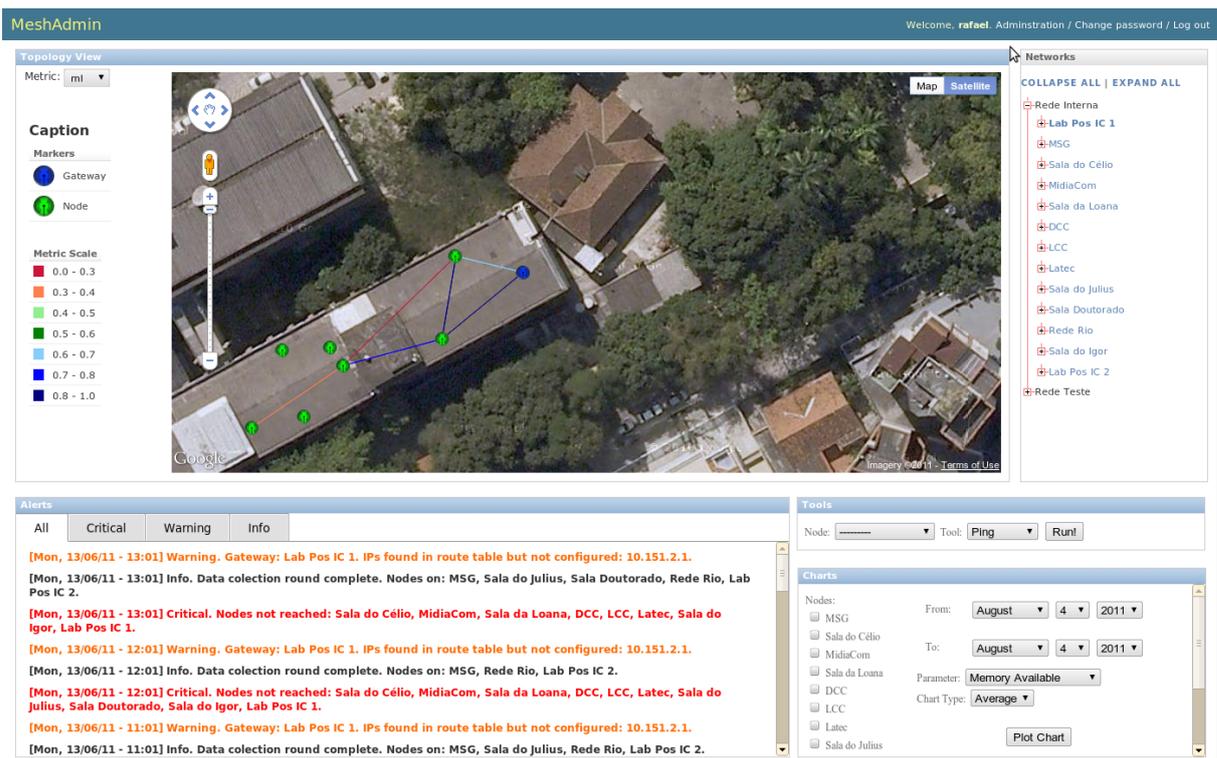


Figura 12: Tela inicial da ferramenta.

busca as informações referentes à posição geográfica na base de dados. As informações de enlace e de posicionamento geográfico dos nós são então processadas e apresentadas utilizando a API Javascript do Google Maps⁴. A integração com essa API permite adicionar funcionalidades de zoom e navegação pelo mapa à interface, o que facilita ainda mais a visualização da topologia. Nessa interface, os enlaces são desenhados na tela e coloridos de acordo com sua qualidade, em correspondência a uma escala de cores que relaciona cor e qualidade. Essa escala de cores pode ser configurada no Painel de Configuração na seção *Metrics* de acordo com a métrica utilizada pelo protocolo de roteamento. À esquerda do *frame* de visualização da topologia está localizada a legenda que relaciona as cores ao intervalo numérico que representa a qualidade. Na parte superior, há um menu onde o usuário pode então selecionar a legenda correspondente à rede que está sendo observada no momento.

4.5.2 MONITORAMENTO DOS NÓS

No lado direito da tela principal (vide Figura 12), uma árvore tem como objetivo facilitar a navegação do administrador por redes e nós. A partir dessa árvore, o administrador da rede pode centralizar o nó que desejar na visão da topologia, apenas clicando em uma das entradas

⁴<http://code.google.com/intl/pt-BR/apis/maps/documentation/javascript/>

da árvore. A partir da árvore também é possível acessar as informações de latitude, longitude e endereço IP da interface WLAN de um nó específico. Ainda, há um *link* disponível para acessar as informações coletadas do nó.

A Figura 13 mostra uma captura da tela da ferramenta mostrando as informações de um nó selecionado na árvore. Os dados que são coletados nos nós estão divididos em 5 grupos de recursos: Recursos dos Nós, das Interfaces, dos Discos e do Sistema de Alimentação. Os recursos dos nós são *uptime*, memória total, memória disponível e carga de CPU. Para cada interface configurada, são coletados os seguintes parâmetros: bytes trafegados, pacotes trafegados, perdidos e descartados, saintes e entrantes. Para cada disco configurado, são coletados o espaço total, disponível e utilizado. Já em relação ao sistema de alimentação são exibidos a tensão no painel solar, corrente gerada pelo painel solar, tensão nas baterias, tensão no roteador, corrente que alimenta o roteador, temperatura interna da caixa que armazena o kit, temperatura externa e luminosidade. Na implementação atual, os dados são coletados a cada hora. Porém o administrador pode configurar o tempo de coleta de acordo com sua necessidade.

Node Resources		Disks	
System		/tmp	
Uptime	0 days, 4:51:5	Total Space	6.62 MB
Total Memory	13.24 MB	Available Space	6.58 MB
Available Memory	1.20 MB	Used Space	44.00 KB
Load	0.06, 0.05, 0.0		

Interfaces		
wlan0	eth0.1	eth0.0
Status UP	Status UP	Status UP
Packets In 232365	Packets In 5288769	Packets In 0
Discards In 0	Discards In 0	Discards In 0
Errors In 0	Errors In 0	Errors In 0
Bytes In 7.83 MB	Bytes In 702.97 MB	Bytes In 406.00 B
Packets Out 66130	Packets Out 5258310	Packets Out 1
Discards Out 0	Discards Out 0	Discards Out 0
Errors Out 0	Errors Out 0	Errors Out 0

Power System Resources						
Vps	Ips	Vbat	Irot	Text	Tint	Luz
12.4 V	420 mA	11.7 V	238 mA	21 °C	27 °C	0.79

Figura 13: Captura de tela com a informação de um determinado nó.

O administrador pode fazer consultas a diversas informações sobre os nós da rede, armazenadas pelo Módulo de Armazenamento, através da ferramenta para geração de gráficos disponível no canto inferior direito da ferramenta (vide Figura 12).

Na implementação atual de MeshAdmin é possível a geração de gráficos dos seguintes parâmetros:

- Bytes entrantes e saídas das interfaces LAN e WAN;
- Memória disponível e utilizada;
- Espaço em disco disponível e utilizado;
- Carga de CPU;
- Tensão nas baterias do sistema de alimentação;
- Corrente gerada pelo painel solar;
- Corrente que alimenta o roteador;
- Temperatura externa e interna do sistema de alimentação;
- Luminosidade.

O administrador fornece, através da interface, as datas inicial e final do período de observação, o parâmetro que será observado, os nós que serão observados e o tipo de gráfico que será gerado. MeshAdmin possibilita a criação de um gráfico de barras contendo a média dos valores armazenados durante o período de observação ou o valor absoluto durante período de observação. Isso depende se o parâmetro crescerá indefinidamente ao longo do tempo, como a quantidade de bytes trafegados por exemplo, ou varia dentro de um certo limite, como a quantidade de memória disponível. Cada barra no gráfico gerado faz referência a um nó selecionado na interface.

A Figura 14 mostra o processo de criação de um gráfico para observar a média da memória utilizada pelos nós “Lab Pos IC 2”, “Sala do Igor”, “Rede Rio”, “Sala do Julius”, “Sala Doutorado”, “Latec”, “LCC”, “DCC” e “Sala da Loana”, entre 12 e 19 de abril de 2011. Já a Figura 15 mostra um exemplo de gráfico de barras com a memória disponível nesses nós.

Ainda, a plataforma possui um mecanismo simples para a realização de medidas ativas na rede. No canto direito da tela (vide Figura 12) há um *frame* chamado de *Tools*. Neste *frame* é possível executar testes com as ferramentas Iperf⁵, para medição de vazão TCP, e Ping, para medição do RTT (*Round Trip Time*). Na implementação atual, é possível apenas medir estes parâmetros de um determinado nó ao servidor de gerência. A Figura 16 mostra uma captura de

⁵<http://iperf.sourceforge.net/>



Figura 14: Definição de parâmetros para criação do gráfico.

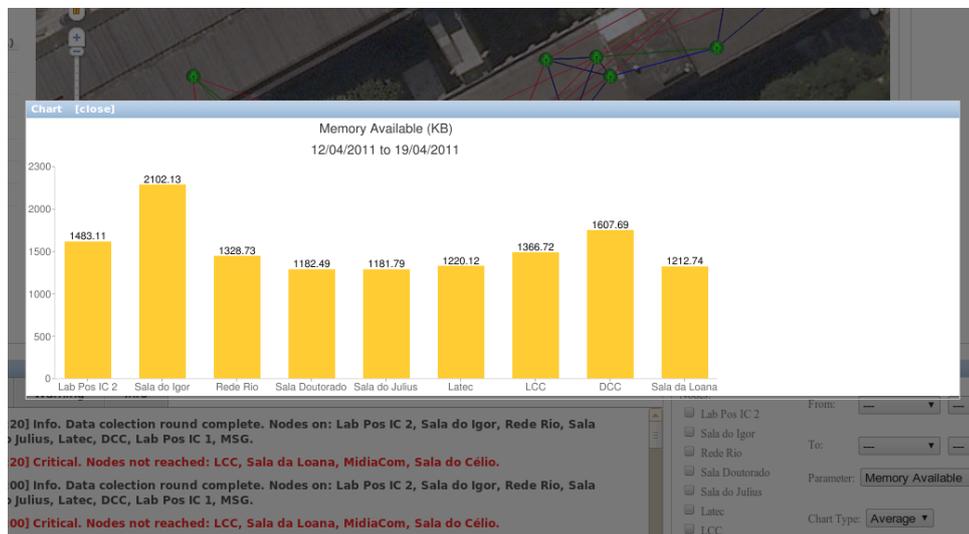


Figura 15: Exemplo de gráfico gerado pela ferramenta.

tela com o *frame* para seleção do nó e da ferramenta que será utilizada. Já a Figura 17 mostra o resultado de uma medida de vazão TCP feita com a ferramenta.

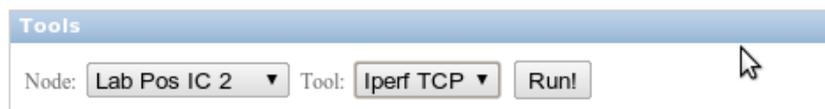


Figura 16: Captura de tela do frame de medidas ativas.

4.5.3 EXIBIÇÃO DE ALERTAS

A tela inicial da plataforma MeshAdmin apresenta, na parte inferior, diversas mensagens de alerta geradas pela ferramenta (vide Figura 12).

As mensagens são apresentadas em 4 abas diferentes. A primeira aba, denominada *All*,

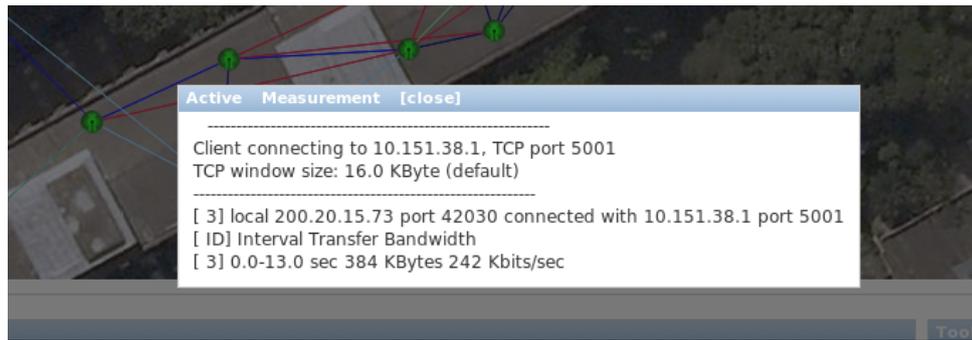


Figura 17: Captura de tela com o resultado de uma medida de vazão TCP com Iperf.

apresenta todas as mensagens de alerta. Já as demais, *Critical*, *Warning* e *Info*, apresentam os alertas correspondentes ao seu nível – Crítico, Aviso e Informação, respectivamente, tal como detalhado na Tabela 4.

4.6 DETALHES DA IMPLEMENTAÇÃO

Um dos principais requisitos de uma plataforma de gerência, como definido anteriormente, é a possibilidade de a ferramenta ser acessada de qualquer estação sem necessidade de instalação de softwares em cada estação em que será utilizada. Assim, a plataforma foi desenvolvida como uma ferramenta web, podendo ser acessada de qualquer navegador.

Para o desenvolvimento da plataforma foi utilizado o framework para desenvolvimento web utilizando Python, chamado Django [Django 2011], que vem sendo amplamente utilizado para desenvolvimento de sistemas web. O Django foi desenvolvido para o <http://www.washingtonpost.com> (famoso portal de notícias norte americano), que até hoje o utiliza. Outros sites que utilizam Django podem ser encontrados em <http://www.djangosites.org>.

O Django possui uma arquitetura multicamada baseada no modelo MVC (*Model View Controller*) [Deacon 2005] que é chamada de MTV (*Model Template View*). Esses modelos de arquitetura permitem o desenvolvimento de aplicações divididas em partes independentes. Os componentes do modelo MTV são os seguintes:

- *Model*: Camada da base de dados. É onde as estruturas de dados utilizadas pela aplicação são definidas. Aqui define-se também como os dados serão tratados e armazenados, as tabelas que a aplicação irá utilizar, os tipos de variáveis e outros parâmetros para definição de uma base de dados.
- *Template*: Camada de apresentação. Basicamente é responsável pelo formato de exibição das informações na página web. Recebe as informações processadas pela camada *View*.

- *View*: Camada lógica da aplicação. É responsável por tratar as requisições feitas através da interface web e retornar as respostas. Essa camada também interage com a base de dados fazendo inserções e consultas.

Um projeto desenvolvido em Django pode ser constituído de uma ou mais aplicações (*apps*, termo utilizado na documentação do Django) que podem ou não depender umas das outras. Cada aplicação possui um arquivo para definição dos dados chamado *models.py*, um arquivo que ficará responsável pela lógica da aplicação chamado de *views.py*, um arquivo chamado *admin.py* responsável por habilitar a configuração das tabelas definidas no *models.py* através do Painel de Configuração e um arquivo chamado *urls.py* que mapeia as *views* criadas em *views.py* em *urls* que serão requisitadas através de um navegador web. Além disso, quando necessário, a aplicação poderá ter um *template* associado que será responsável pela apresentação das informações no navegador.

Para contemplar os módulos de MeshAdmin, as seguintes *apps* foram desenvolvidas:

- *configuration*
- *nodeMonitoring*
- *topologyMonitoring*
- *alert*
- *chart*
- *tools*
- *wrap*

Além disso, foi desenvolvido um programa em python, chamado *net_info_collection.py*, que é responsável pela coleta de informações dos nós e da topologia.

As aplicações e o programa são explicados a seguir.

net_info_collection.py

É um programa desenvolvido em Python responsável pela coleta de informações da rede em malha sem fio. Ele utiliza as bibliotecas do Django, faz inserções no Módulo de Armazenamento e alimenta o Módulo de Alerta, porém não faz parte de nenhuma das *apps* de MeshAdmin. O administrador deve configurar o agendador de tarefas de seu sistema operacional para executar o programa na periodicidade que lhe for mais conveniente. Em sistemas

GNU/Linux, o agendador de tarefas é conhecido com *crontab*. Um exemplo de configuração do *crontab* para que o programa seja executado de hora em hora é visto a seguir:

```
00 * * * * python /usr/local/meshadmin/net_info_collection.py
```

O *net_info_colletion.py* contempla o Módulo de Coleta de MeshAdmin e possui duas funções principais: uma para coleta de informações dos nós, *nodeInfoCollection()* e outra para coleta de informações sobre a topologia, *topologyInfoCollection()*.

A função *nodeInfoCollection()* implementa o processo gerente na arquitetura SNMP. Ela estabelece uma sessão SNMP com todos os nós inseridos no Painel de Configuração e, para cada nó, faz a requisição das variáveis descritas na Seção 4.2.1.

A função *topologyInfoCollection()* é responsável pela coleta de informações sobre a topologia da rede. Um *socket* é criado para escutar a porta 2004 do *gateway* da rede, que atualiza as informações de topologia a cada 10 segundos. Esse valor de porta pode ser alterado no arquivo de configuração do OLSR no roteador, *olsrd.conf*.

As informações coletadas por ambas as funções são armazenadas na base de dados da plataforma definida pelo Módulo de Armazenamento. Este módulo é composto pelas definições de estruturas presentes nos arquivos *models.py* de cada *app*. As *apps* são explicadas a seguir.

configuration app

Define as estruturas das variáveis que mantêm as informações sobre redes e nós que serão monitorados pela ferramenta em seu arquivo *models.py*. Seu *views.py* é responsável por alimentar o Módulo de Exibição de Informação de Redes e Nós para formar a árvore de redes e nós.

nodeMonitoring app

Define as estruturas das variáveis que serão coletadas pelo Módulo de Coleta de Informação dos Nós em seu arquivo *models.py*. O arquivo *views.py* trata essas informações armazenadas passando para o *template* os valores em um formato mais amigável para leitura do administrador (por exemplo, transformar bytes em Megabytes, *timetics* em horas etc.). Essa *app* possui um *template* associado que faz parte do Módulo de Exibição de Informação dos Nós e mostra as últimas informações coletadas e armazenadas pela rotina de coleta da plataforma.

topologyMonitoring app

Define as estruturas das variáveis para armazenamento das informações de enlace coletadas pelo Módulo de Coleta em *models.py*. O Módulo de Exibição da Topologia é definido em duas *views* nesta aplicação – *topologyInformation()* e *index()* – e pelo seu *template*. A *view index()*

é responsável por alimentar este módulo com as coordenadas de redes e nós, enquanto a *view topologyInformation()* consulta o gateway da rede em busca das informações de enlace. Existem duas diferenças dessa *view* para a função *topologyInfoCollection()*: (i) a *view* só é ativada quando a ferramenta está sendo exibida em algum navegador e (ii) os dados coletados não são armazenados na base de dados, pois esta *view* tem como único objetivo exibir a topologia em tempo real. O método para obter as informações de topologia é o mesmo.

O *template* associado a esta *app* faz requisições periódicas utilizando a JQuery Ajax API⁶ (a periodicidade é definida pelo administrador no Painel de Configuração). Além disso, o *template* possui um código Javascript que interage com a API do Google Maps para exibição do mapa com os nós posicionados de acordo com sua posição geográfica que faz parte do Módulo de Exibição da Topologia.

alert app

Em seu *models.py*, *alert app* define a estrutura dos registros de alerta. Possui duas *views*: uma para arquivar os registros no Módulo de Armazenamento, chamada *maLogger()*, e outra para exibição pelo Módulo de Exibição de Alertas, chamada *log()*. A primeira é utilizada na execução da rotina de coleta. Chamadas para esta *view* são feitas quando alguma exceção é encontrada e ao fim de cada rotina, para informar que a coleta foi executada com sucesso. Já a segunda é chamada quando a ferramenta está em execução e exibe os registros armazenados através do Módulo de Exibição.

chart app

Chart app é responsável pela ferramenta de criação de gráficos da plataforma. Seu *models.py* define o formulário de inserção de dados – presente no canto inferior direito da interface (Figura 12) – para a criação dos gráficos. Possui uma *view* para processar as informações inseridas no formulário, consultar a base de dados e enviar para o *template* o gráfico que será exibido. O gráfico é gerado utilizando a ferramenta Google Chart API⁷ em conjunto com o módulo Python Google Chart Wrapper⁸. Essa *view* consulta a base de dados em busca dos parâmetros necessários para a criação do gráfico e utiliza a API para gerar a imagem que é apresentada pelo *template* da *app*.

tools app

É responsável pela ferramenta de medidas ativas da plataforma. No *models.py*, define o formulário que recebe o nó e o tipo de teste que será executado. Sua *view* processa as informações

⁶<http://api.jquery.com/jquery.ajax/>

⁷<http://code.google.com/apis/chart/>

⁸<http://code.google.com/p/google-chartwrapper/>

inseridas na interface e faz uma chamada de sistema executando o comando de acordo com as informações fornecidas. Para que este módulo funcione corretamente, é necessário que o servidor de gerência possua as ferramentas Ping e Iperf instaladas. Além disso, os roteadores devem possuir um processo servidor Iperf sendo executado em *background*.

wrap app

Wrap app tem como objetivo unificar as outras *apps* em uma única interface. Possui apenas um *template* associado que agrega os *templates* integrantes do Módulo de Exibição.

O manual de instalação da ferramenta MeshAdmin está disponível no Apêndice B.

CAPÍTULO 5 - AVALIAÇÃO DA PLATAFORMA MESHADMIN

5.1 AVALIAÇÃO DA PLATAFORMA

Com intuito de avaliar o desempenho da plataforma no que diz respeito ao *overhead* gerado pelo tráfego adicional de monitoramento injetado na rede pelo Módulo de Coleta e à capacidade da ferramenta em apontar falhas na rede, foram realizados testes de uso da ferramenta MeshAdmin na rede em malha sem fio mantida pelo Laboratório MídiaCom no Instituto de Computação da UFF.

5.1.1 DESEMPENHO DO MÓDULO DE COLETA

Como descrito anteriormente, MeshAdmin possui um mecanismo de coleta de informações dos nós da rede em malha sem fio. Esse processo de coleta injeta tráfego adicional de monitoramento na rede. Esse tráfego adicional não deve atrapalhar o funcionamento da rede mesh, interferindo na troca de informações por usuários e aplicações. Os testes de desempenho têm como objetivo quantificar o tráfego adicional na rede e por quanto tempo as rotinas de monitoramento injetam tráfego na rede.

Os testes foram realizados em um ambiente real, na rede em malha sem fio mantida pelo Laboratório MídiaCom no Instituto de Computação da UFF. Foram elaborados 5 cenários diferentes para a realização dos testes: com 5, 7, 9, 10 e 12 nós. Para cada cenário foram realizadas 30 medições a cada 20 minutos. Os gráficos apresentados a seguir mostram a média e a barra de erro com intervalo de confiança de 95% para a quantidade de bytes de monitoramento trafegados e duração da rotina de coleta de informações dos nós nos 30 testes realizados para cada cenário. Todas as rotinas de teste foram efetuadas com sucesso pela plataforma MeshAdmin, coletando informação de todos os nós conectados à rede em malha. A Figura 18 mostra os cenários em que foram realizados os testes.

A Figura 19 mostra a quantidade de bytes injetados na rede em malha pela rotina de monitoramento. Como esperado, a quantidade de tráfego injetado cresce linearmente à medida que o número de nós da rede aumenta. Podemos perceber também, que a quantidade de bytes em cada

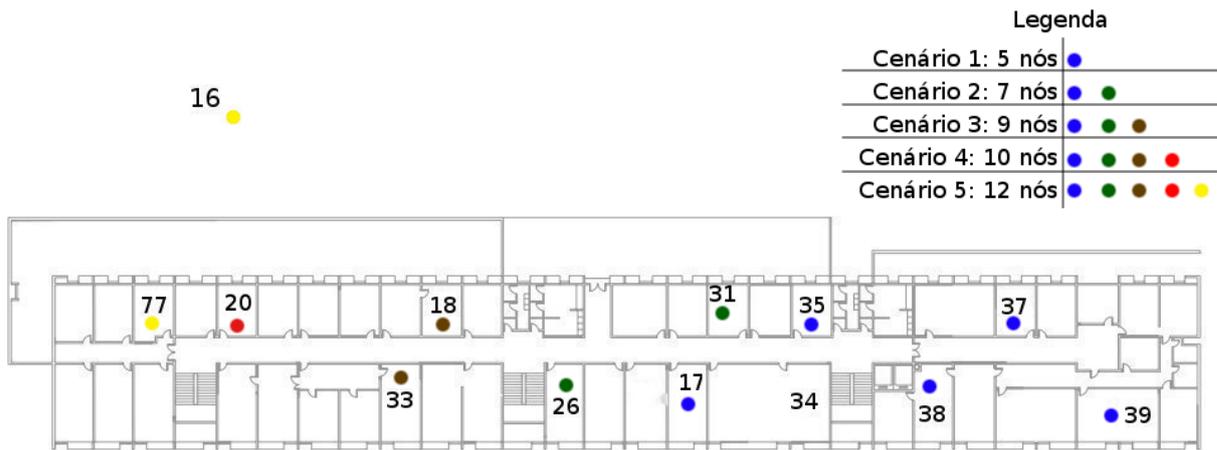


Figura 18: Cenários dos testes.

cenário varia muito pouco. Essa pequena variação se deve a alguma retransmissão por conta de alguma informação incompleta recebida pelo processo gerente na plataforma de gerência.

No cenário com 12 nós, trafega na rede uma média de 23 Kbytes de informação de monitoramento em cada rotina de monitoramento. Esse é um valor relativamente baixo considerando-se a capacidade de transmissão da rede. É possível perceber ainda que a quantidade de bytes de monitoramento trafegados cresce linearmente de acordo o número de nós na rede, pois os bytes transmitidos se referem às informações coletadas de cada nó. Para cada nó monitorado são trafegados aproximadamente 2 Kbytes de informação de monitoramento.

A Figura 20 mostra o gráfico, com a média e a barra de erro com intervalo de confiança de 95%, do tempo de duração de cada rotina de testes para cada um dos cenários. Como era esperado o tempo de duração também aumenta com o número de nós. Mas seu aumento não se deve apenas à quantidade de nós adicionada, mas também é devido à quantidade média de saltos de cada nó ao *gateway*.

Na rede utilizada para os testes, o *gateway* (nó 39) fica em uma das extremidades da mesma. Dessa forma, à medida que nós são adicionados aos cenários de testes, a quantidade média de saltos de um nó até o *gateway* também aumenta. Com isso, aumenta também a latência média dos nós ao *gateway*, e por sua vez, ao servidor de gerência. Isso faz com que o tempo de coleta de dados seja ainda maior. Porém, mesmo em um cenário com 12 nós, estando os 2 últimos nós a uma quantidade média de 8 saltos do *gateway*, o tempo total médio de coleta de informação de todos os nós da rede foi de apenas 4 segundos. Ou seja, para cada rodada de coleta de dados nesse cenário, a plataforma de gerência insere tráfego de monitoramento por apenas 4 segundos. Ainda, é possível afirmar, baseado na barra de erro para esse cenário, que o administrador poderia realizar coletas a cada 5 segundos na rede com 12 nós, caso precise de

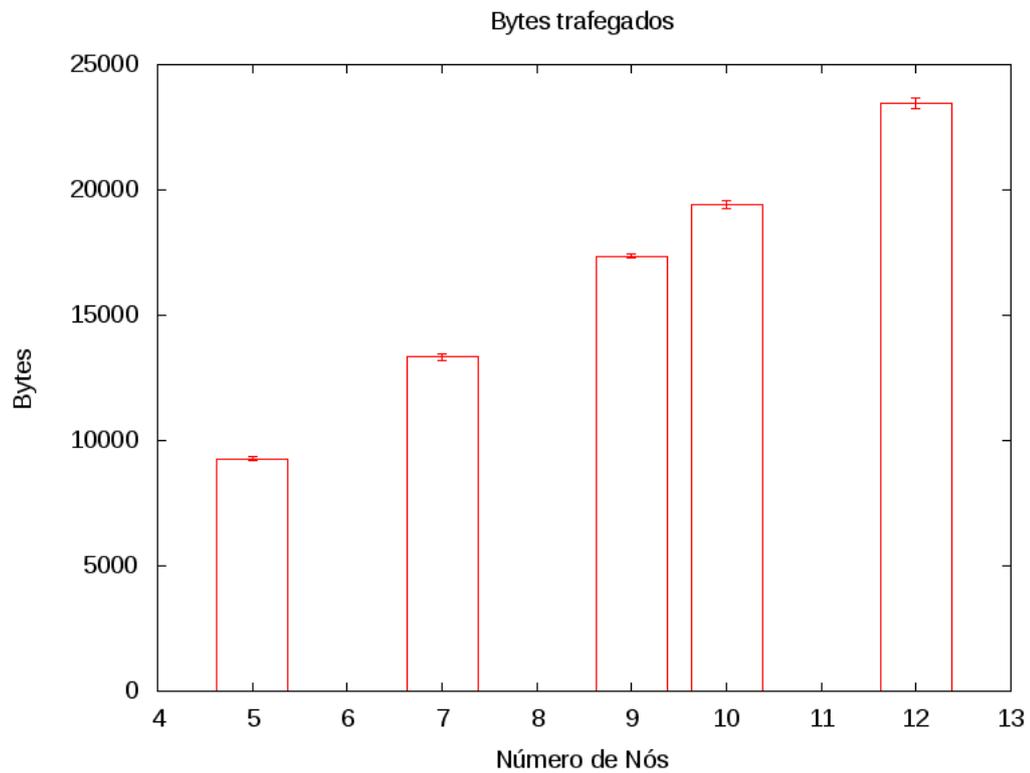


Figura 19: Gráfico com a quantidade de bytes de monitoramento trafegados na rede.

uma granularidade alta de informações.

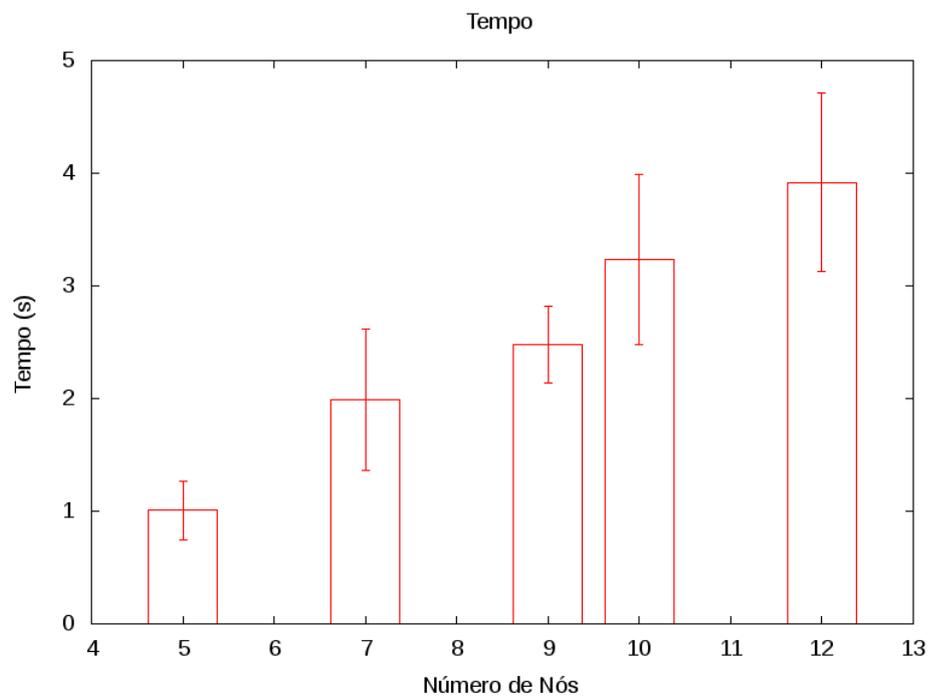


Figura 20: Gráfico do tempo médio para execução das rotinas de monitoramento.

5.1.2 FUNCIONAMENTO DO MECANISMO DE ALERTAS

Foi montado um cenário na rede de testes mantida pelo Laboratório MídiaCom com o intuito de avaliar o funcionamento dos mecanismos de alertas e a capacidade da plataforma em informar ao administrador da rede a ocorrência de falhas. A Figura 21 mostra a plataforma MeshAdmin monitorando a rede de testes do Instituto de Computação da UFF. Apesar de na figura os nós aparecerem posicionados no teto do prédio, a rede fica interna ao prédio. O Módulo de Exibição de Alertas informa apenas que as rotinas de monitoramento dos nós foram executadas com sucesso.

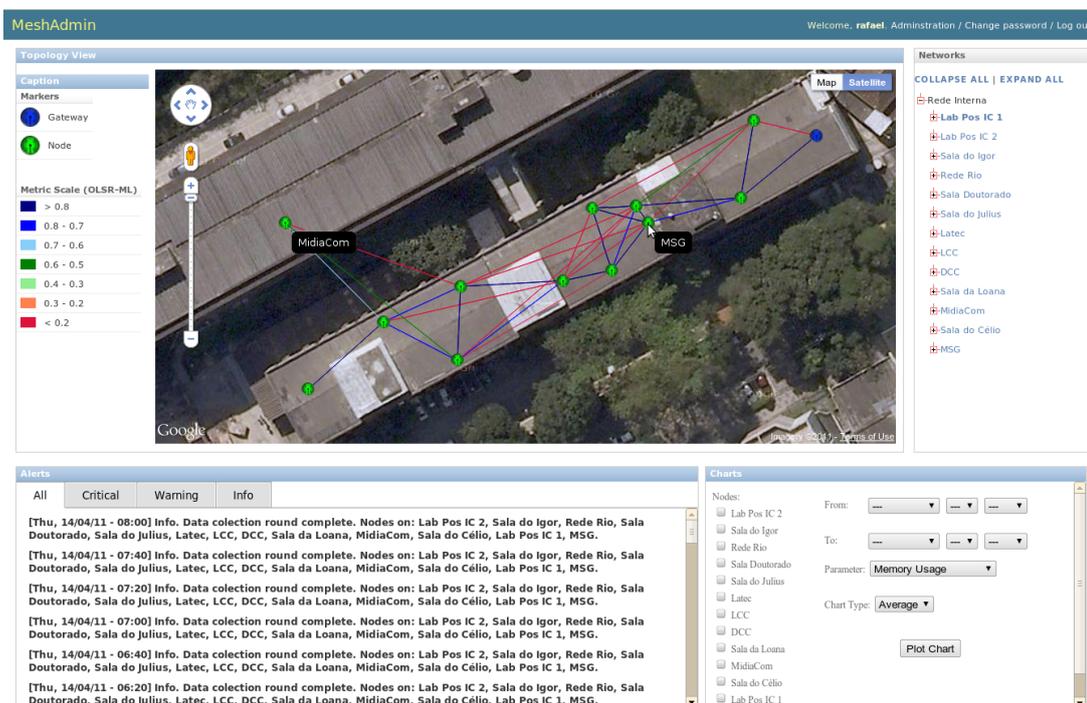


Figura 21: Rede de testes antes da inserção de falhas.

Para testar o funcionamento do Módulo de Alerta, os seguintes pontos de alerta foram inseridos na rede:

- Nó “MídiaCom” desligado;
- Interface cabeada do nó “Lab Pos IC 2” desativada;
- Nó “MSG” não adicionado ao Painel de Configuração;

A Figura 22 mostra uma captura da tela de MeshAdmin após a inserção dessas falhas na rede. O nó MSG já não se encontra mais no Módulo de Exibição da Topologia, pois foi removido do Painel de Configuração. Podemos observar também, que não há enlaces para o nó

Mídiacom, já que o mesmo está desligado. Ainda, no Módulo de Exibição de Alertas, temos as seguintes mensagens:

1. [Thu, 14/04/11 - 08:20] Warning. Gateway: Lab Pos IC 1. IPs found in route table but not configured: 10.151.34.1.
2. [Thu, 14/04/11 - 08:20] Info. Data collection round complete. Nodes on: Lab Pos IC 2, Sala do Igor, Rede Rio, Sala Doutorado, Sala do Julius, Latec, LCC, DCC, Sala da Loana, Sala do Célio, Lab Pos IC 1.
3. [Thu, 14/04/11 - 08:20] Critical. Nodes not reached: MidiaCom.
4. [Thu, 14/04/11 - 08:20] Warning. Node: Lab Pos IC 2. Interface eth0.0 is down.

A primeira diz respeito ao nó com endereço IP 10.151.34.1, encontrado na tabela de roteamento do gateway “Lab Pos IC 1”. Esse endereço é o endereço IP do nó “MSG” que foi retirado do Painel de Configuração. A segunda informa ao administrador que a rotina de coleta foi executada e os nós que estão ativos na rede. A terceira alerta ao administrador que o nó “MidiaCom”, configurado no Painel de Configuração, não foi alcançado pelo Módulo de Monitoramento. Já a quarta, informa ao administrador que a interface eth0.0 do “Lab Pos IC 2” está inativa.

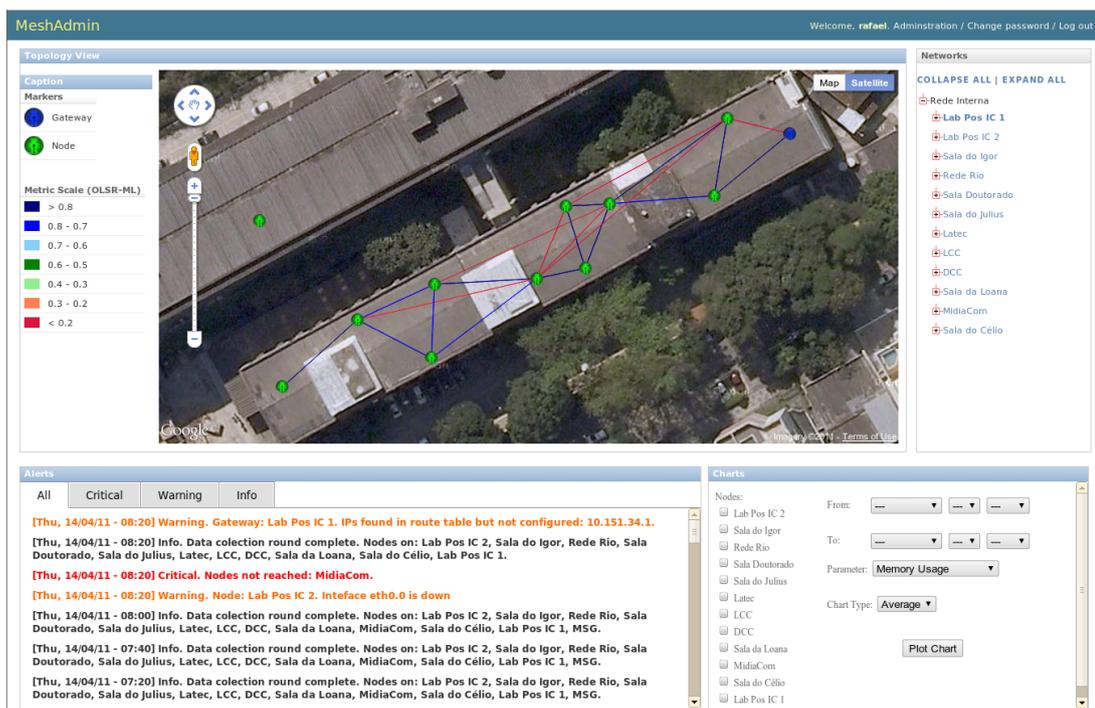


Figura 22: Rede de testes após a inserção de falhas.

Em uma situação real de falhas, como a apresentada, os alertas de MeshAdmin têm como propósito informar de maneira objetiva os problemas encontrados na rede. A partir daí, o administrador pode fazer consultas à base de dados, utilizando os módulos de exibição de Gráficos e Nós, para tirar qualquer dúvida sobre o problema reportado.

5.2 COMPARAÇÃO COM TRABALHOS RELACIONADOS

A Tabela 5 compara MeshAdmin às outras ferramentas apresentadas no Capítulo 2.

Tabela 5: Comparação das ferramentas apresentadas com MeshAdmin.

	Coleta de Dados			Visualização da Topologia	Monitoramento de falhas		Config. de Nós
	Usuários	Nós	Rede		Detecção	Notificação	
MeshAdmin		x	x	x	x	x	
[Sailham et al.]			x		x	x	
MeshMon			x		x	x	
Mesh-Mon	x		x				
MeshFlow	x		x		x		
Abaré		x	x		x	x	x
Maya							x
SCUBA			x	x	x		
MTV			x	x	x		
WiFiDog	x						
CoovaChilli	x						

Das ferramentas citadas, MeshAdmin e Abaré são as únicas ferramentas que se preocupam com o monitoramento dos nós da rede. Enquanto as outras ferramentas coletam apenas informações de parâmetros da rede e de fluxos de usuários, MeshAdmin monitora diversos parâmetros dos nós. Em sistemas embarcados com recursos de disco, processamento e memória limitados, é interessante observar o comportamento desses recursos e, com isso, prever possíveis problemas com esses equipamentos. Em relação ao *framework* Abaré, que apresenta grande contribuição na parte de configuração e resolução autônoma de problemas, MeshAdmin contribui ainda, com a possibilidade de visualização da topologia e dos diversos dados coletados de forma gráfica e em um ambiente web, podendo ser acessada a partir de qualquer estação de trabalho conectada à Internet ou à rede em malha.

Além disso, em sua implementação atual, a plataforma já atende a boa parte dos requisitos necessários para gerência de redes em malha sem fio, como visualização da topologia e o próprio monitoramento de nós sem gerar tráfego de monitoramento que prejudique o funcionamento da rede. MeshAdmin também se mostrou escalável, não tendo queda de rendimento na medida em que nós foram sendo adicionados à rede de teste.

Os resultados da avaliação mostram que a plataforma é capaz de coletar dados da rede com precisão, inserindo uma quantidade mínima de tráfego de monitoramento para coleta de informações dos nós e fazendo a coleta de informações de enlaces sem inserção de tráfego adicional, aproveitando-se apenas das informações fornecidas pelo protocolo de roteamento OLSR, disponíveis através de consulta ao *gateway* da rede mesh. Para que o tráfego de dados concorrente não interfira no monitoramento da rede, é desejável que seja implementada uma política de diferenciação de tráfego exclusiva para o tráfego SNMP na rede. Isso evitará que as informações de monitoramento sejam perdidas por conta de excesso de tráfego na rede. Como a banda consumida pelo tráfego de monitoramento utiliza uma pequena fração da capacidade do canal, isso não seria um limitante para o tráfego de dados concorrente.

5.3 LIMITAÇÕES DA IMPLEMENTAÇÃO ATUAL

Na implementação atual, a ferramenta para geração de gráficos permite apenas a geração de gráficos de barras, que mostram a média de parâmetros que variam dentro de um intervalo definido (como memória disponível ou espaço em disco utilizado, por exemplo) ou o valor absoluto de parâmetros que crescem indefinidamente (bytes trafegados, por exemplo). Também é interessante a visualização de gráficos lineares que permitam observar a variação de um determinado parâmetro em um determinado nó ao longo do tempo. Ainda, a API do Google Chart possui limitações em relação ao tamanho da imagem gerada pelo gráfico. Por conta disso, a ferramenta de geração de gráficos em barra se limita a plotar informações de até onze nós.

O Módulo de Exibição da Topologia utiliza imagens de satélite, disponibilizadas pela API do Google Maps. Apesar de ser muito útil para visualização de redes externas, ela impossibilita a visualização de topologias de redes internas, o que pode ser observado no exemplo da Figura 21. Seria interessante adicionar a possibilidade de incluir uma figura definida pelo usuário (da planta baixa de um edifício, por exemplo) facilitando o monitoramento de redes internas. Além disso, seria interessante que o Módulo de Exibição da Topologia permitisse a visualização do histórico da topologia, utilizando os dados armazenados na base de dados da plataforma.

O Módulo de Coleta de Informações dos Nós já coleta as informações sobre os rádios dos nós mesh. As informações sobre os vizinhos conectados são dinâmicas, pois variam com as mudanças na topologia da rede. Porém, as variáveis definidas no módulo SNMP devem ter tamanho fixo e, devido à limitação nos tamanhos dessas variáveis, o Módulo de Coleta pode coletar apenas informações de dez vizinhos conectados a cada nó, que serão os dez primeiros listados pelo agente SNMP. As informações referentes aos outros vizinhos serão ignoradas.

CAPÍTULO 6 - CONCLUSÃO

A gerência de redes em malha sem fio é uma tarefa mais complexa que gerência de redes cabeadas tradicionais e de redes sem fio infraestruturadas devido, principalmente, à limitação de recursos nos roteadores e à grande variabilidade de qualidade nos enlaces do *backbone* sem fio. Apesar de existirem propostas de ferramentas de gerência de redes mesh na literatura, ainda existem poucas soluções não-proprietárias disponíveis na prática para a gerência dessas redes.

Neste contexto, esta dissertação apresentou os principais desafios para a gerência de redes em malha sem fio baseada em trabalhos encontrados na literatura e na experiência no desenvolvimento de soluções para redes em malha sem fio adquirida pelo grupo de pesquisa do Laboratório MídiaCom.

A partir daí, foram definidos os requisitos de uma plataforma integrada para gerência de redes em malha sem fio, de modo que o administrador da rede consiga identificar falhas e diagnosticar problemas na rede sem precisar consultar outras ferramentas e necessite de um mínimo de interação com a plataforma. Foram apresentados, neste texto, os requisitos de uma plataforma que seja capaz de coletar e armazenar estatísticas, auxiliar no monitoramento, efetuar o controle de usuários e configurar uma rede em malha sem fio.

Ainda, este trabalho descreveu MeshAdmin, uma plataforma integrada para gerência de redes em malha sem fio, cuja implementação atual contempla um subconjunto desses requisitos: coleta de dados de nós e da rede monitorada, visualização da topologia e detecção e notificação de falhas.

Essa ferramenta já está em operação na rede de testes mantida pelo Laboratório MídiaCom, localizada no Instituto de Computação da UFF. A plataforma apresentou contribuições em relação ao monitoramento dos nós redes, funcionalidade pouco abordada pelos trabalhos citados, podendo inclusive ser utilizada para monitoramento do sistema de alimentação para roteadores que utilizam energia solar. Além disso, fornece uma interface intuitiva que permite ao administrador a configuração de parâmetros da ferramenta e acessar as diversas informações presentes na ferramenta rapidamente.

Por fim, MeshAdmin foi avaliada a respeito da quantidade de tráfego de monitoramento inserido na rede e quanto a capacidade de indicação de falhas. Os resultados mostraram que

a quantidade de dados inserida na rede é pequena comparada à capacidade do canal de transmissão, sendo em torno de 2Kbytes por nó mesh monitorado, e que o tempo de coleta desses dados permite uma alta granularidade de captura de informações, da ordem de segundos para o maior cenário avaliado com 12 nós. Ainda, MeshAdmin se mostrou eficaz no que diz respeito à indicação de falhas na rede.

6.1 CONTRIBUIÇÕES DA DISSERTAÇÃO

Pode-se destacar como principais contribuições desta dissertação:

- Levantamento bibliográfico sobre gerência de redes em malha sem fio;
- Especificação de requisitos de uma plataforma integrada de gerência de redes em malha sem fio;
- Implementação de uma plataforma de gerência integrada chamada MeshAdmin, com as funções de coleta de dados dos nós e da rede, visualização da topologia e detecção e notificação de falhas;
- Execução de testes para avaliação da plataforma proposta.

6.2 TRABALHOS FUTUROS

A implementação atual de MeshAdmin já atende a boa parte dos requisitos propostos para uma plataforma de gerência integrada, porém, algumas funcionalidades ainda precisam ser implementadas. Além disso, alguns módulos da ferramenta ganharão novas funcionalidades, deixando a plataforma mais completa.

Está em desenvolvimento um módulo para monitoramento dos fluxos que trafegam pela rede. Esse módulo fará a captura de informações sobre fluxos, as armazenará na base de dados e fará a exibição de forma gráfica através do Módulo de Exibição. Ainda, esse módulo ficará responsável pela configuração de acesso (capacidade de banda, endereço IP, login etc) e monitoramento de tráfego dos usuários que acessam a rede.

Outras mensagens de alerta são interessantes para que o administrador consiga identificar ou até mesmo evitar falhas na rede. Essas mensagens podem informar anomalias nos dados capturados pelo Módulo de Monitoramento, como por exemplo, pouca memória RAM disponível, disco com capacidade esgotada, aumento considerável no tráfego de dados etc. Para isso, são

necessários mecanismos inteligentes que comparem os dados que estão sendo coletados, com os dados já armazenados e avaliem se há ou não alguma anomalia, para então gerar um alerta através do Módulo de Alertas. Para isso, será desenvolvido um módulo autônomo de diagnóstico que será incorporado à plataforma.

Ainda, é necessário que seja adicionada à plataforma um mecanismo de controle de usuários, que permita restringir o acesso à rede a usuários previamente cadastrados, baseados nos requisitos definidos na Seção 3.4.

Por fim, é desejado que se integre à ferramenta um módulo de configuração da rede, onde alguns parâmetros de um grupo de nós da rede possam ser configurados de forma automática pela ferramenta, sem necessidade de se acessar nó a nó. Esse módulo pode ser desenvolvido baseado nos requisitos definidos na Seção 3.5.

REFERÊNCIAS BIBLIOGRÁFICAS

- [802.11 2007] 802.11, I. (2007). IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999).
- [Campista et al. 2008] Campista, M., Esposito, P., Moraes, I., Costa, L., Duarte, O., Passos, D., de Albuquerque, C., Muchaluat-Saade, D., and Rubinstein, M. (2008). Routing metrics and protocols for wireless mesh networks. *Network, IEEE*, 22(1):6–12.
- [Clausen et al. 2003] Clausen, T., Jacquet, P., Laouiti, A., and et al. (2003). Optimized link state routing protocol (olsr). *IETF RFC 3626*.
- [CoovaChilli 2011] CoovaChilli (2011). Open source captive portal access controller and radius software. <http://coova.org/wiki/index.php/CoovaChilli>. Acessado em Maio de 2011.
- [Couto et al. 2003] Couto, D., Aguayo, D., Bicket, J., and Morris, R. (2003). A high-throughput path metric for multi-hop wireless routing. In *ACM MobiCom*, San Diego, CA, USA.
- [DD-WRT 2011] DD-WRT (2011). <http://www.dd-wrt.org>. Acessado em Maio de 2011.
- [Deacon 2005] Deacon, J. (2005). Model-view-controller (MVC) architecture. *JOHN DEACON Computer Systems Development, Consulting & Training*.
- [Django 2011] Django (2011). High-Level Python Web Framework. <http://http://www.djangoproject.com/>. Acessado em Maio de 2011.
- [Duarte et al. 2008] Duarte, J., Passos, D., and de Albuquerque, C. (2008). DynTun: Túneis dinâmicos e a escalabilidade de redes em malha. In *Simpósio Brasileiro de Redes de Computadores (SBRC 2008)*, Rio de Janeiro, RJ, Brasil.
- [Duarte et al. 2007] Duarte, J., Passos, D., Valle, R., Oliveira, E., Muchaluat-Saade, D., and Albuquerque, C. (2007). Management issues on wireless mesh networks. In *5th Latin American Network Operations and Management Symposium (LANOMS 2007)*, Petrópolis, RJ, Brasil.
- [Gerk et al. 2009] Gerk, L., Passos, D., Saade, D., and de Albuquerque, C. (2009). Infraestrutura de comunicação em malha sem fio para supervisão e controle de sistemas de transmissão de energia. *Espaço Energia*, 10:1–10.
- [Huang et al. 2007] Huang, F., Yang, Y., and He, L. (2007). A Flow-Based Network Monitoring Framework for Wireless Mesh Networks. *IEEE Wireless Communications*, 14(5).

- [Jardosh et al. 2008] Jardosh, A., Suwannat, P., Hollerer, T., Belding, E., and Almeroth, K. (2008). SCUBA: Focus and Context for Real-Time Mesh Network Health Diagnosis. *Lecture Notes in Computer Science*, 4979:162.
- [Lenczner 2005] Lenczner, M. (2005). Wireless portals with wifidog. *Linux J.*, 2005(140):8.
- [Manzano et al. 2008] Manzano, D., Cano, J., Calafate, C., and Manzoni, P. (2008). Maya: A tool for wireless mesh networks management. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pages 1–6. IEEE.
- [Mills 1991] Mills, D. (1991). Internet time synchronization: the network time protocol. *Communications, IEEE Transactions on*, 39(10):1482–1493.
- [Muchalut-Saade et al. 2007] Muchalut-Saade, D., Albuquerque, C., Magalhaes, L., Passos, D., Duarte, J., and Valle, R. (2007). Redes em malha: Solução de baixo custo para popularização do acesso a internet no Brasil. In *Simpósio Brasileiro de Telecomunicações (SBrT 2007)*, Recife, PE, Brasil.
- [Nanda e Kotz 2008] Nanda, S. and Kotz, D. (2008). Mesh-Mon: A multi-radio mesh monitoring and management system. *Computer Communications*, 31(8):1588–1601.
- [Netflow 2011] Netflow (2011). Cisco IOS Netflow. www.cisco.com/web/go/netflow. Acessado em Maio de 2011.
- [OpenWrt 2011] OpenWrt (2011). <http://openwrt.org>. Acessado em Maio de 2011.
- [Passos et al. 2006] Passos, D., Teixeira, D., Muchalut-Saade, D., Magalhães, L., and Albuquerque, C. (2006). Mesh network performance measurements. In *International Information and Telecommunications Technologies Symposium (I2TS)*, Cuiabá, MT, Brasil.
- [Perkins et al. 2003] Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad hoc on-demand distance vector (aodv) routing. *Internet RFCs*.
- [Pinheiro et al. 2010] Pinheiro, B., de Brito Nascimento, V., Cerqueira, E., Abelém, A., and Neto, A. (2010). Abaré: Um Framework para Implantação, Monitoramento e Gerenciamento Coordenado e Autônomo para Redes em Malha sem Fio. In *XV Workshop de Gerência e Operação de Redes e Serviços - SBRC*, Gramado, RS, Brasil.
- [Raghavendra et al. 2009] Raghavendra, R., Acharya, P., Belding, E., and Almeroth, K. (2009). MeshMon: a multi-tiered framework for wireless mesh network monitoring. In *Proceedings of the 2009 MobiHoc S 3 workshop*, New York, NY, USA. ACM.
- [Rigney et al. 2000] Rigney, C., Willens, S., Rubens, A., and Simpson, W. (2000). Remote authentication dial in user service (RADIUS). Technical report, RFC 2865.
- [Sailhan et al. 2007] Sailhan, F., Fallon, L., Quinn, K., Farrell, P., Collins, S., Parker, D., Ghamri-Doudane, S., and Huang, Y. (2007). Wireless mesh network monitoring: Design, implementation and experiments. In *IEEE DANMS workshop*, Washington, DC, USA.
- [Silva et al. 2010] Silva, B., Gomes, A., Ziviani, A., and Saade, D. (2010). Provisão de qualidade de serviço para o sistema atoms em redes em malha sem fio. In *XXII Congresso Brasileiro de Engenharia Biomédica (CBEB 2010)*, Tiradentes, MG.

- [Stallings 1998] Stallings, W. (1998). *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 3rd edition.
- [SVG 2003] SVG (2003). Scalable vector graphics (svg) 1.1 specification, w3c recommendation. disponível em <http://www.w3.org/TR/SVG11/>.
- [Teixeira et al. 2009] Teixeira, I., Viçoso, R., Correa, B. S. P. M., Gomes, A. T. A., and Ziviani, A. (2009). Suporte Remoto ao Atendimento Médico Emergencial via Dispositivos Móveis. *Revista Eletrônica de Iniciação Científica (REIC)*, 9(3).
- [Valle et al. 2009] Valle, R., Justen, A., Silva, J., Passos, D., Magalhaes, L., Albuquerque, C., and Muchaluat-Saade, D. (2009). Infraestrutura de Comunicação para Linhas de Transmissão de Energia através de Redes em Malha Sem Fio. In *8th International Information and Telecommunication Technologies Symposium (I2TS 2009)*, Florianópolis, SC, Brasil.
- [Valle e Muchaluat-Saade 2007] Valle, R. and Muchaluat-Saade, D. (2007). Visualização da Topologia de Redes Mesh Usando o Padrão SVG. In *Salão de Ferramentas - Simpósio Brasileiro de Redes de Computadores (SBRC 2007)*, Belém, PA, Brasil.
- [Valle e Muchaluat-Saade 2011] Valle, R. and Muchaluat-Saade, D. (2011). Meshadmin: Plataforma integrada de gerência para redes em malha sem fio. In *XVI Workshop de Gerência e Operação de Redes e Serviços - SBRC*, Campo Grande, MS, Brasil.
- [Valle et al. 2008] Valle, R., Passos, D., Albuquerque, C., and Muchaluat-Saade, D. (2008). Mesh Topology Viewer (MTV): an SVG-based interactive mesh network topology visualization tool. In *IEEE Symposium on Computers and Communications, 2008. ISCC 2008*, pages 292–297, Marrakesh, Marrocos.

APÊNDICE

Apêndice A

Simple Network Management Protocol (SNMP)

O termo SNMP é, na verdade, utilizado para referenciar um conjunto de padrões para gerência de redes TCP/IP que incluem o próprio protocolo, a especificação para a base de informações de gerência – MIB (*Management Information Base*) – e a especificação da estrutura dos objetos de gerência - SMI (*Structure and Identification of Management Information*) [Stallings 1998].

A arquitetura da gerência de redes TCP/IP possui os seguintes elementos:

- Estação de gerência;
- Agente;
- Base de informações de gerência (MIB);
- O protocolo de gerência de rede (SNMP).

A estação de gerência é o ponto de conexão entre o sistema de gerência e o administrador da rede. Tipicamente, ela contém um conjunto de aplicações para análise dos dados, uma interface gráfica para monitoramento e controle da rede e uma base de dados contendo informações coletadas dos diversos elementos da rede.

Cada elemento da rede, que será monitorado pela estação de gerência, deve possuir um agente SNMP. O agente responde às requisições feitas pela estação de gerência, sejam elas de coleta de informações ou de alterações nas configurações dos elementos de rede, e ainda, podem fornecer de modo assíncrono informações importantes que não foram solicitadas pela estação de gerência.

Os recursos de cada elemento da rede são representados como objetos. Cada objeto é uma variável que representa um aspecto do agente na gerência do elemento de rede. O conjunto desses objetos é definido como uma MIB. Uma MIB define basicamente uma coleção de

parâmetros contidos no agente que podem ser gerenciados pela estação de gerência. Uma MIB em particular é um padrão multiplataforma relacionado a uma classe de objetos de um agente. Por exemplo, a MIB do 802.11 é responsável por fornecer diversas informações de gerência a respeito do padrão em diversas plataformas que implementam esse mesmo padrão. A estação de gerência realiza o monitoramento através da coleta dos valores de variáveis definidas em uma MIB e pode alterar as configurações no agente através da alteração de alguma dessas variáveis.

A ligação entre a estação de gerência e os agentes é feita pelo SNMP, protocolo *de facto* para gerência de redes TCP/IP. As principais funções do SNMP são:

- GET – permite à estação de gerência obter informações dos agentes;
- SET – permite à estação de gerência modificar variáveis nos agentes;
- TRAP – permite ao agente notificar à estação de gerência sobre determinados eventos.

O SNMP é um protocolo de aplicação parte da pilha de protocolos TCP/IP, que utiliza o protocolo UDP (*User Datagram Protocol*) na camada de transporte. A Figura 23, retirada de [Stallings 1998], mostra a arquitetura típica dos protocolos utilizados para gerência de redes TCP/IP com SNMP.

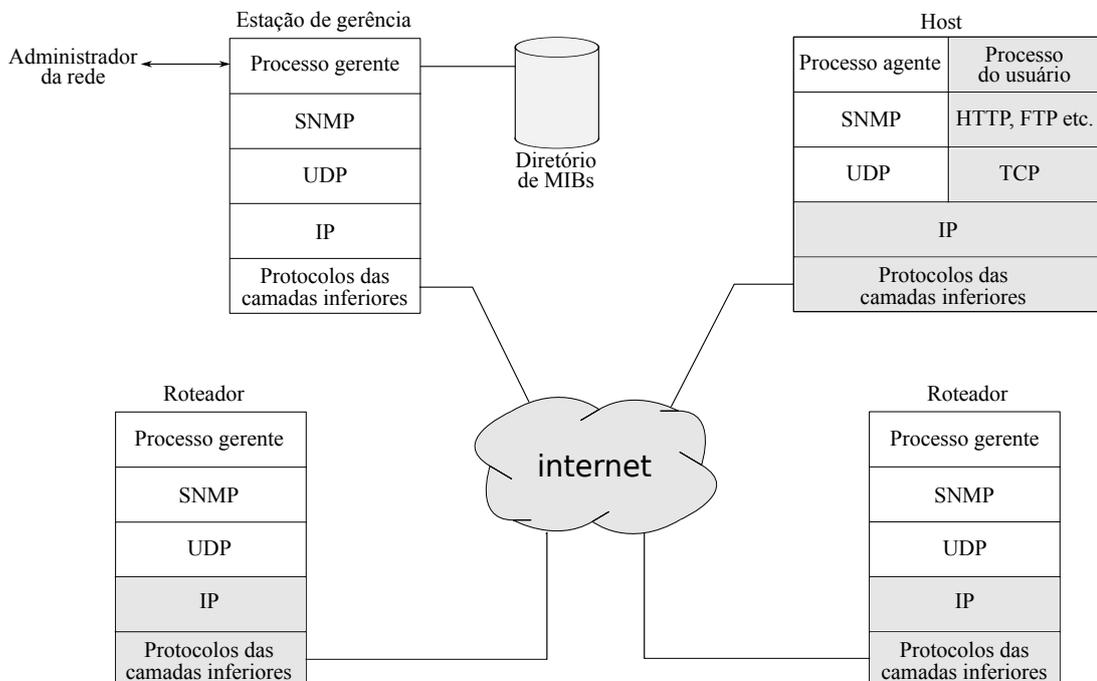


Figura 23: Arquitetura SNMP [Stallings 1998].

Uma estação de gerência possui um processo gerente que acessa um diretório de MIBs e possui uma interface de comunicação com o administrador da rede. A estação de gerência se comunica com os elementos de rede através do SNMP, implementado através do UDP, na camada de transporte, do IP, na camada de rede, e dos diversos protocolos da camada de enlace (por exemplo, Ethernet e 802.11).

Cada agente também deve implementar SNMP, UDP e IP. Ainda, o processo agente deve interpretar as mensagens SNMP e verificar as MIBs do agente. Outros processos e protocolos são suportados por *hosts* e roteadores de uma rede. Os protocolos e processos são os objetos de gerência do agente. Cada processo, protocolo ou variável de sistema faz parte do ambiente operacional e deve estar mapeado em uma MIB que seja implementada pelo agente e esteja no diretório de MIBs para que possa ser compreendida pela estação gerente. Na Figura 23, o ambiente operacional é destacado pelo fundo sombreado. As demais partes são as que dão suporte à gerência da rede.

A Figura 24, retirada de [Stallings 1998], mostra como funciona a troca de mensagens SNMP. Na figura são ilustradas três mensagens, *GetRequest*, que utiliza a função, *get*, *SetRequest*, que utiliza a função *set* e *Trap*, que utiliza a função *trap*. As duas primeiras são feitas pela estação de gerência ao agente, que deve confirmar seu recebimento com uma mensagem do tipo *GetResponse*. Já a terceira, é gerada pelo agente no caso da ocorrência de algum evento que afete os objetos de gerência.

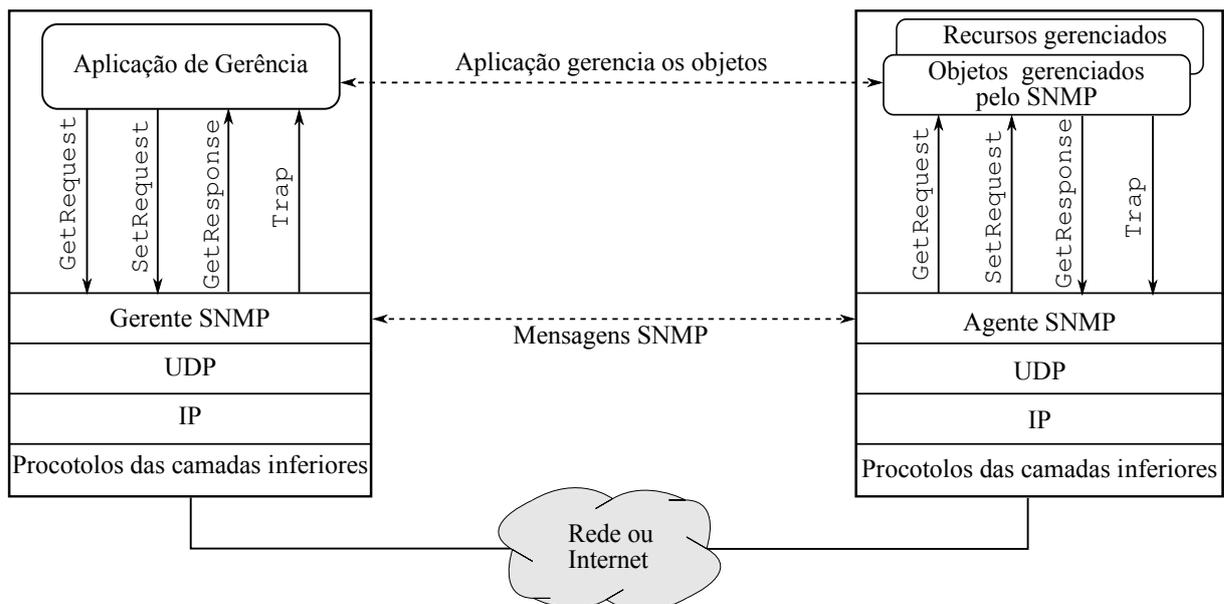


Figura 24: Exemplo de troca de mensagens no SNMPv1 [Stallings 1998].

Além do funcionamento básico referente à primeira versão do protocolo (SNMPv1), novas versões foram desenvolvidas, apresentando melhorias em relação à versão inicial. Em sua segunda versão (SNMPv2), foram incluídas melhorias em relação ao desempenho e à estrutura das informações de gerência (SMI) e à possibilidade de comunicação entre as estações de gerência. Na sua terceira versão (SNMPv3), foram incluídas melhorias em relação à segurança, como autenticação e controle de acesso.

Apêndice B

Manual de Instalação da Plataforma MeshAdmin

Visão Geral

Este guia descreve o processo de instalação e configuração da ferramenta MeshAdmin. Está dividido nas seguintes partes:

- Requisitos do Sistema
- Instalação
- Configuração
- Acesso à plataforma

Requisitos

A plataforma foi desenvolvida para ambientes web de sistemas operacionais baseados em GNU/Linux, kernel versão 2.6.

Cada distribuição Linux pode ou não possuir pacotes disponíveis para os componentes listados abaixo. Caso não possuam, os componentes devem ser obtidos nos sites de seus respectivos projetos e instalados a partir do código fonte conforme recomendado na documentação dos mesmos.

Para o correto funcionamento da ferramenta, os seguintes componentes devem ser instalados no sistema:

Python versão 2:

Linguagem de script de alto nível

<http://python.org/download/releases/2.7.1/>

PostgreSQL versão 9:

Sistema de banco de dados relacional

<http://www.postgresql.org.br/downloads>

Django:

Framework web de alto nível baseado em Python

<http://www.djangoproject.com/download/>

Psycopg versão 2:

Backend para utilização do Django em conjunto do PostgreSQL

<http://initd.org/psycopg/download/>

Apache versão 2:

Servidor HTTP

<http://www.apache.org/dyn/closer.cgi>

mod_wsgi:

Um módulo para que o Apache suporte aplicações Python WSGI

<http://code.google.com/p/modwsgi/wiki/DownloadTheSoftware>

Net-SNMP versão 5.6.1:

Conjunto de aplicações que implementam o protocolo SNMP

<http://www.net-snmp.org/download.html>

GChartWrapper 0.9:

Conjunto de funções em Python a API Google Chart

<http://pypi.python.org/pypi/GChartWrapper/>

Instalação

Há duas opções de instalação da plataforma:

Opção 1:

Baixando o código fonte compactado e extraindo o código para a pasta de sua preferência. Link para download: <http://www.midiacom.uff.br/downloads/MeshAdmin/meshadmin.tar.gz>

Na linha de comando vá para o diretório em que deseja instalar a ferramenta (no nosso exemplo, vamos instalar em /usr/local):

```
# cd /usr/local/
```

Depois baixe o código e extraia:

```
# wget -c http://www.midiacom.uff.br/downloads/MeshAdmin/meshadmin.tar.gz
# tar -xzf meshadmin.tar.gz
```

Se o processo for executado corretamente será possível listar o diretório:

```
# ls /usr/local/MeshAdmin
gerencia  media  net_info_collection.py  templates
```

Opção 2:

Baixando o código do repositório SVN. Para isso, é necessário instalar um cliente subversion (<http://subversion.apache.org>). Na linha de comando vá para o diretório em que deseja instalar a ferramenta (no nosso exemplo, vamos instalar em /usr/local) e faça o checkout do código.

```
# cd /usr/local/
# svn checkout http://www.midiacom.uff.br/svn/Ferramentas/MeshAdmin
```

Se o processo for executado corretamente será possível listar o diretório:

```
# ls /usr/local/MeshAdmin
gerencia media net_info_collection.py templates
```

Configuração**APACHE**

Para a configuração de mod_wsgi, adicione a seguinte linha ao arquivo de configuração do Apache:

```
LoadModule wsgi\module modules/mod\wsgi.so
```

Em seguida, ainda no arquivo de configuração, é necessário incluir as configurações dos diretórios de MeshAdmin.

```
# MeshAdmin Config
Alias /site_media/ /usr/local/MeshAdmin/media/
<Directory /usr/local/MeshAdmin/media>
    Order deny,allow
    Allow from all
</Directory>
WSGIScriptAlias /meshadmin /usr/local/MeshAdmin/gerencia/apache/django.wsgi
<Directory /usr/local/MeshAdmin/gerencia/apache>
    Order deny,allow
    Allow from all
</Directory>
```

POSTGRESQL

Inicie o servidor PostgreSQL (varia de acordo com a distribuição Linux utilizada). Primeiro, assegure-se que o usuário postgres foi criado e pertence ao grupo postgres:

```
# groups postgres
```

Em seguida, acesse a conta do usuário postgres:

```
# su root
# su - postgres
```

Crie um usuário que será responsável pela manipulação do banco de dados. Aqui, dê preferência para utilizar um login de acesso já existente no sistema, isso facilita o acesso do usuário à base de dados.

```
# createuser <username>
```

Três perguntas serão feitas em seguida. No nosso caso, só há necessidade de poder criar bases de dados, assim podemos responder da seguinte forma:

```
Shall the new role be a superuser? (y/n)  n
Shall the new role be allowed to create databases? (y/n)  y
Shall the new role be allowed to create more new roles? (y/n)  n
```

Finalmente, criamos a base de dados:

```
# createdb -0 <nome_usuario> <base_dados>
```

A base de dados pode então ser acessada da seguinte maneira:

```
# su - <nome_usuario>
$ psql <base_dados>
```

Nota: Na configuração de MeshAdmin (no arquivo gerencia/settings.py), por padrão o nome de usuário e senha são configurados como admin/admin e a base de dados nomeada 'management'. A seguir, esses parâmetros serão configurados.

MESHADMIN

Por último é necessário configurar alguns parâmetros da própria ferramenta.

- No arquivo gerencia/settings.py, no bloco DATABASES, altere os campos NAME, USER e PASSWORD, como o nome da base de dados, o nome do usuário que criou a base e sua senha, respectivamente.

- Em seguida, para criar as tabelas na base de dados, vá para a pasta gerencia e execute o comando:

```
# python2.7 manage.py syncdb
```

Será solicitado pelo programa, a criação de um usuário para acesso à plataforma. Este usuário é armazenado na base da plataforma e não tem relação com a base de usuários do sistema operacional.

Configure o script de coleta no crontab do Sistema Operacional. Edite o arquivo:

```
# crontab -e
```

Adicione a linha de acordo com o período de coleta desejado. Exemplos:

De hora em hora:

```
* * * * * /usr/bin/python2.7 /usr/local/MeshAdmin/net_info_collection.py
```

De 20 em 20 minutos:

```
*/20 * * * * /usr/bin/python2.7 /usr/local/MeshAdmin/net_info_collection.py
```

Por fim, reinicie o Apache (o comando varia de acordo com a distribuição Linux utilizada).

Acesso à plataforma

Caso os procedimentos acima tenham sido executados corretamente, a plataforma já pode ser acessada através do navegador através do endereço <http://seudominio.com.br/meshadmin>. Insira, o login e a senha criados na configuração de MeshAdmin para acessar o sistema. A partir daí, o administrador pode acessar o Painel de Configuração através do link no canto superior direito e inserir as redes e nós através da rede.

Problemas

Reportar problemas para meshadmin@midia.com.uff.br.