UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIAS, GERÊNCIA E SEGURANÇA DE REDES DE COMPUTADORES

OSCAR EDUARDO PATRÓN GUILLERMO

Uso de Agentes SNMP para monitoramento de Servidores e equipamentos de rede com mobilidade.

Trabalho de Conclusão apresentado como requisito parcial para a obtenção do grau de Especialista

Profa. Dra. Liane Margarida Rockenbach Tarouco
Orientadora

Prof. Dr. Sérgio Luis Cechin
Prof. Dr. Luciano Paschoal Gaspary
Coordenadores do Curso

Porto Alegre, dezembro de 2008.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

IP	Internet Protocol
MIB	Management Information Base
RFC	Request For Comment
SNMP	Simple Nerwork Management Protocol
TCP	Transmission Control Protocol

LISTA DE FIGURAS

RESUMO

Este trabalho apresenta um estudo de caso prático sobre o uso de agentes SNMP, para monitoramento de servidores Linux e Microsoft, assim como bases Wireless, especificamente o modelo DWL AP2100 da D-link, configuração dos agentes nos servidores a serem monitorados e a escolha do software de monitoramento.

Todas as instalações e configurações foram implementadas num ambiente de rede real, dentro da rede de informática do Instituto de Pesquisas Hidráulicas da Universidade Federal do Rio Grande do Sul.

Este é o inicio de um trabalho geral mais amplo, havendo continuidade em todo o processo de monitoramento e definição do esboço geral do contexto. Num trabalho posterior serão analisados todos os dados colhidos pelos programas de monitoramento, para poder fazer uma avaliação de todos os parâmetros monitorados e poder estabelecer critérios para a otimização da rede e verificações sobre a segurança da rede.

Palavras-Chave: Agentes SNMP, MIB Wireless, monitoramento via SNMP.

Use of SNMP agents to monitor servers and network equipment with mobility.

ABSTRACT

This paper presents a case study on the practical use of SNMP agents to monitor Linux and Microsoft servers, and access point Wireless, specifically the model of the AP2100 DWL D-Link, configuration of agents on servers to be monitored and the choice of software Tracking.

All installations and configurations were implemented in a real network environment, in the network at the Institute of Hydraulic Research - Federal University of Rio Grande do Sul

This is the beginning of a wider general work, with continuity throughout the process of tracking and defining the general outline of the context. In a subsequent work will be analyzed all the data collected by tracking programs, to make an assessment of all parameters monitored and can establish criteria for the optimization of the network and checks on the security of the network.

Keywords: SNMP Agents, MIB Wireless, monitoring via SNMP.

1 INTRODUÇÃO

Dimensionamento de redes precisam de ajustes contínuos, por isso o monitoramento de parâmetros importantes das mesmas são importantes para o bom reajuste ou otimização da rede. Os dados de monitoramento traçam um perfil de comportamento da rede e são muito úteis para a análise dos problemas potenciais, e servem para ter um diagnóstico mais preciso do uso dos recursos da rede.

Devido ao grande aumento de novos dispositivos que se comunicam através da rede, o monitoramento de redes de computadores está se tornando uma tarefa cada vez mais complexa, afetando o gerenciamento de redes em diversos aspectos como, por exemplo, a escalabilidade, sobrecarga dos dispositivos e canal de comunicação, além disso, o gerenciamento remoto de dispositivos de rede é importante, pois a necessidade de obter informações de um servidor, por exemplo, tais como espaço livre em disco, uso de memória, uso de cpu, entre outras, permite que possamos tomar atitudes prévias para o bom gerenciamento destes dispositivos, e a disponibilidade dos serviços que são executados nestes servidores.

O gerenciamento de redes essencialmente envolve o monitoramento e a coleta de informações para possível análise dos dispositivos monitorados. Convencionalmente tal gerenciamento é realizado através de agentes SNMP (*Simple Network Management Protocol*), e é a ele que este trabalho vai se ater, e mais especificamente à versão 2 do mesmo.

1.1 Contexto

Uma intranet com centenas de microcomputadores demanda um gerenciamento da eficiência dos serviços disponibilizados via rede de acordo com métricas associadas principalmente a tempo de resposta e disponibilidade. No entanto, a atual forma de gerenciamento centralizado não permite obter o valor dessas métricas do ponto onde o usuário está operando o serviço em relação ao serviço utilizado.

No caso deste trabalho o sistema é de gerenciamento centralizado, pois tão somente importa o monitoramento de poucos dispositivos, existindo duas estações servidoras com o software de monitoramento instalado: um servidor Linux com software de monitoramento livre e outro servidor Microsoft, rodando software de monitoramento proprietário nas suas versões “trial” ou “free”, porém com todas as funcionalidades, já que é um ambiente de testes.

O ambiente analisado é a rede do Instituto de Pesquisas Hidráulicas da Universidade Federal do Rio Grande do Sul, e mais precisamente os servidores deste Instituto e 5 bases Wireless – DWL 2100AP. A rede como um todo possui quase 300 computadores, dos quais 280 estão em rede. As bases wireless estão dispostas em diversos prédios do Instituto e ligadas à rede por switchs gerenciáveis de 100 Mbits.

Figura 1.1: Localização do IPH no contexto do Campus do Vale da UFRGS.

Estas bases wireless estão numa rede autenticada da UFRGS, onde tão somente usuários com vínculo à UFRGS podem acessar, mediante seu número de registro único como login e senha do portal (figura 1.2 e 1.3), mas também pode ser emitido um ticket para usuários temporários que precisem utilizar este serviço de rede (figura 1.4). Desta maneira se tem um controle maior, podendo saber que micros estão conectados à rede wireless, podendo fazer bloqueios individuais à máquina com problemas (por exemplo, vírus), não precisando bloquear o Access Point como um todo e, todos seus usuários conectados nele naquele momento do incidente de segurança.

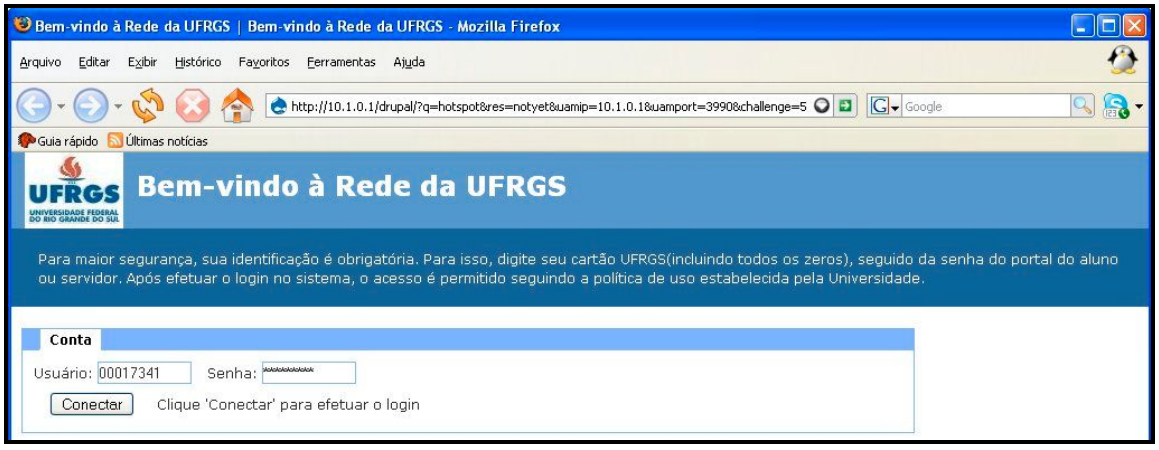


Figura 1.2: Autenticação na rede wireless da UFRGS.

Figura 1.3: Conexão estabelecida.

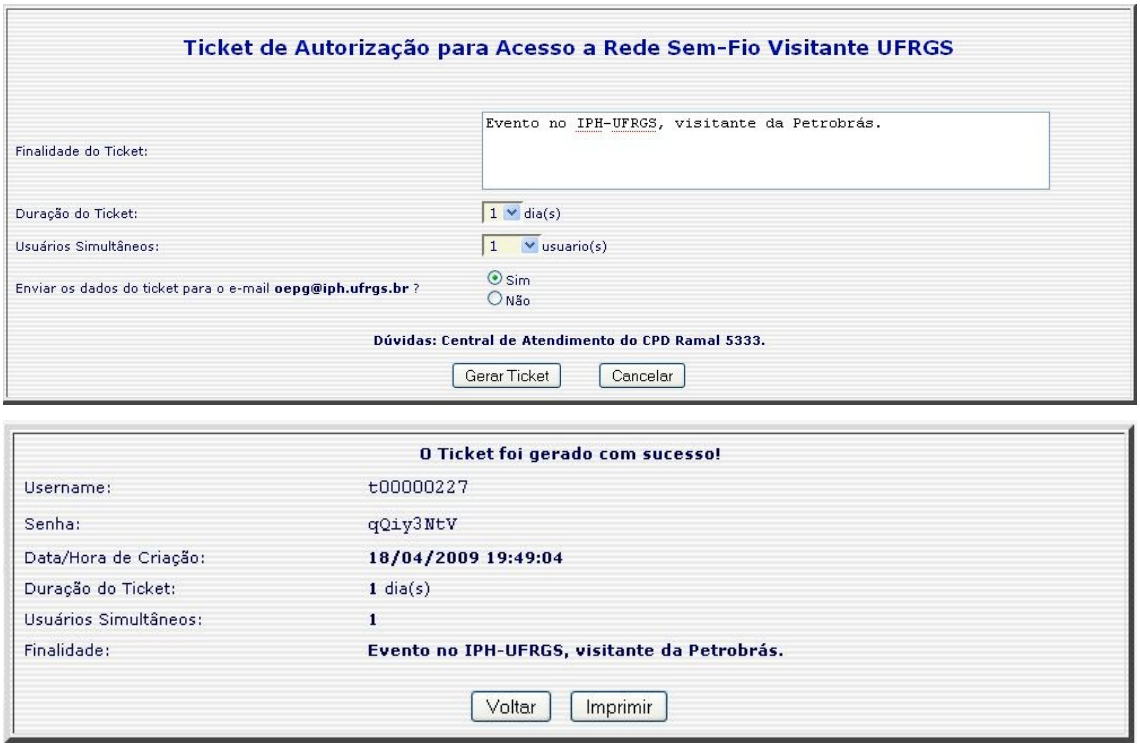


Figura 1.4: Solicitação e missão de ticket de acesso temporário.

2 GERENCIAMENTO SNMP

O gerenciamento de redes é uma tarefa complexa, envolvendo a configuração, monitoração e controle dos mais variados componentes de hardware e software. Suas principais funções envolvem a configuração e monitoração do desempenho dos equipamentos, o controle de acesso aos recursos da rede, a contabilização dos recursos disponíveis e custos envolvidos na sua utilização e a localização e correção dos problemas (falhas) ocorridos nas redes.

Para estas atividades, a habilidade de adquirir informações sobre os equipamentos envolvidos e as mudanças ocorridas nestes é um fator fundamental. Assim, para manusear a grande quantidade de dados provenientes da ampla gama de tipos de equipamentos existentes nas redes, o uso de protocolos de gerenciamento padronizados específicos para o gerenciamento de redes se torna necessário. O protocolo SNMP (Simple Network Management Protocol) é um protocolo desenvolvido para este fim, permitindo o acesso às informações em ambientes com equipamentos de múltiplos fabricantes.

O modelo de gerenciamento de redes baseado em SNMP refere-se a um grupo de padrões para gerenciamento de redes, incluindo o protocolo, o conjunto de objetos de dados e a especificação da estrutura de dados. Ele foi adotado como um padrão para redes TCP/IP quase na década de 90 e é largamente utilizado. Do modelo de gerenciamento SNMP são incluídos os seguintes elementos:

· Uma ou mais estações de gerenciamento contendo aplicações de gerenciamento (gerentes).

· Um ou mais nodos gerenciados contendo uma entidade de processamento denominada agente

· As informações de gerenciamento (denominadas objetos) presentes em cada nodo gerenciado (agente), que descrevem a configuração, o estado, as estatísticas e controlam as ações do nodo gerenciado.

O conjunto de dados que os nodos gerenciados suportam é definido através de especificações denominadas MIB (Management Information Base).

3 AGENTES SNMP

O SNMP é um protocolo de gerenciamento típico de redes TCP/IP, da camada de aplicação que facilita o intercâmbio de informação entre os dispositivos de rede. O agente SNMP consulta as informações armazenadas por uma base de dados denominada MIB (*Management Information Base*), que procura abranger todas as informações necessárias para a gerência da rede. O SNMP também possibilita aos administradores de rede gerir o seu desempenho, como também encontrar e resolver problemas, e planejar o crescimento desta [RFC 1156].

Os comandos para as consultas são simples e baseados no mecanismo de busca e alteração. Neste mecanismo estão disponíveis as operações de manipulação de um objeto, de obtenção dos seus valores e suas variações. Sua utilização com um número limitado de operações, torna o protocolo de fácil implementação, simples, estável e flexível.

O funcionamento do SNMP é baseado em dois dispositivos denominados agentes e gerentes. No contexto deste trabalho um gerente refere-se ao *host* que recebe todas as informações a respeito dos *hosts* gerenciados e um agente é o *host* que permite estender o gerente ou também ser gerenciada. Em outras palavras um *host* agente permite distribuir a aplicação de gerenciamento. Cada *host* gerenciado deve possuir um agente SNMP e uma base de informações MIB [RFC 3413].

Os agentes SNMP fornecem respostas, somente às solicitações do sistema de gerenciamento, com exceção de uma situação anormal, onde uma mensagem TRAP é enviada. Dentre as operações disponíveis, os seguintes comandos podem ser utilizados:

• *get*: um valor de um contador específico é solicitado ao agente SNMP;

• get-next: o valor do próximo contador é solicitado ao agente SNMP;

• *walk*: solicita os valores de todos os contadores de um objeto SNMP;

• *trap*: envia ao sistema de gerenciamento informações sobre situações anormais interceptadas pelo agente SNMP;

• *set*: instrui o agente para alterar um parâmetro configurável.

O SNMP possui falhas no quesito segurança. Intrusos na rede podem ter acesso a informações dos dispositivos, além de podem modificar as variáveis dos mesmos alterando o funcionamento da rede.

Com a expansão do SNMP para SNMPv2 foi possível melhorar a segurança desse protocolo adicionando mecanismos como: privacidade de dados (prevenir os intrusos de ganhar acesso a informação levadas pela rede), autenticação (impedir os intrusos de

enviar falsos dados pela rede), e controle de acesso (que restringe acesso de variáveis particulares a certos usuários, removendo assim a possibilidade de um usuário derrubar a rede).

SNMP é um protocolo essencial, uma vez que desempenha atividades importantes na monitoração das redes podendo realizar um trabalho preventivo como a detecção do aumento do número de pacotes errados nos roteadores, como até corretivo, alterando objetos nos dispositivos da rede.

É um protocolo popular, pois além de ser largamente utilizado nas redes em todo mundo, muitos aplicativos foram criados para uso do SNMP. É rápido, pois o gerente não precisa fazer um login no agente e estabelecer uma conexão TP/IP para receber seus dados, já que as mensagens são enviadas em pacotes UDP.

Neste trabalho a versão utilizada do protocolo SNMP é a versão 2 ou SNMPv2.

4 BASE DE INFORMAÇÃO GERENCIAL - MIB

A MIB (Management Information Base) é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para a gerência da rede. O RFC - Request For Comment 1066 apresentou a primeira versão da MIB, a MIB I em Agosto de 1998. Este padrão explicou e definiu a base de informação necessária para monitorar e controlar redes baseadas na pilha de protocolos TCP/IP. Ela foi atualizada pela RFC 1156 [RFC 1156] em maio de 1990 e posteriormente a versão da MIB-II foi publicada na RFC 1213 [RFC 1213] em maio de 1991, aumentando de 114 para 171 o número de objetos da versão anterior.

Basicamente são definidos três tipos de MIBs: MIB II, MIB experimental, MIB privada. A MIB II, que é considerada uma evolução da MIB I, fornece informações gerais de gerenciamento sobre um determinado equipamento gerenciado. Através das MIB II podemos obter informações como: número de pacotes transmitidos, estado da interface, entre outras. A MIB experimental é aquela em que seus componentes (objetos) estão em fase de desenvolvimento e teste, em geral, eles fornecem características mais específicas sobre a tecnologia dos meios de transmissão e equipamentos empregados.

MIB privada é aquela em que seus componentes fornecem informações específicas dos equipamentos gerenciados, como configuração, colisões e também é possível reinicializar, desabilitar uma ou mais portas de um roteador.

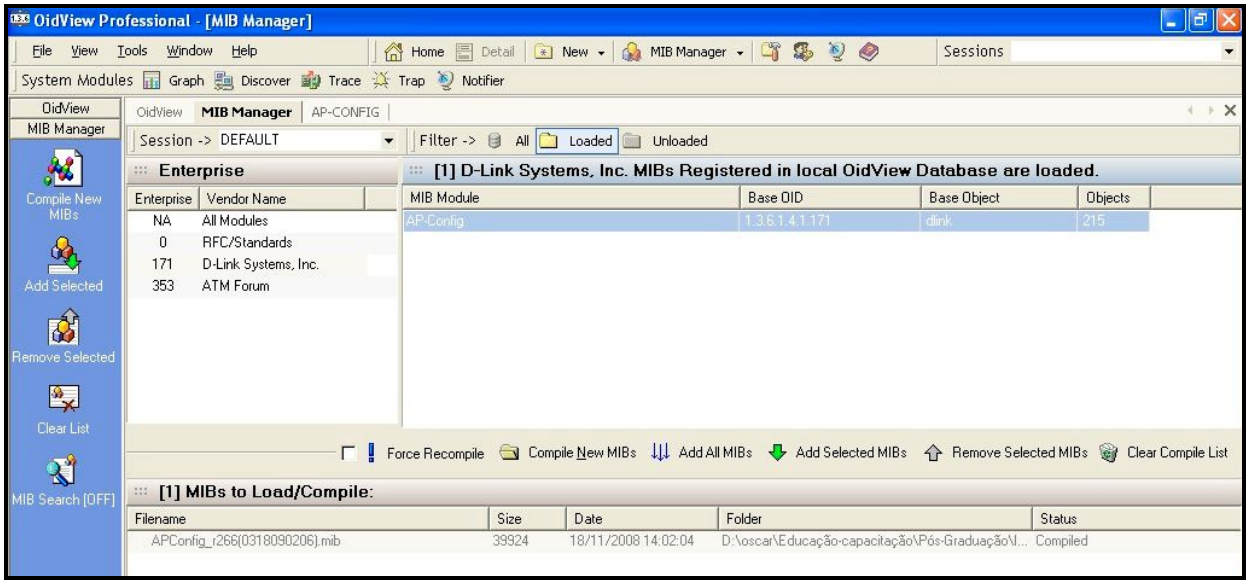


Figura 4.1: MIB Wireless do dispositivo DWL 2100AP

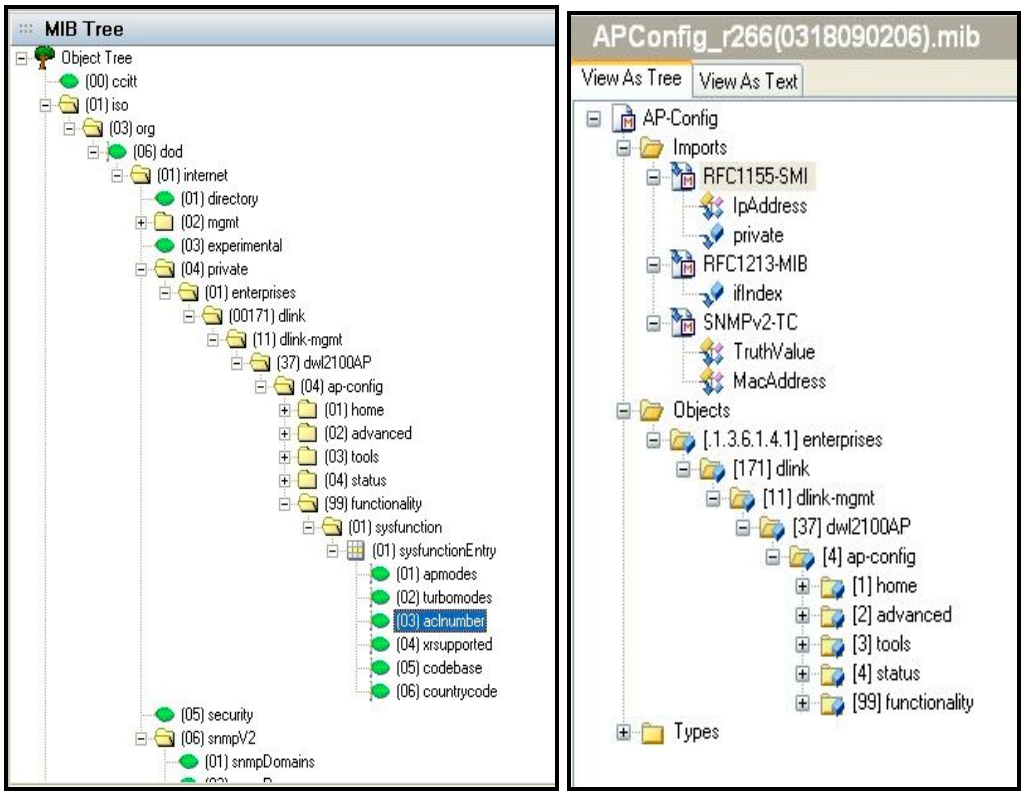


Figura 4.2: Árvore MIB Wireless do dispositivo DWL 2100AP.

A MIB SNMPv2 define objetos que transcrevem o comportamento de uma entidade SNMPv2. Esta MIB consiste de cinco grupos:

a) Grupo Estatística SNMPv2: uma coleção de objetos provendo instrumentação básica da entidade SNMPv2.

b) Grupo Estatística SNMPv1: uma coleção de objetos provendo instrumentação básica da entidade SNMP que também implementa SNMPv2.

c) Grupo Recursos do Objeto: uma coleção de objetos permitindo uma entidade SNMPv2 atuar em uma função de agente para descrever sua configuração dinâmica dos recursos do objeto.

d) Grupo Traps: uma coleção de objetos que permitem cooperação com entidades SNMPv2 quando atuando em uma função de gerência para ser configurado para gerar trap PDU SNMPv2.

e) Grupo set: uma coleção de objetos que permite que várias entidades possam agir em conjunto, todas em função de gerência, para coordenar o uso operação set sobre SNMPv2.

5 SOFTWARE DE MONITORAMENTO

Para fazer o monitoramento dos dispositivos de rede, utilizando os agentes SNMP, foram pesquisados diversos softwares, tanto para ambientes Linux como para ambiente Windows, tentando avaliar o que de melhor cada um tem a oferecer, e que se adaptasse aos anseios de gerenciamento destes dispositivos. Os mesmos estão instalados em 2 servidores (um Open Susse Linux e outro Windows Server), sendo que todos os listados estão sendo avaliados ainda, para conhecer melhor seu potencial, para depois poder determinar se vai ser comprado algum dos pagos, ou até usar algum na sua versão aberta e free.

Linux: Zenoss (Core), Cacti, Zabix e Wifi Manager.

Microsoft: PRTG Graphic monitor e PRTG Traffic Grapher, SNMPc, Pagglo, Servers Check, além de ferramentas Mib Browsers para edição da MIB das bases Wireless.

Ainda não se tem uma decisão sobre quais deles irão monitorar a rede no futuro, porém, o Wifi Manager tem se mostrado mais completo, assim como o PRTG Traffic Grapher tem se mostrado interessante na parte gráfica de visualização dos dados, além de ter uma ferramenta de importação de mibs para poder visualizar no Traffic Grapher (figura 5.1).

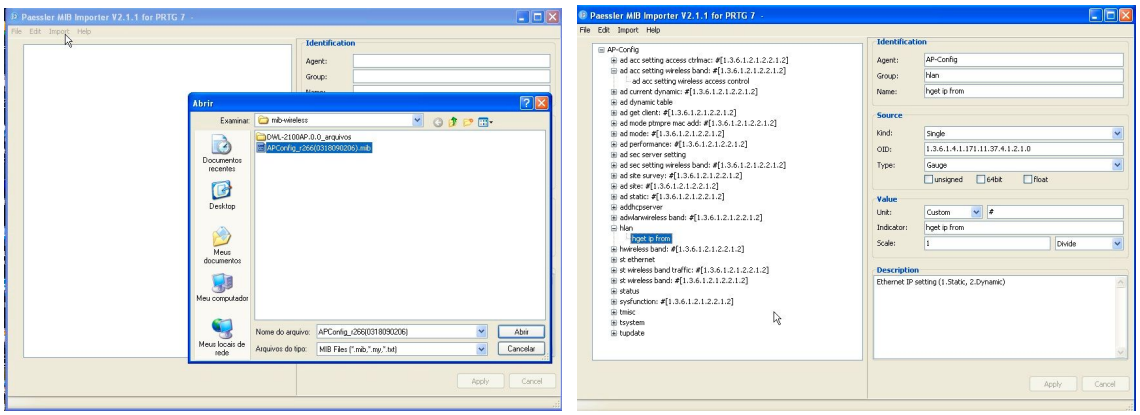


Figura 5.1: Importação de mibs e exportação para o Traffic Grapher.

Todos os dados de monitoramento dos servidores e bases wireless serão devidamente analisados num trabalho posterior, mas nos anexos deste trabalho encontram-se telas dos diversos softwares em avaliação nesta rede. Destaca-se a seguir maiores detalhes dos 3 softwares escolhidos inicialmente para monitoramento das bases wireless do Instituto.

PRTG Traffic Grapher

PRTG Traffic Grapher é um software para ambiente Windows fácil de usar, serve para monitorar uso de banda, assim como diversos outros parâmetros de rede como memória e a utilização da CPU. Fornece aos administradores de sistemas dados e tendências de utilização de linhas, roteadores, firewalls, servidores, e muitos outros dispositivos de rede.

O acompanhamento ou monitoramento ajuda a otimizar a rede, e este software oferece informações de uso de banda e o uso da rede de dados, ajudando a otimizar a eficiência das redes. O entendimento da largura de banda e o consumo de recursos é a chave para ter uma melhor gestão da rede, pois isto permite:

- Evitar estrangulamentos na banda, diminuindo o desempenho de servidores da rede.
- Saber o que servidores de aplicações utilizam de recursos de rede.
- Plano de upgrades da infraestrutura estratégica .
- Entregar uma melhor qualidade de serviço aos usuários, podendo ser proativo nesta resposta.
- Reduzir custos através da compra de banda e hardware de acordo com a carga real.

O PRTG Traffic Grapher é executado em uma máquina Windows na rede durante 24 horas, todos os dias e constantemente registra os parâmetros de utilização de rede. Os dados gravados são armazenados em uma base de dados interna para referência posterior. Utilizando uma interface fácil de usar, podendo configurar os sensores monitorados, bem como criar relatórios de utilização. Para o acesso remoto, o Traffic Grapher PRTG vem com um “built-in” para o servidor web fornecer o acesso a gráficos e tabelas.

São suportados todos os métodos comuns para Aquisição de dados de uso de rede, como os listados a seguir:

- **SNMP:** Simple Network Management Protocol é o método básico de coleta de banda e o uso da rede de dados. Ela pode ser usada para monitorar uso de banda de roteadores e switches porta a porta, bem como dispositivo como memória, CPU, etc.
- **Packet sniffing:** Com seu Packet Sniffer o PRTG pode inspecionar todos os pacotes de rede de dados que passam numa placa de rede. Pode-se acompanhar quer apenas o tráfego da máquina executando PRTG ou todo o tráfego da rede.
- **Netflow:** o protocolo de Netflow é suportado pela maioria dos roteadores para medir o uso de banda. Embora sendo o mais complexo tipo de configurar-se, também é o mais poderoso método adequado para redes de alto tráfego.

Funcionalidades Base

- Suporta aquisição de dados através de SNMP, pacotes sniffing, ou protocolo Netflow.
- Classifica o tráfego de rede por endereço IP, protocolo e outros parâmetros.
- Trabalha com a maioria dos switches, roteadores, firewalls e outros dispositivos de rede.
- Fácil instalação com apenas alguns cliques no Windows 2000/XP/2003/Vista.
- Funcionamentos como serviços NT.
- O mecanismo de acompanhamento é capaz de monitorar até vários milhares de sensores.
- O monitoramento dos dados pode ser acessado através de uma interface gráfica Windows e um "front-end" baseado na web.
- Intuitiva interface de configuração e recuperação de dados.
- Servidor web integrado para acesso remoto.
- Os resultados são apresentados em diversos gráficos e tabelas.

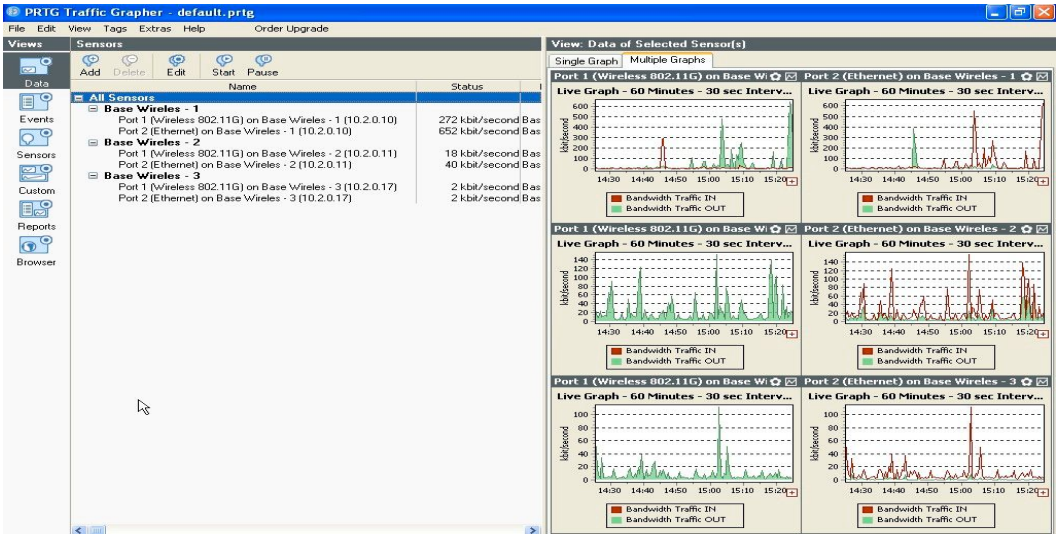


Figura 5.2: Gráficos do comportamento dos dispositivos monitorados

O PRTG Traffic Grapher tem uma interface de visualização dos dispositivos gerenciados, que permite visualizar graficamente o comportamento do mesmo, mostrando o gráfico de comportamento em tempo real, 24 horas, últimos 30 dias e de 365 dias (um ano), como mostra a figura 5.2; todos os itens são editáveis fazendo com que possa editar porções do gráfico, individualizar algum período desejado, assim como mudar características de monitoramento para ter uma resposta personalizada.

Além da visualização do comportamento de maneira gráfica, pode-se verificar em tabelas os dados de entrada e saída, porcentagem de cobertura, entre ouras variáveis, como mostra a figura 5.3.

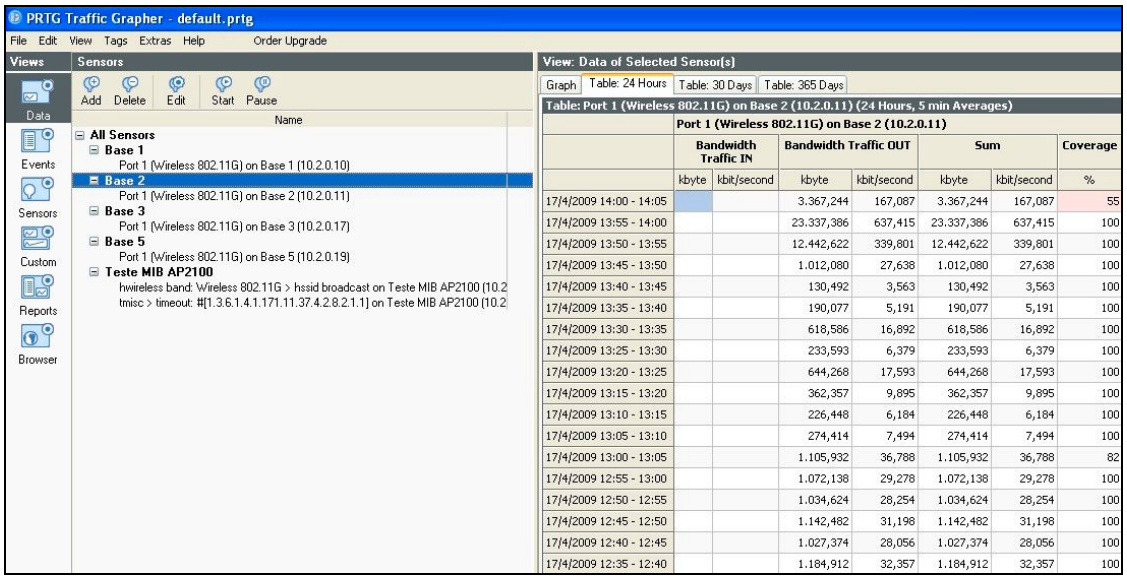


Figura 5.3: Estatísticas do monitoramento

Uma questão importante é a de criação de relatórios automáticos com o comportamento da rede e seus dispositivos; o Traffic Grapher permite especificar que tipo de relatório deve ser gerado, quando deve ser gerado e de que maneira enviar o mesmo. A figura 5.4 mostra um e-mail enviado pelo sistema com um arquivo em pdf anexo, contendo o relatório dos dispositivos monitorados.

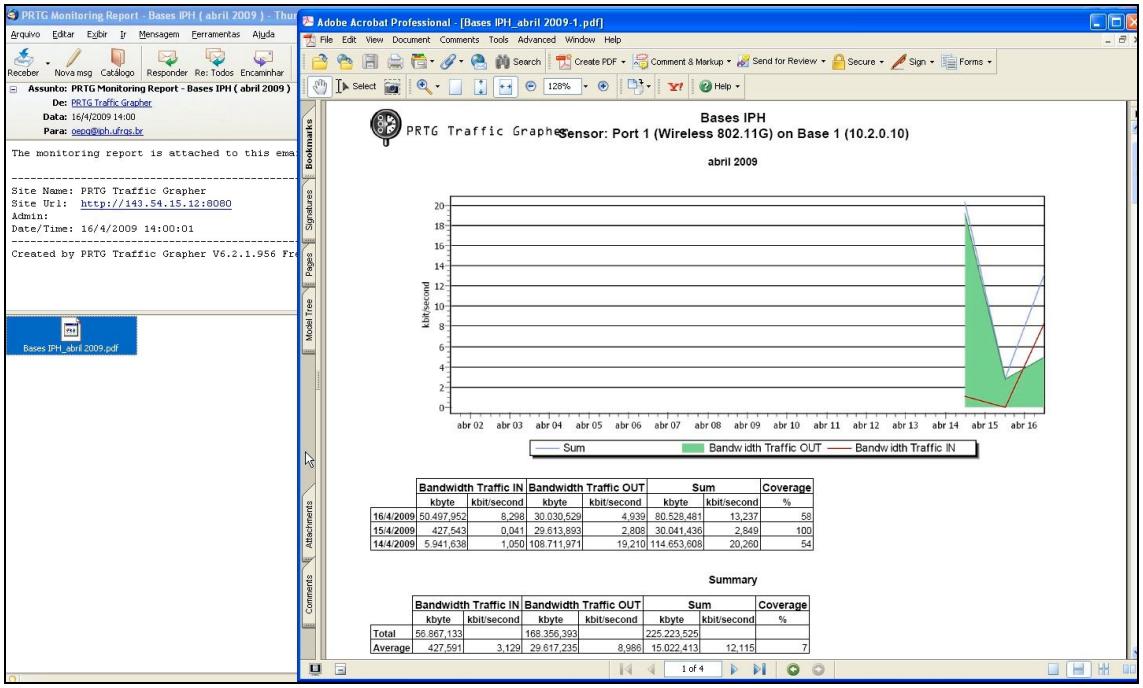


Figura 5.4: Sistema de reporte via e-mail do monitoramento.

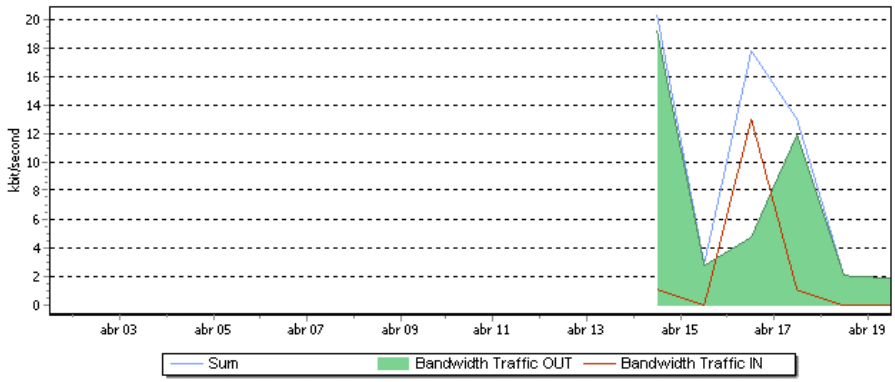
No caso especifico do monitoramento ao qual este trabalho se refere, o arquivo anexo contém 4 páginas de relatórios (gráficos e tabelas) de cada dispositivo monitorado, sendo mostrado na figura 5.5 o relatório de uma das bases.

PRTG Traffic Grapher

Bases IPH

Sensor: Port 1 (Wireless 802.11G) on Base 1 (10.2.0.10)

abril 2009



	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
19/4/2009	0,000	0,000	11.857,150	1,871	11.857,150	1,871	60
18/4/2009	0,000	0,000	21.569,448	2,046	21.569,448	2,046	100
17/4/2009	11.679,153	1,110	125.517,350	11,925	137.196,503	13,035	100
16/4/2009	135.252,578	12,974	50.015,506	4,802	185.268,084	17,776	99
15/4/2009	427,543	0,041	29.613,893	2,808	30.041,436	2,849	100
14/4/2009	5.941,638	1,050	108.711,971	19,210	114.653,608	20,260	54

Summary

	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
Total	153.300,912		347.285,317		500.586,229		
Average	37.241,501	2,529	56.904,738	7,110	47.073,120	9,640	17

Created with PRTG Traffic Grapher ©2003-2008 Paessler AG (www.paessler.com)

15/4/2009 14:25:52 - Page 1 of 4 - PRTG Traffic Grapher V6.2.1956 ©2008 Paessler AG (www.paessler.com)

Figura 5.5: Reporte do sistema via e-mail.

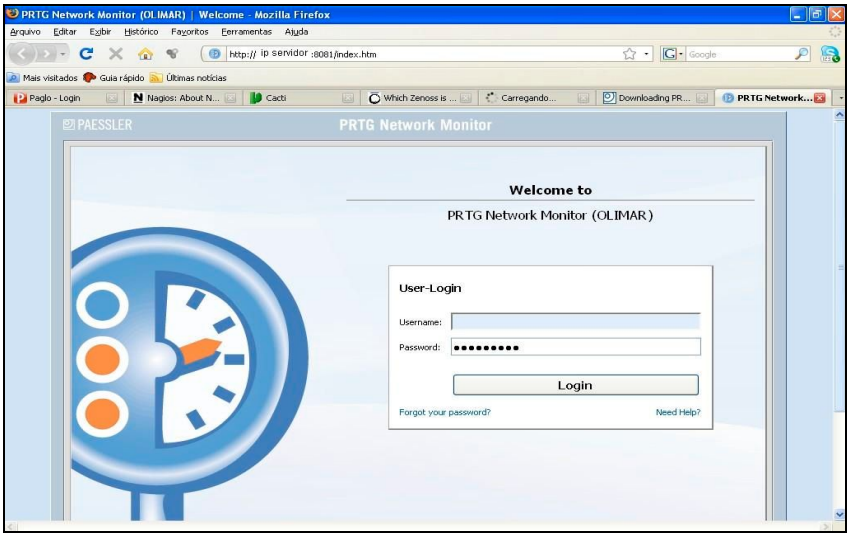
PRTG Network Monitor

PRTG Network Monitor provê desempenho e uso de monitoramento para LAN, WAN e redes VPN com até 30.000 sensores. O servidor web embutido e a interface web AJAX, facilitam o trabalho, com características poderosas como a descoberta de rede completamente automática, avaliação de dispositivos e sistema de configuração de sensores, como também alertas informando e traçando características da rede. A opção utilizada neste trabalho é a versão free do produto, que permite o monitoramento de 10 sensores, a partir da versão 7.1 liberada em 21/04/2009 a versão free monitora 20 sensores.

Instalação/Configuração

PRTG Network Monitor foi aperfeiçoado para uma instalação, configuração e uso fáceis. Permite montar uma rede completa monitorando toda uma situação particular em pouco tempo. Algumas funções:

- Descoberta de rede automática.
- Dispositivos modelos pré-configurados com sensores para vários dispositivos.
- Interfaces interativas e customizáveis para uma usabilidade aperfeiçoada
- Configuração é organizada dentro de uma árvore hierárquica com herança de configurações.



O PRTG Network Monitor pode ser usado nas seguintes situações:

· monitorar e alertar para serviço ativo/inativo (uptimes/downtimes) ou servidores lentos.

· monitorar largura de banda usada e uso de dispositivos de rede.

· monitorar uso de sistema (carga de CPU, memória livre, espaço de disco livre, etc), como mostra a figura 5.6.

· classificar tráfico de rede por origem/destino e conteúdo.

· descobrir atividade incomum, suspeita ou maliciosa com dispositivos ou usuários.

· controlar acordos de SLA (Service Level Agreement), que é um documento que define a relação entre duas partes: o provedor e o receptor.

· descobrir e avaliar dispositivos de rede.

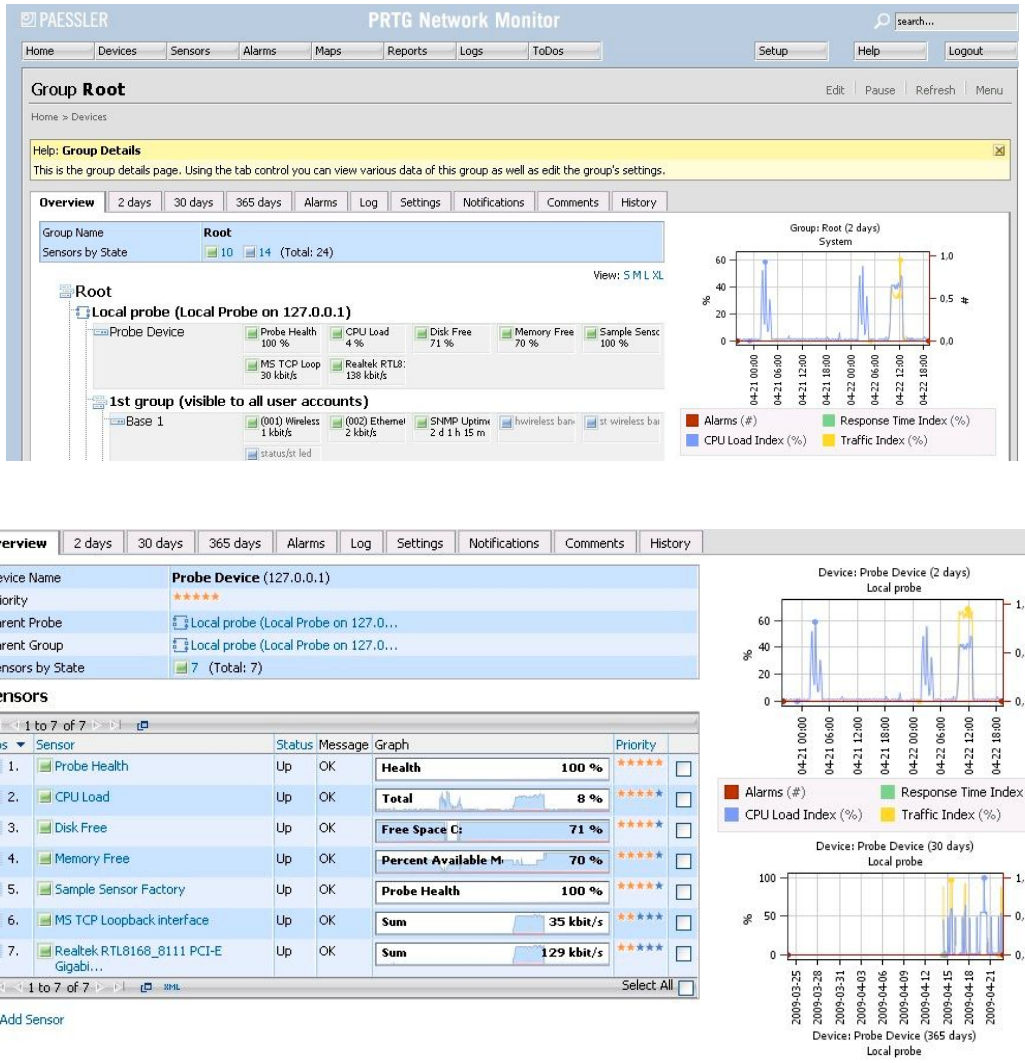


Figura 5.6: Inventário do uso do sistema.

WiFi Manager

O WiFi Manager é um programa para gerenciamento centralizado e de segurança de conexões Wireless (sem fio). Este software potencializa a segurança de LANS monitorando continuamente a rede e o ambiente, podendo detectar quase todas as ameaças a conexões sem fio. Pode-se também configurar pontos de acesso com poucos cliques. Ou seja, com o WiFi Manager, se tem controle completo sobre os dispositivos sem fio.

Para promover a segurança de sua LAN sem fio, o WiFi Manager detecta e bloqueia ameaças, intrusos e vulnerabilidades. O programa também monitora associações, uso de banda, rádio, parâmetros de segurança, tráfego e tempo de resposta.

Usando o WiFi Manager, pode-se gerar relatórios diários, semanais e mensais; também é possível preparar relatórios sobre um espaço de tempo determinado pelo usuário. E estes informativos podem ser exportados para outros formatos, como o PDF, para fácil leitura e impressão.

O WiFi Manager também configura pontos de acesso. É possível alterar configurações básicas, de rádio, de segurança e de serviços; também é possível definir controles de acesso. Os administradores podem ainda utilizar “templates” pré-determinados de configuração, oferece atualização de “firmware” para os pontos de acesso, sendo que estas podem ser agendadas.

Instalação

A instalação do WiFi Manager segue o padrão de aplicações Windows. Depois de concluída, deve-se escolher qual versão do programa vai usar, sendo que as opções são:

- Trial Edition: pode usar todos os recursos do programa durante 30 dias;
- Free Edition: pode usar o programa por tempo indeterminado. No entanto, só poderá gerenciar três pontos de acesso e 10 computadores;
- Registered Edition: versão completa, paga, que requer registro.

Depois de escolher a versão, aparecerá também uma tela de registro, mas, para a versão Free ele é opcional. No caso deste trabalho foi usada a versão Free Edition que suporta o monitoramento de até 10 dispositivos, pois tão somente 5 bases serão monitoradas nesta fase experimental.

Para começar a usar

Na primeira vez que é executado o programa, um assistente guia para informar as configurações básicas sobre a rede sem fio. Em seguida, é direcionado para o site da administração, onde deverá ser informado o login e uma senha, inicialmente os dados de acesso são "admin", tanto para nome de usuário e senha. Após é acessado o site de gerenciamento e poderá ser monitorada a rede sem fio e suas bases.

Recursos

- Gerenciamento de Wlan
- Configuração de Pontos de Acesso
- Atualização do Firmware das bases
- Detecção de intrusos nas bases
- Bloqueio de intrusos nas bases
- Detecção de Cliente intruso
- Bloqueio de cliente intruso
- Detecção de vulnerabilidades na WLAN

Em linhas gerais o Wifi Manager oferece uma série de opções de gerenciamento contidas ou organizadas como mostra o menú acima, e detalhadas a seguir:

Home:

Neste painel do WiFi Manager, dá um rápido panorama sobre a saúde global da rede monitorada. Sob este guia, do lado esquerdo, pode-se encontrar links rápidos para o Access Point, as redes com e sem fios podem ser visualizadas a partir deste link. Também podem ser acessados alguns relatórios como a associação corrente, sinal da qualidade de rede, segurança, etc.

Fault:

O guia Fault, dá detalhes sobre as várias deficiências ou lacunas na segurança das redes. Este guia dá uma vista de todos os alarmes das informações levantadas como Status, Data / Hora, Detalhes de Alarme, Fontes e qualquer nota técnica marcada nos alarmes. No lado esquerdo estão as ligações para o cenário de falhas, onde temos opções para as seguintes opções: “alarm setting, notification profile, watchlist settings, alarms category: WatchList Alarms, Recent Alarms, Active Alarms, Acknowledged Alarms, Info Events, Intrusion,Vulnerability, Denial Of Service, Operational Performance”.

Inventory:

O guia Inventário contém um resumo sobre os vários dispositivos na rede. No âmbito deste guia, pode-se encontrar uma lista de todos os dispositivos na rede e os detalhes destes dispositivos em suas respectivas categorias. A lista mostra o opção de adicionar uma rede ou dispositivo, detalhes de dispositivos, detalhes da rede e SSID.

Configuration:

Este guia ajuda a configurar os pontos de acesso e também envia a configuração para outros access points quando precisar adicionar um grande número deles. Existem também opções para ajudar a atualizar o firmware dos Access Point. Para cada

configuração de um AP também pode-se aplicar a mesma em um grupo de APS. Este guia tem a seguintes opções: “security channels, Access Control setting option e the configuration of Firmware details”.

Reports:

O Guia Relatórios É destinado a ajudar o usuário a visualizar os detalhes obrigatórios num formulário bem organizado. Os diferentes relatórios que estão disponíveis incluem “Bandwidth Report”, “Rádio Reports”, Segurança e relatórios detalhados de tráfego. Esses relatórios podem ser visualizados seleccionados para período de tempo e são complementadas por gráficos de mais fácil compreensão.

Admin:

O guia Admin é de vital importância, uma vez que atua como o próprio console a partir de onde pode-se configurar o WiFi Manager. Aqui podem ser feitas operações como a criação de novo usuário, acrescentar nova rede / dispositivos de rede, configurações do servidor de correio e também configurar o seu cliente para detectar intrusos.

Support:

O guia Suporte é um lugar onde pode-se fazer uso das diferentes formas de ajuda disponíveis. As opções disponíveis são: Solicitar apoio, Userforum, dicas de solução de problemas e números de telefone.

Other Links:

No guia outros links, existem links para o fórum, Ajuda. Estes links podem ser utilizados para enviar questões relativas ao WiFi Manager, opiniões e sugestões.

Resumidamente vão ser apresentadas algumas funcionalidades do software com suas respectivas telas, para desta maneira poder mostrar a potencialidade desta ferramenta de monitoramento. O Wifi Manager permite cadastrar redes e dispositivos de rede, classificando os mesmos em amigáveis, confiáveis e intrusos, associa a cada dispositivo os usuários conectados no mesmo, no caso especifico de Access Points (AP) mostra o número de micros ligados a cada AP, seus respectivos endereços Macs e a qualidade do sinal de conexão, como mostra a figura 5.7.

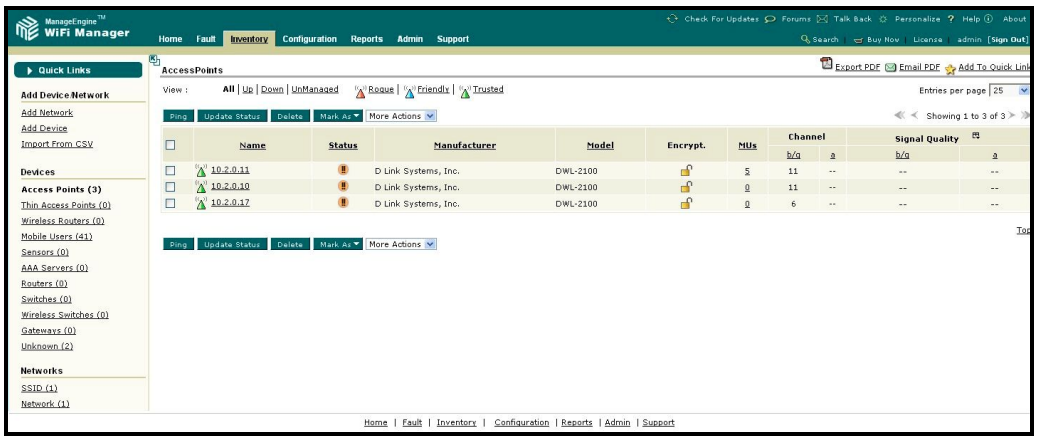


Figura 5.7: Inventário de dispositivos de rede.

Uma vez selecionado um dispositivo, é mostrada uma descrição geral do mesmo, a qualidade do sinal em %, os micros associados a este dispositivo, assim como o tempo de conexão de cada um deles (Figura 5.8 e 5.10).

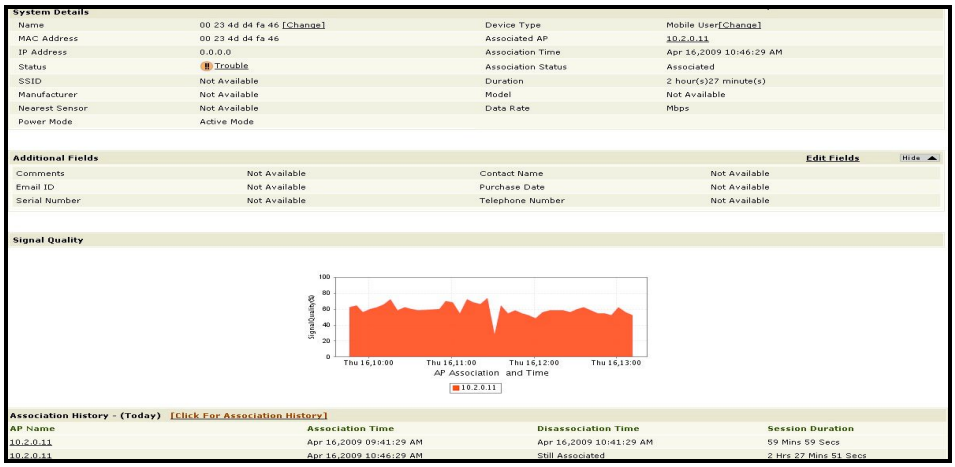


Figura 5.8: Detalhes do sistema e qualidade de sinal.

Também é possível determinar o número de usuários conectados em cada dispositivo, sua distribuição durante o dia, endereços Mac de cada um, número mínimo e máximo de conexões, assim como um reporte específico por cada máquina associada, como mostra a figura 5.9.

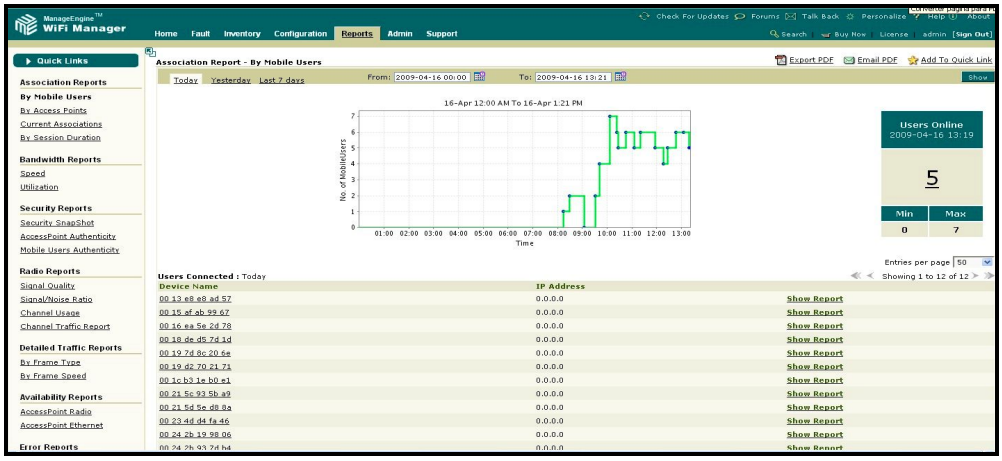


Figura 5.9: Interface de usuários móveis.

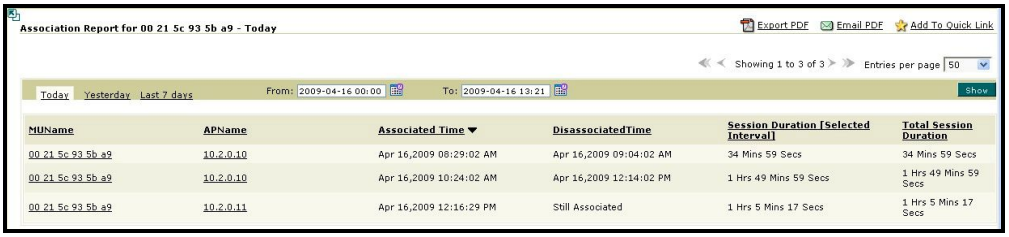


Figura 5.10: Tempo de sessão dos dispositivos móveis.

O Wifi Manager permite associar graficamente o número de usuários por Access Point, individualizando o mesmo pelo reporte diário, do dia anterior ou 7 dias antes, mas também permite elaborar o relatório gráfico por períodos pré-determinados, podendo exportar via e-mail ou em arquivo pdf o relatório gerado, como mostra a figura 5.11.

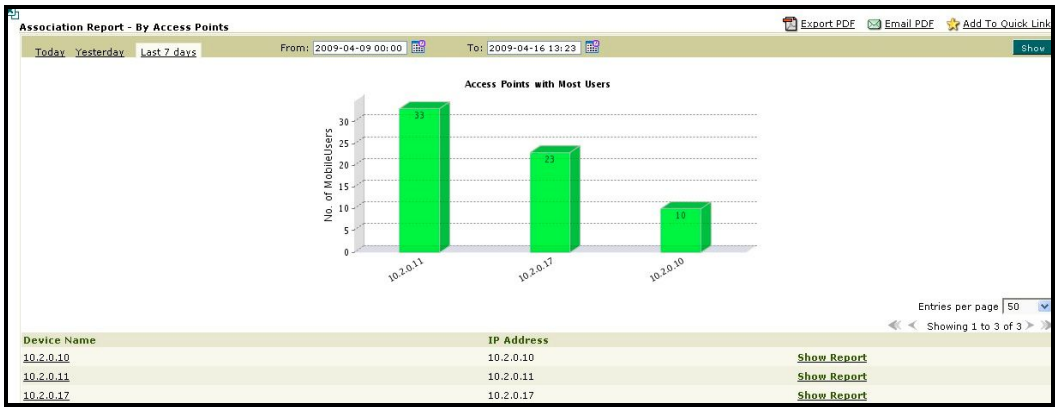


Figura 5.11: Número de usuários conectados por AP.

Uma das outras funcionalidades do software é poder estabelecer por faixas de tempo de conexão, o número de usuários por faixa ou intervalo de tempo conectado. Ao mesmo tempo associa o endereço Mac da máquina conectada, sua associação a um Acces Point, inclusive mostrando um aspecto importante que os AP2100 da D-Link tem, que é a conexão entre bases da mesma rede, desde que tenham uma superposição na área de cobertura, fazendo com que usuários possam andar percorrendo uma área dentro de um prédio, e quando sai da área de alcance de um AP, passa para outro melhor localizado sem desconectar, esta situação é percebida pelo software, pois relaciona tempo de conexão ao IP do Access Point, mostrando que um mesmo micro vai "caminhando" pela área de cobertura dos dispositivos AP. Isto é exemplificado nas figuras 5.12 e 5.13, sendo que nesta última permite ainda gerar algumas estatísticas gráficas.

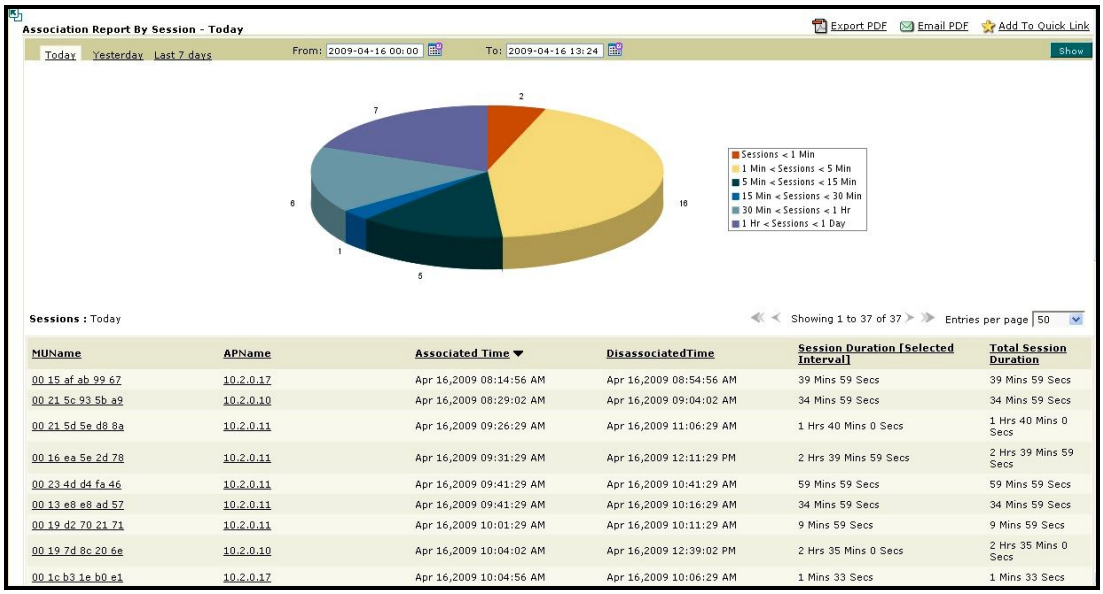


Figura 5.12: Número de usuários conectados por faixa de tempo.

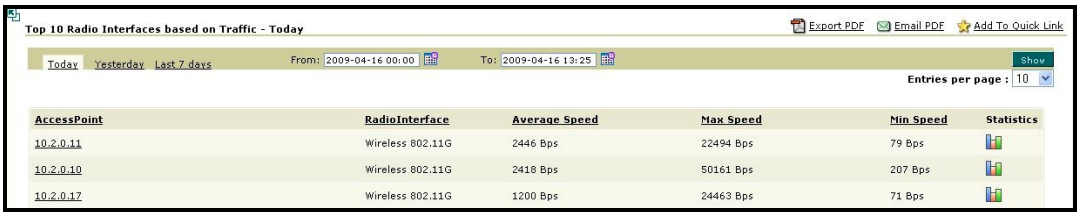


Figura 5.13: Estatísticas gerais por AP.

No menú relacionado a falhas e alarmes, o Wifi Manager permite configurar uma série de itens como: Intrusion, Operational, Performance, Availability, Vulnerability, Dos e Sniffers, sendo que cada uma destas opções é desmembrada em diversas novas opções de configuração. Após a configuração nesta seção "Fault" (figura 5.14) poderemos obter, por exemplo, estatísticas de erros de pacotes como mostra a figura 5.15.

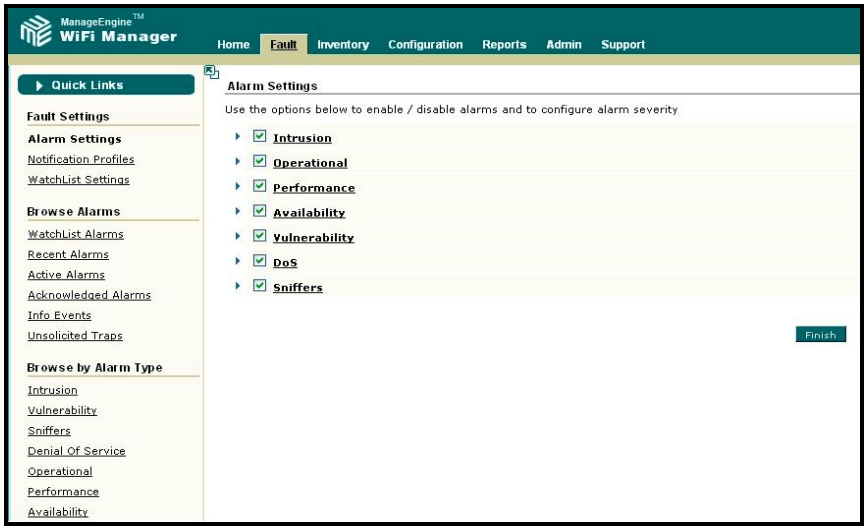


Figura 5.14: Configuração do sistema de falhas.

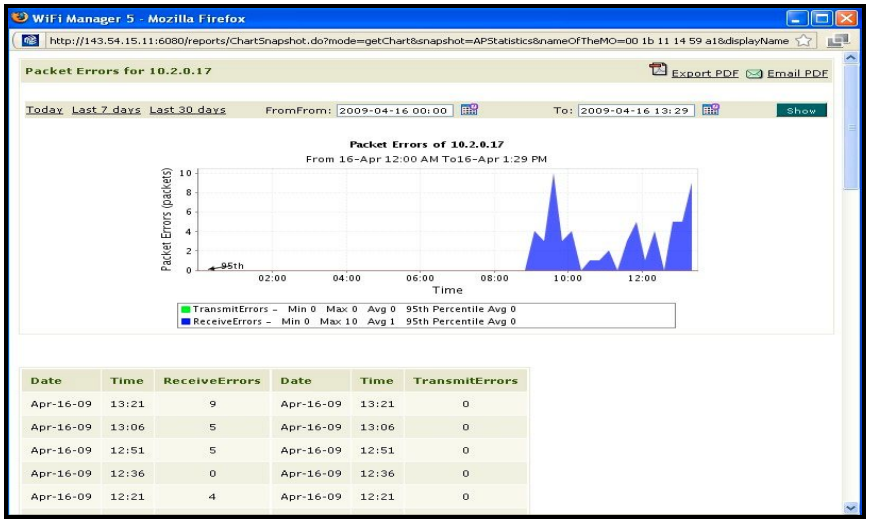


Figura 5.15: Erro de pacotes por AP.

6 CONFIGURAÇÃO DOS AGENTES SNMP

Nas estações a monitorar, primeiramente deve-se instalar o agente SNMP, para que desta maneira o servidor posa contatar a estação, e desta maneira poder receber os parâmetros de interesse para monitorar.

Em estações Windows os passos são os seguintes:

- Painel de Controle/Ferramentas Administrativas
- Serviços/Serviço SNMP
- Na aba **Interceptações** digita o nome da Comunidade SNMP
- Na aba **Segurança** estabelece direitos para a comunidade e define de que hosts vai poder receber pacotes SNMP.
- Se o serviço SNMP não estiver iniciado, ele deve ser setado para que inicie automaticamente.

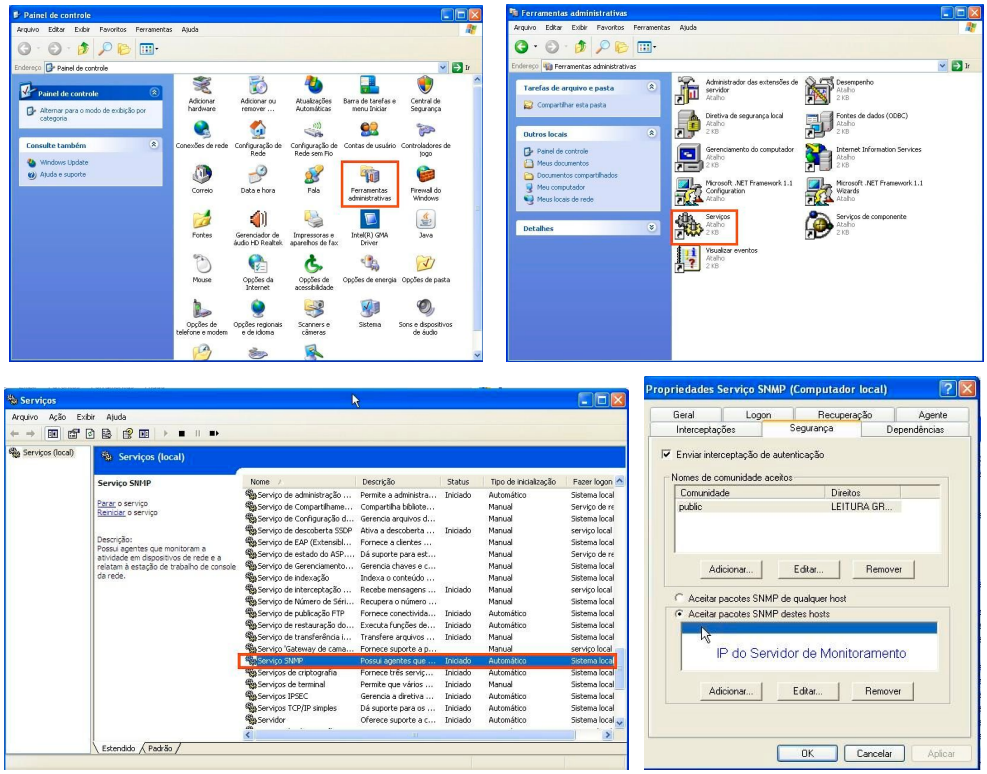


Figura 6.1: Configuração de agentes SNMP em estações Windows.

Em estações Linux os passos são os seguintes:

- Acesse o diretório **/etc/snmp**
- Editar o arquivo **snmp.conf**
- Na linha rocomunity, trocar o IP descrito pelo IP do servidor que coletará as informações desta estação
- Num terminal iniciar o serviço snmpd – **start snmpd**
- Digitar num terminal **chkconfig snmpd on** para que inicie automaticamente o serviço quando a estação for rebootada ou iniciada.

7 CONCLUSÃO

A necessidade de gerenciamento de redes é evidente e tende a crescer à medida que as redes se tornam maiores e mais complexas. Como essas redes estão cada vez mais heterogêneas, uma padronização do protocolo de gerência garante que estas sejam gerenciadas de maneira uniforme.

É imprescindível o monitoramento para a segurança e gerenciamento adequado das redes de computadores, o administrador da rede deve estar completamente a par de toda a situação da rede, para identificar a presença de comportamentos indesejados dentro da rede, tais como: máquinas não autorizadas conectadas aos Access point, consumo excessivo de banda, descoberta de algum vírus que gera um broadcast excessivo em algum sentido na rede (pra dentro dela ou para fora do perímetro da mesma).

Deve-se definir melhor ainda os objetos gerenciáveis relevantes, seus limiares críticos, assim como a geração de alertas quando estes forem ultrapassados.

Uma das principais vantagens do protocolo SNMP é sua simplicidade, podendo ser facilmente implementado numa rede de computadores. Muitos dispositivos já vem de fábrica preparados para o uso do SNMP, sendo fácil a configuração dos agentes na rede.

A rapidez é uma característica do SNMP, pois o gerente não precisa fazer um login no agente e estabelecer uma conexão TP/IP para receber seus dados, além disso o SMTP é muito popular, pois a maioria dos fabricantes de dispositivos para redes os projetam para suportar SNMP, inclusive projetando mibs privadas com objetos particulares de cada dispositivo, como é o caso das bases utilizadas neste trabalho.

Dentro do Instituto de Pesquisas Hidráulicas, ainda não se tem um número significativo de equipamentos móveis acessando simultaneamente as bases, para ter uma boa quantidade de dados para já ter tendências devidamente diagnosticadas, mas com o monitoramento já pode-se ter um perfil de uso dos recursos da rede wireless. A configuração de alertas para alguns limiares críticos permitiu diagnosticar quando as máquinas ultrapassam os mesmos.

Um dos limites de diagnóstico desejado ainda não pode ser estabelecido, que é o número de máquinas conectadas simultaneamente numa mesma base wireless, para começar a comprometer a mesma, ou chegar a ponto de não conectar adequadamente os micros conectados à base; pelo fato de não ter muitos usuários conectados simultaneamente ainda não foi conseguido estabelecer o limite de máquinas conectadas simultaneamente a uma base, sem comprometer a navegação na Internet. Até o limite máximo já estabelecido de 18 conexões simultâneas os equipamentos AP2100 não tem tido problemas em ofertar conexão com a Internet em taxas compatíveis com a

necessidade de trabalhos acadêmicos e de pesquisa, permitindo download nessa situação a pouco mais de 50 Kb, o que é uma taxa razoável, o que não impede o trabalho do usuário.

Em termos de uso da conexão das bases, até o momento a utilização em média foi de 2,30 horas por sessão, o que caracteriza que em geral os usuários ficam um bom tempo com o navegador aberto pelo menos, ou com serviços conectados durante este tempo.

Com o uso destas ferramentas de gerenciamento, permite também elaborar uma lista histórica de máquinas e suas conexões, relacionando com os endereços MAC de cada máquina, podendo assim individualizar usuários autorizados e usuários intrusos – “Rogue users”. Também permite dar subsídios para o planejamento ou re-planejamento da rede, detectando possíveis lugares onde deveria melhorar o sinal ou aumentar mais uma base na mesma área de cobertura, podendo visualizar melhor a demanda por conexões em cada base, definindo assim setores mais críticos ou preferenciais dentro de um prédio ou ambiente.

REFERÊNCIAS

BAYLOR UNIVERSITY. A Case Study of Baylor University´s Wireless Network. USA, 2003. Disponível em:< http://www.utdallas.edu/library/> . Acesso em: abr. 2009.

CACTI. Software. Disponível em: < http://www.cacti.net/ >. Acesso em: out. 2008.

FERREIRA, P. do A. Rede sem Fio. I Workshop em Administração de Redes do Pop-MG. Brasil, 02 e 03/09/2004. Disponível em: <http://www.pop-mg.rnp.br/eventos/wksp2004/trabalhos/redesemfio.pdf>. Acesso em: abr. 2009.

GRANVILLE, L. **Agentes Móveis no Gerenciamento de Redes**. Disponível em: < http://forum.comp.pucpcaldas.br/viewtopic.php?t=139 >. Acesso em: nov. 2008.

HOST MONITOR MIB BROWSER. Software. Disponível em: < http://www.ks-soft.net/hostmon.eng/ >. Acesso em: out. 2008.

MANAGEMENT Information Base for Network Management of TCP/IP based internets: RFC 1156. Disponível em: < http://www.faqs.org/rfcs >. Acesso em: nov. 2008.

MANAGEMENT Information Base for Network Management of TCP/IP-based internets:MIB-II: RFC 1213. Disponível em: < http://www.faqs.org/rfcs >. Acesso em: nov. 2008.

NAGIOS. Software. Disponível em: < http://www.nagios.org/ >. Acesso em: out. 2008.

OID VIEW MIB BROWSER. Software. Disponível em: < http://www.oidview.com/oidview.html >. Acesso em: out. 2008.

PAGLO Network Management. Software. Disponível em: < http://paglo.com/ >. Acesso em: set. 2008.

PRTG Graphic monitor . Software. Disponível em: < http://www.paessler.com/prtg7/download >. Acesso em: set. 2008.

PRTG Traffic Grapher . Software. Disponível em: < http://www.paessler.com/prtg6 >. Acesso em: set. 2008.

SIMPLE Network Management Protocol (SNMP): RFC 1157. Disponível em: < http://www.faqs.org/rfcs >. Acesso em: nov. 2008.

SERVERS CHECK. Software. Disponível em: < http://www.serverscheck.com.br >. Acesso em: ago. 2008.

SIMPLE Network Management Protocol (SNMP) Application: RFC 3413 (Standard 62). Disponível em: < http://www.faqs.org/rfcs >. Acesso em: nov. 2008.

SIMPLE Network Management Protocol (SNMP).Disponível em: < http://www.snmpworld.com/ >. Acesso em: Abr. 2009.

SNMPc. Software. Disponível em: < http://www.castlerock.com/ >. Acesso em: out. 2008.

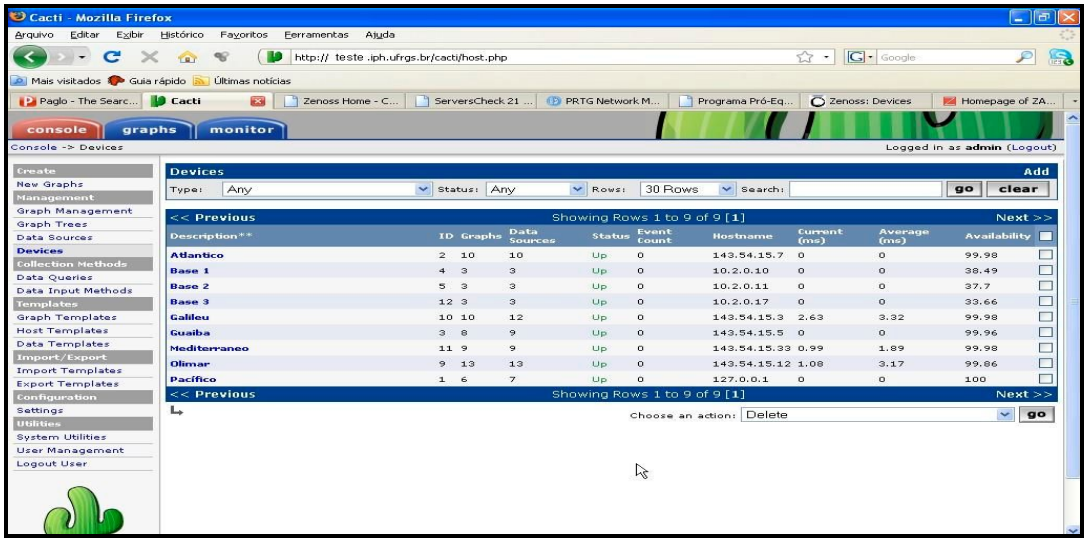
WIFI MANAGER. Software. Disponível em: < http://www.manageengine.com/products/wifi-manager/ >. Acesso em: mar. 2009.

ZABBIX. Software. Disponível em: < http://www.zabbix.com/ >. Acesso em: out. 2008.

ZENOSS. Software. Disponível em: < http://www.zenoss.com/ >. Acesso em: out. 2008.

ANEXO A TELAS DO SOFTWARE CACTI

Cacti é um completo frontend para rrdtool, ele armazena todas as informações necessárias para criar gráficos e preenchê-los com dados em um banco de dados MySQL. O frontend é completamente orientado para PHP. Juntamente com a possibilidade de manter gráficos, fontes de dados armazenados em um banco de dados. Existe também o suporte SNMP para os que são utilizados para criar gráficos de tráfego com MRTG (Multi Router Traffic Grapher).

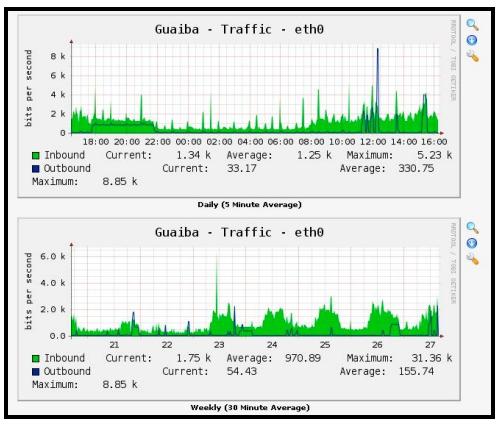


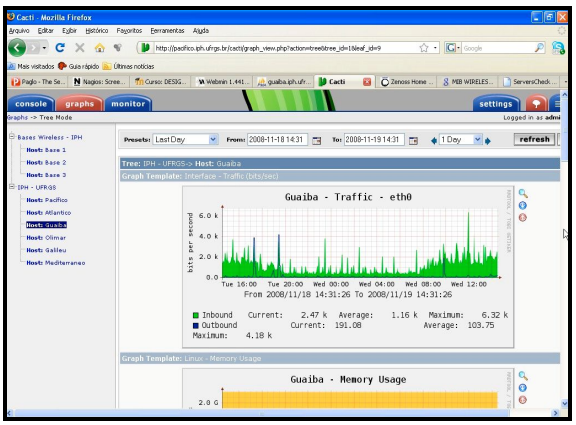
A figura acima mostra a tela principal do Cacti, mostrando a console com os dispositivos cadastrados, seu estatus ou estado, host ou IP e algumas métricas.

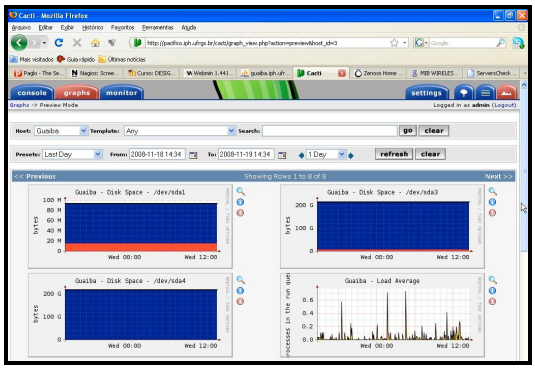
A figura acima mostra o estatus dos dispositivos cadastrados, identificando na cor vermelha os dispositivos que por algum motivo estão fora do ar (out), neste caso o cabo de rede que conecta à rede wireless foi desligado de propósito.

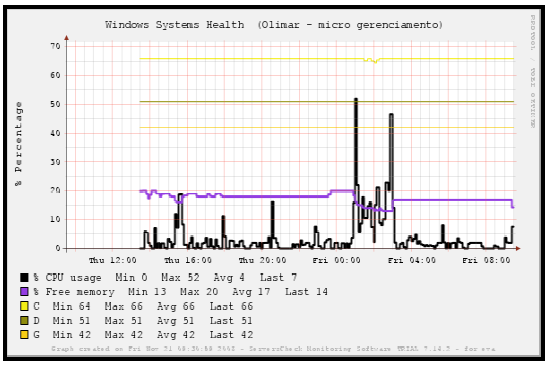
Já a figura seguinte mostra os mesmos dispositivos após o restabelecimento da conexão, voltando as bases wireless a ter conectividade com a estação de monitoramento.

O sistema CACTI permite fazer um inventário dos recursos do sistema monitorado, no caso das figuras abaixo de um servidor Web Linux do IPH, onde mostra os gráficos de número de processos rodando, tráfico nas placas de rede do servidor, memória usada, número de usuários logados, epaço em disco usado, carga de CPU entre outros. Desta maneira pode-se manter um perfil permanente dos recursos da máquina que estão sendo utilizados.



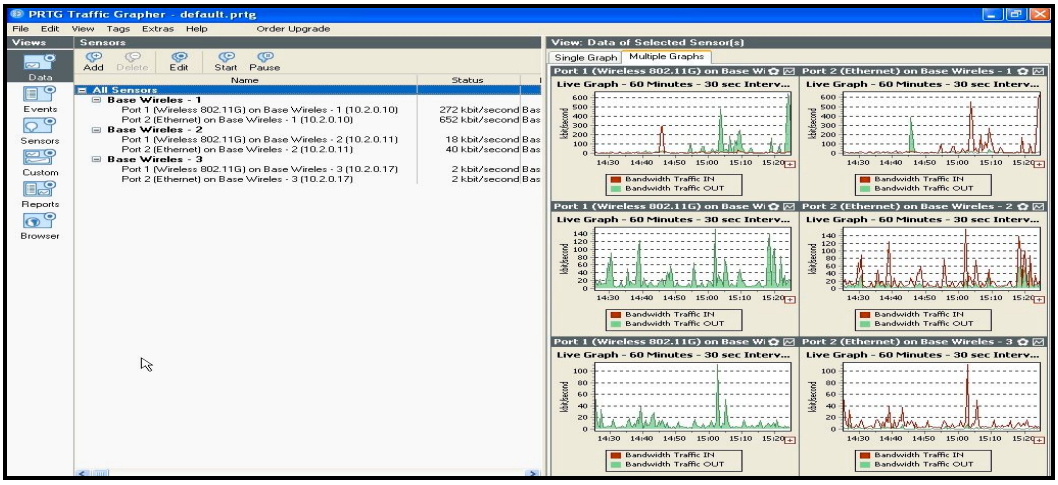



Cacti - Mozilla Firefox
console
graphs
monitor
settings
refresh
clear
Guaiba - Disk Space - /dev/sda1
Guaiba - Disk Space - /dev/sda3
Guaiba - Disk Space - /dev/sda4
Guaiba - Load Average

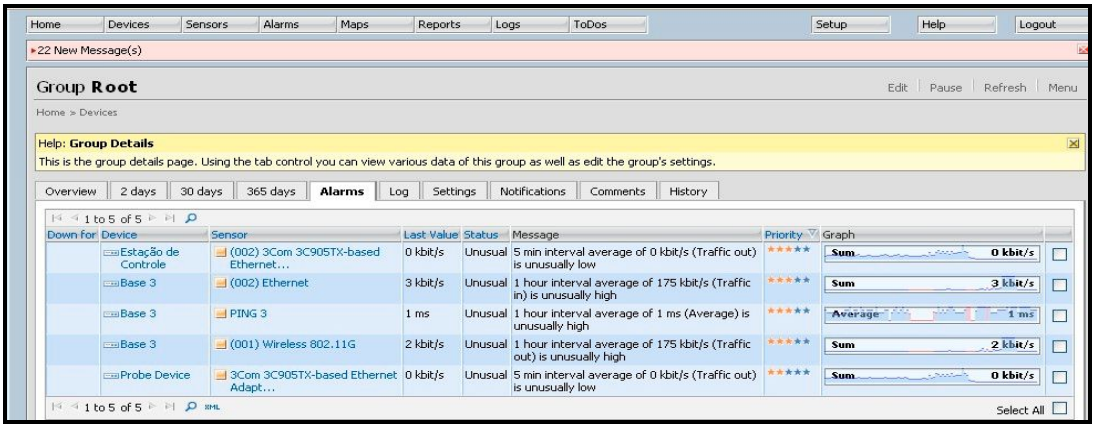

Windows Systems Health (Olimar - micro gerenciamento)
% Percentage
Thu 12:00
Thu 16:00
Thu 20:00
Fri 00:00
Fri 04:00
Fri 08:00
% CPU usage Min 0 Max 52 Avg 4 Last 7
% Free memory Min 13 Max 20 Avg 17 Last 14
C Min 64 Max 66 Avg 66 Last 66
D Min 51 Max 51 Avg 51 Last 51
G Min 42 Max 42 Avg 42 Last 42

ANEXO B TELAS DO SOFTWARE PRTG

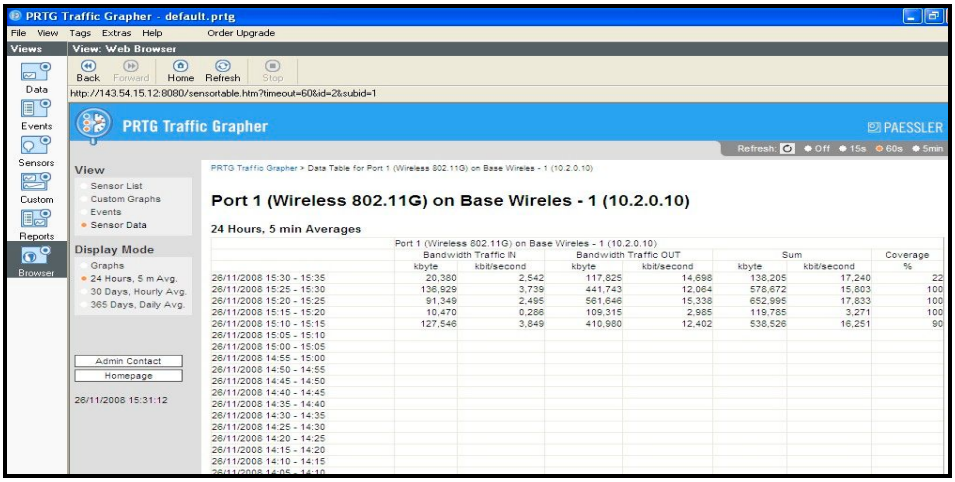
O PRTG Traffic Grapher permite visualizar o comportamento dos dispositivos de rede de maneira gráfica, mostrando o estado dos dispositivos e guardando todas as informações numa base de dados para posteriores estatísticas e cruzamento de informações. A figura abaixo mostra o monitoramento do uso de banda de 3 bases wireless no Instituto de Pesquisas Hidráulicas.



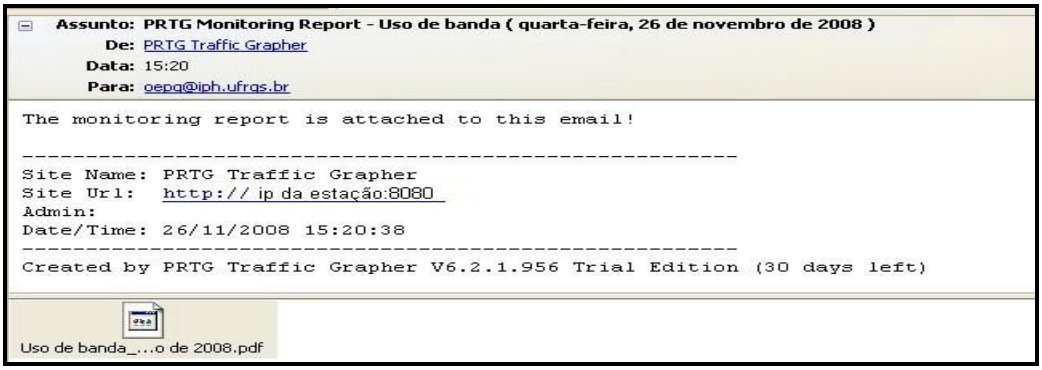
O sistema mantem uma organização de visualização dos dados em: dados atuais, 2 dias, 30 dias e 365 dias, mostrando também os alarmes gerados, logs, configurações, notificações, comentários e histórico de todo o monitoramento.



Além de mostrar os dados graficamente para melhor visualização do comportamento do dispositivo, o sistema permite visualizar em tabelas os dados pontuais em intervalos de 5 em 5 minutos, também mostrando algumas estatísticas de cobertura, totais, etc.



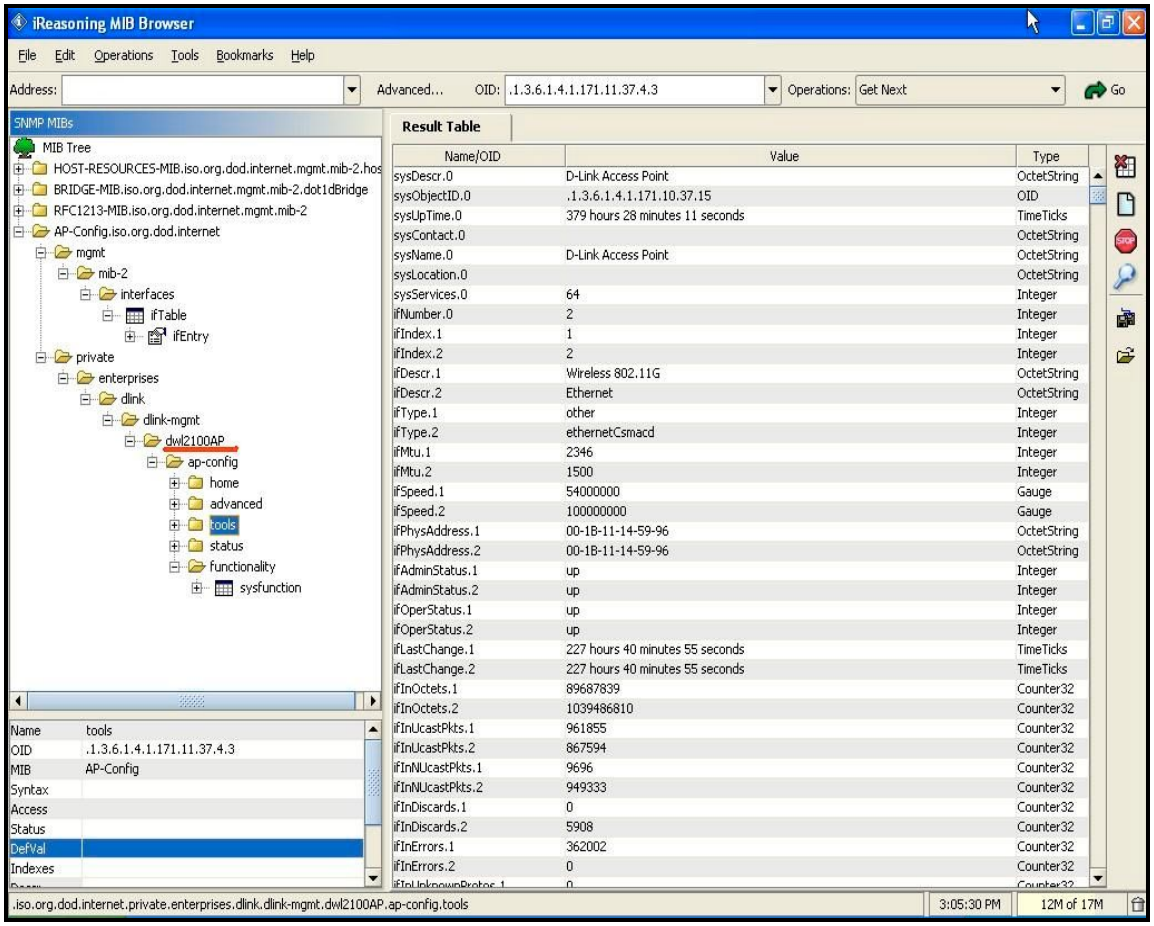
Também emite relatórios previamente configurados, relatando situações e alarmes alertando situações críticas. Na figura abaixo a figura mostra um relatório em pdf emitido via e-mail para o administrador do sistema, além disso, na porta 8080 do servidor de monitoramento podem ser visualizados os relatórios via web.



ANEXO C TELAS DOS MIB BROWSERS

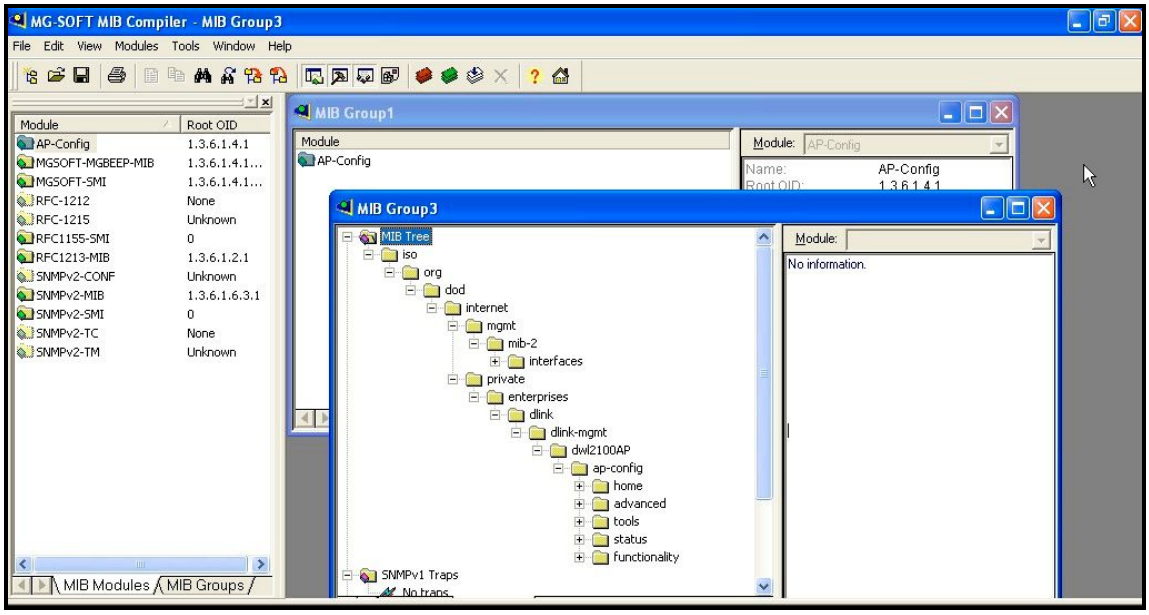
A MIB (Management Information Base) é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para a gerência da rede, e neste caso especifico para gerenciar o equipamento DWL 2100 AP da D-Link. Browsers ou navegadores de MIB permitem ver a hierarquia de variáveis SNMP da MIB, na forma de uma árvore e proporciona informação adicional sobre cada nodo, sendo que cada nodo possui um identificador (OID).

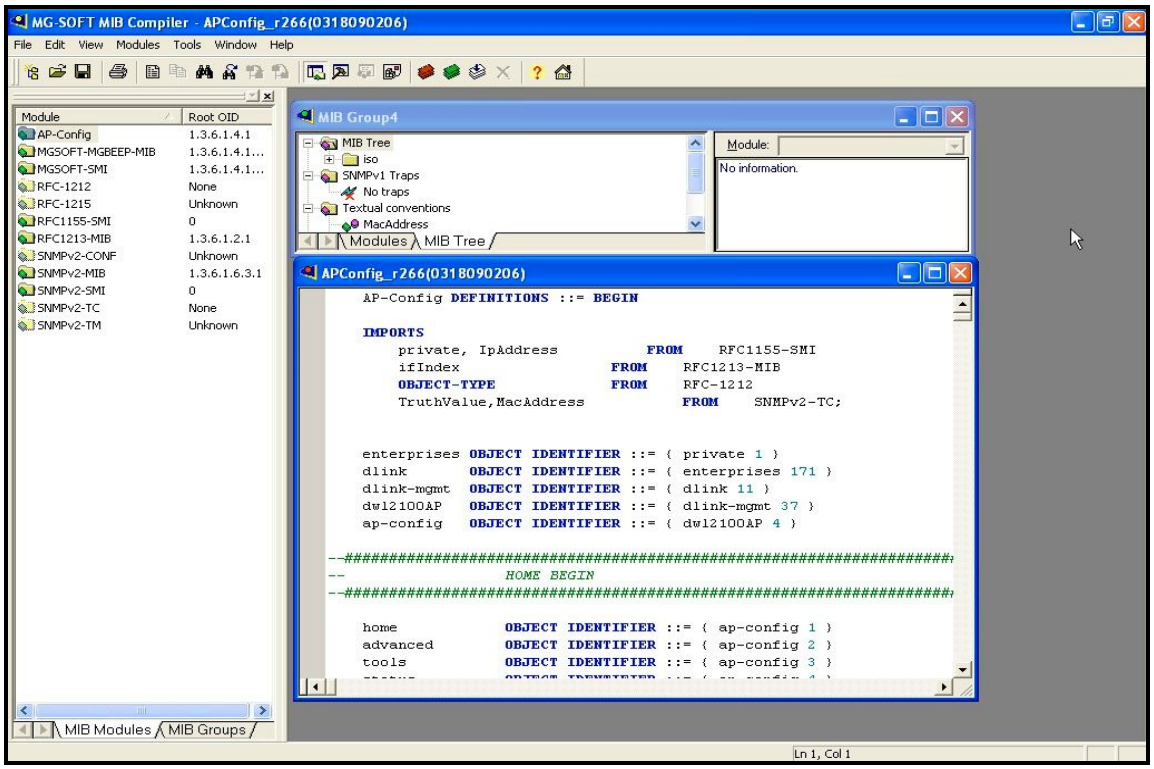
As figuras abaixo neste anexo, apresentam telas de softwares Mib Browsers editando a MIB do AP2100, podendo fazer uma caminhada (walk), fazer consultas para chegar até um determinado parâmetro ou função dentro da MIB.



Por exemplo, após uma série de consultas (Get, walk e Get next) na MIB pelo endereço IP da base, sua máscara de rede, gateway, SSID da base, versão do firmware e endereço MAC, o MIB Browser retornou corretamente os valores descritos abaixo.

stMACAddress.0	00-1B-11-14-59-96	OctetString
stFirmwareVersion.0	v2.20na	OctetString
stMACAddress.0	00-1B-11-14-59-96	OctetString
stGetIPFrom.0	manual	Integer
stIPAddress.0	10.2.0.10	IpAddress
stSubnetMask.0	255.255.248.0	IpAddress
stDefaultGateway.0	10.2.0.1	IpAddress
stSSID.1	ufrgs	OctetString
stChannel.1	11	Integer





ANEXO D MIB DO ACCESS POINT D-LINK AP2100

Browsers ou navegadores de MIB permitem ver a hierarquia de variáveis SNMP da MIB, na forma de uma árvore e proporciona informação adicional sobre cada nodo, sendo que cada nodo possui um identificador (OID), que é composto pelo OID do seu pai mais o seu próprio identificador relativo. Com Browser de MIB pode-se carregar ou compilar facilmente MIBs padrão e MIBs proprietárias, visualizando e manipulando dados que estão disponível em um agente de SNMP.

A MIB (Management Information Base) é forma de estrutura de dados do SNMP. São basicamente tabelas com dados de gerenciamento, cujo formato é especificado pela SMI (Structure of Management Information). Sua descrição é efetuada em linguagem ASN.1 (Abstract Syntax Notation One). Esta MIB do equipamento D-Link AP2100, é uma MIB proprietária, que ao ser compilada por um MIB Browser permite visualizar, consultar e manipular os dados.

```
AP-Config DEFINITIONS ::= BEGIN

IMPORTS
private, IpAddress FROM RFC1155-SMI
ifIndex FROM RFC1213-MIB
OBJECT-TYPE FROM RFC-1212
TruthValue,MacAddress FROM SNMPv2-TC;

enterprises OBJECT IDENTIFIER ::= { private 1 }
dlink OBJECT IDENTIFIER ::= { enterprises 171 }
dlink-mgmt OBJECT IDENTIFIER ::= { dlink 11 }
dwl2100AP OBJECT IDENTIFIER ::= { dlink-mgmt 37 }
ap-config OBJECT IDENTIFIER ::= { dwl2100AP 4 }

--
##########################################################################
-- HOME BEGIN--
##########################################################################

home OBJECT IDENTIFIER ::= { ap-config 1 }
advanced OBJECT IDENTIFIER ::= { ap-config 2 }
tools OBJECT IDENTIFIER ::= { ap-config 3 }
status OBJECT IDENTIFIER ::= { ap-config 4 }
```

```
functionality OBJECT IDENTIFIER ::= { ap-config 99 }

hWirelessBandTable OBJECT-TYPE
SYNTAX SEQUENCE OF HWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Interface Table"
::= { home 1 }

hWirelessBandEntry OBJECT-TYPE
SYNTAX HWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Interface Entry"
INDEX {ifIndex}
::= { hWirelessBandTable 1 }

HWirelessBandEntry ::= SEQUENCE {
hSSID OCTET STRING,
hSSIDBroadcast INTEGER,
hChannel INTEGER,
hChannelList OCTET STRING,
hRadioFrequency OCTET STRING
}

hSSID OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (1..32))
ACCESS read-write
STATUS mandatory
DESCRIPTION "Service Set ID"
::= { hWirelessBandEntry 1 }

hSSIDBroadcast OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "SSID broadcast (0.Disabled, 1.Enabled) "
::= { hWirelessBandEntry 2 }

hChannel OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION "Support Channel"
::= { hWirelessBandEntry 3 }

hChannelList OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
```

```
STATUS mandatory
DESCRIPTION "Channel List"
::= { hWirelessBandEntry 4 }

hRadioFrequency OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Radio Frequency"
::= { hWirelessBandEntry 5 }

hLan OBJECT IDENTIFIER ::= { home 2 }

hGetIPFrom OBJECT-TYPE
SYNTAX INTEGER {static(1), dynamic(2)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Ethernet IP setting (1.Static, 2.Dynamic)"
::= { hLan 1 }

hIPAddress OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Ethernet IP Address"
::= { hLan 2 }

hSubnetMask OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Ethernet Subnet Mask"
::= { hLan 3 }

hDefaultGateway OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Ethernet Default Gateway"
::= { hLan 4 }

adModeTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdModeEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "AP Mode setting Table"
::= { advanced 1 }

adModeEntry OBJECT-TYPE
```

SYNTAX AdModeEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "AP Mode setting Entry"
INDEX {ifIndex}
::= { adModeTable 1 }

AdModeEntry ::= SEQUENCE {
adAPMode INTEGER,
adPtPRemoteAPMACAddress MacAddress,
adAPRRootAPMACAddress MacAddress,
adAPCRootAPMACAddress MacAddress
}

adAPMode OBJECT-TYPE
SYNTAX INTEGER {ap(1), ptpBridge(2), ptmpBridge(3), apRepeater(4), apClient(5)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "AP Mode (1.AP, 2.PtP Bridge, 3.PtPm Bridge, 4.AP Repeater, 5.AP Client)"
::= { adModeEntry 1 }

adPtPRemoteAPMACAddress OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Remote AP MAC"
::= { adModeEntry 2 }

adAPRRootAPMACAddress OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Remote AP MAC"
::= { adModeEntry 3 }

adAPCRootAPMACAddress OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Remote AP MAC"
::= { adModeEntry 4 }

adModePtMPReMacAddTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdModePtMPReMacAddEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "AP Mode setting Table"

```
::= { advanced 2 }

adModePtMPReMacAddEntry OBJECT-TYPE
SYNTAX AdModePtMPReMacAddEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "AP Mode setting Entry"
INDEX {ifIndex, adPtMPReAPMACAddrIndex}
::= { adModePtMPReMacAddTable 1 }

AdModePtMPReMacAddEntry ::= SEQUENCE {
adPtMPReAPMACAddrIndex INTEGER,
adPtMPReAPMACAddress MacAddress
}

adPtMPReAPMACAddrIndex OBJECT-TYPE
SYNTAX INTEGER ( 1..8 )
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Index of MAC"
::= { adModePtMPReMacAddEntry 1 }


adPtMPReAPMACAddress OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Remote AP MAC"
::= { adModePtMPReMacAddEntry 2 }

adPerformanceTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdPerformanceEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "AP Mode setting Table"
::= { advanced 3 }

adPerformanceEntry OBJECT-TYPE
SYNTAX AdPerformanceEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "AP Mode setting Entry"
INDEX {ifIndex}
::= { adPerformanceTable 1 }

AdPerformanceEntry ::= SEQUENCE {
adFrequency OCTET STRING,
adChannel INTEGER,
adChannelList OCTET STRING,
```

```
adDataRate OCTET STRING,
adDataRateList OCTET STRING,
adBeaconInterval INTEGER,
adDTIM INTEGER,
adFragmentLength INTEGER,
adRTSLength INTEGER,
adTransmitPower INTEGER,
adSuperMode INTEGER,
adRadioWave INTEGER,
ad80211gonly INTEGER,
adAutoChannelScan INTEGER
}

adFrequency OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Frequency Usde by Wireless Interface"
::= { adPerformanceEntry 1 }

adChannel OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION "Channel Used by Wireless Interface"
::= { adPerformanceEntry 2 }

adChannelList OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Channel List"
::= { adPerformanceEntry 3 }

adDataRate OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS mandatory
DESCRIPTION "Data Rate of Wireless Interface"
::= { adPerformanceEntry 4 }

adDataRateList OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Channel List"
::= { adPerformanceEntry 5 }

adBeaconInterval OBJECT-TYPE
```

```
SYNTAX INTEGER ( 20..1000 )
ACCESS read-write
STATUS mandatory
DESCRIPTION "Beacon Interval of Wireless Interface"
::= { adPerformanceEntry 6 }

adDTIM OBJECT-TYPE
SYNTAX INTEGER ( 1..255 )
ACCESS read-write
STATUS mandatory
DESCRIPTION "Delivery Traffic Indecation Map"
::= { adPerformanceEntry 7 }

adFragmentLength OBJECT-TYPE
SYNTAX INTEGER ( 256..2346 )
ACCESS read-write
STATUS mandatory
DESCRIPTION "the fragment length of wireless interface"
::= { adPerformanceEntry 8 }

adRTSLength OBJECT-TYPE
SYNTAX INTEGER ( 256..2346 )
ACCESS read-write
STATUS mandatory
DESCRIPTION "RTS Length of Wireless Interface"
::= { adPerformanceEntry 9 }

adTransmitPower OBJECT-TYPE
SYNTAX INTEGER {full(1), half(2), quarter(3), eighth(4), min(5)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Transmit Power"
::= { adPerformanceEntry 10 }

adSuperMode OBJECT-TYPE
SYNTAX INTEGER {super(1), static(2), dynamic(3), disabled(4)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Super Mode: 1.Super without Turbo, 2.Super with Static Turbo,
3.Super with Dynamic Turbo, 4.Disabled "
::= { adPerformanceEntry 11 }

adRadioWave OBJECT-TYPE
SYNTAX INTEGER {off(0), on(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Radio Wave (0.off, 1.on)"
::= { adPerformanceEntry 12 }
```

```
ad80211gonly OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "802.11 g mode only (0.Disabled, 1.Enabled)"
::= { adPerformanceEntry 13 }

adAutoChannelScan OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Auto Channel Scan (0.Disabled, 1.Enabled)"
::= { adPerformanceEntry 14 }

adFilters OBJECT IDENTIFIER ::= { advanced 4 }

adAccSettingWirelessBandTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdAccSettingWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Band Table"
::= { adFilters 1 }

adAccSettingWirelesBandsEntry OBJECT-TYPE
SYNTAX AdAccSettingWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Band Table"
INDEX {ifIndex}
::= { adAccSettingWirelessBandTable 1 }

AdAccSettingWirelessBandEntry ::= SEQUENCE {
adAccSettingWirelessAccessControl INTEGER
}

adAccSettingWirelessAccessControl OBJECT-TYPE
SYNTAX INTEGER { accept(1), reject(2), disabled(3)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Access Contrl (1.Accept, 2.Reject,3.Disabled)"
::= { adAccSettingWirelesBandsEntry 1 }

adAccSettingAccessCtrlMACTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdAccSettingAccessCtrlMACEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Access Control List Table"
::= { adFilters 2 }
```

```
adAccSettingAccessCtrlMACEntry OBJECT-TYPE
SYNTAX AdAccSettingAccessCtrlMACEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Access Control List Entry"
INDEX { ifIndex, adAccSettingAccessCtrlIndex }
::= { adAccSettingAccessCtrlMACTable 1 }

AdAccSettingAccessCtrlMACEntry ::= SEQUENCE {
adAccSettingAccessCtrlIndex INTEGER,
adAccSettingAccessCtrlMACAddr MacAddress
}

adAccSettingAccessCtrlIndex OBJECT-TYPE
SYNTAX INTEGER ( 1..16 )
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Index of MAC"
::= { adAccSettingAccessCtrlMACEntry 1 }

adAccSettingAccessCtrlMACAddr OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION ""
::= { adAccSettingAccessCtrlMACEntry 2 }

adWLANWirelessBandTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdWLANWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Band Table"
::= { adFilters 3 }

adWLANWirelessBandEntry OBJECT-TYPE
SYNTAX AdWLANWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Band Entry"
INDEX {ifIndex}
::= { adWLANWirelessBandTable 1 }

AdWLANWirelessBandEntry ::= SEQUENCE {
adInternalStationConnection INTEGER,
adEthernet2WLANAccess INTEGER
}

adInternalStationConnection OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
```

```
ACCESS read-write
STATUS mandatory
DESCRIPTION "Internal Station Connection (0.Disabled, 1.Enabled)"
::= { adWLANWirelessBandEntry 1 }

adEthernet2WLANAccess OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Ethernet to WLAN Access (0.Disabled, 1.Enabled)"
::= { adWLANWirelessBandEntry 2 }

adEncryption OBJECT IDENTIFIER ::= { advanced 5 }

adSecSettingWirelessBandTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdSecSettingWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Bands Table"
::= { adEncryption 1 }

adSecSettingWirelessBandEntry OBJECT-TYPE
SYNTAX AdSecSettingWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Band Entry"
INDEX {ifIndex}
::= { adSecSettingWirelessBandTable 1 }

AdSecSettingWirelessBandEntry ::= SEQUENCE {
adSecAuthentication INTEGER,
adSecEncryption INTEGER,
adSecValidKey INTEGER,
adSecPassPhrase OCTET STRING,
adSecCipherType INTEGER,
adSecGroupKeyUpdateInterval INTEGER
}

adSecAuthentication OBJECT-TYPE
SYNTAX INTEGER { opensystem(1), sharedkey(2), opensystem-sharedkey(3), wpa-
psk(4), wpa-eap(5) }
ACCESS read-write
STATUS mandatory
DESCRIPTION "Authentication Manner"
::= { adSecSettingWirelessBandEntry 1 }

adSecEncryption OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
```

```
STATUS mandatory
DESCRIPTION "Enctyprion (0.Disabled, 1.Enabled)"
::= { adSecSettingWirelessBandEntry 2 }

adSecValidKey OBJECT-TYPE
SYNTAX INTEGER { first(1), secend(2), third(3), fourth(4) }
ACCESS read-write
STATUS mandatory
DESCRIPTION "Valid Key"
::= { adSecSettingWirelessBandEntry 3 }

adSecPassPhrase OBJECT-TYPE
SYNTAX OCTET STRING ( SIZE(8..63) )
ACCESS read-write
STATUS mandatory
DESCRIPTION "Pass Phrase"
::= { adSecSettingWirelessBandEntry 4 }

adSecCipherType OBJECT-TYPE
SYNTAX INTEGER { auto(1), aes(2), tkip(3)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Cipher Type"
::= { adSecSettingWirelessBandEntry 5 }

adSecGroupKeyUpdateInterval OBJECT-TYPE
SYNTAX INTEGER ( 300..9999999 )
ACCESS read-write
STATUS mandatory
DESCRIPTION "Group Key Update Interval"
::= { adSecSettingWirelessBandEntry 6 }

adSecServerSetting OBJECT IDENTIFIER ::= { adEncryption 2 }

adSecDNSIPAddress OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Domain Name Server IP Address"
::= { adSecServerSetting 1 }

adSecDNS OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..32))
ACCESS read-write
STATUS mandatory
DESCRIPTION "Description of DNServer"
::= { adSecServerSetting 2 }

adSecRADIUSServer OBJECT-TYPE
```

```
SYNTAX OCTET STRING (SIZE (0..32))
ACCESS read-write
STATUS mandatory
DESCRIPTION "RADIUS Server IP Address"
::= { adSecServerSetting 3 }

adSecRADIUSPort OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION "RADIUS Port"
::= { adSecServerSetting 4 }

adSecRADIUSSecret OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..32))
ACCESS read-write
STATUS mandatory
DESCRIPTION "RADIUS Secret"
::= { adSecServerSetting 5 }



adSecSettingKeyTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdSecSettingKeyEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Bands Table"
::= { adEncryption 3 }

adSecSettingKeyEntry OBJECT-TYPE
SYNTAX AdSecSettingKeyEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Band Entry"
INDEX {ifIndex}
::= { adSecSettingKeyTable 1 }

AdSecSettingKeyEntry ::= SEQUENCE {
adSecKey OCTET STRING
}

adSecKey OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS mandatory
DESCRIPTION "Key"
::= { adSecSettingKeyEntry 1 }
```

```
adDHCPServer OBJECT IDENTIFIER ::= { advanced 7 }

adDyFunction OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "DHCP Function Enable/Disable (0.Disabled, 1.Enabled)"
::= { adDHCPServer 1 }

adDynamicTable OBJECT IDENTIFIER ::= { adDHCPServer 2 }

adDyIPStart OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Start IP Address"
::= { adDynamicTable 1 }

adDyRange OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "IP Range"
::= { adDynamicTable 2 }

adDyMask OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Netmask"
::= { adDynamicTable 3 }

adDyGateway OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Gateway"
::= { adDynamicTable 4 }

adDyWins OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Wins"
::= { adDynamicTable 5 }

adDyDns OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
```

```
STATUS mandatory
DESCRIPTION "DNS"
::= { adDynamicTable 6 }

adDyDomain OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(0..64))
ACCESS read-write
STATUS mandatory
DESCRIPTION "Sub Domain"
::= { adDynamicTable 7 }

adDyLease OBJECT-TYPE
SYNTAX INTEGER (60..31536000)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Lease Time In Secs"
::= { adDynamicTable 8 }

adDyStatus OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "DHCP Server Status (0:Disabled, 1:Enabled )"
::= { adDynamicTable 9 }

adStaticTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdStaticEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Static IP Setting"
::= { adDHCPServer 3 }

adStaticEntry OBJECT-TYPE
SYNTAX AdStaticEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Static IP Entry"
INDEX {ifIndex}
::= { adStaticTable 1 }

--Edited by Builde 04/01/09. Delete the node: "adStFunction".
AdStaticEntry ::= SEQUENCE {
adStIP IpAddress,
adStMask IpAddress,
adStGateway IpAddress,
adStDns IpAddress,
adStWins IpAddress,
adStMAC MacAddress,
adStDomain OCTET STRING,
```

```
adStStatus INTEGER
}

adStIP OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Assigned IP Address"
::= { adStaticEntry 1 }

adStMAC OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Client MAC Address"
::= { adStaticEntry 2 }

adStMask OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Netmask"
::= { adStaticEntry 3 }

adStGateway OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Gateway"
::= { adStaticEntry 4 }

adStDns OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "DNS"
::= { adStaticEntry 5 }

adStWins OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "Wins"
::= { adStaticEntry 6 }

adStDomain OBJECT-TYPE
SYNTAX OCTET STRING(SIZE(0..64))
ACCESS read-write
STATUS mandatory
```

```
DESCRIPTION "Sub Domain"
::= { adStaticEntry 7 }

adStStatus OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "DHCP Server Status(1:Enabled, 2:Disabled)"
::= { adStaticEntry 8 }

adCurrentList OBJECT IDENTIFIER ::= { adDHCPServer 4 }

adCurrentDynamicTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdCurrentDynamicEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Current DHCP Dynamic Pools Table"
::= { adCurrentList 1 }

adCurrentDynamicEntry OBJECT-TYPE
SYNTAX AdCurrentDynamicEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Current DHCP Dynamic Pools Entry"
INDEX {ifIndex}
::= { adCurrentDynamicTable 1 }

AdCurrentDynamicEntry ::= SEQUENCE {
adCuDyMAC MacAddress,
adCuDyAssignedIP IpAddress,
adCuDyLease INTEGER
}

adCuDyMAC OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "Binding MAC"
::= { adCurrentDynamicEntry 1 }

adCuDyAssignedIP OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "Assigned IP"
::= { adCurrentDynamicEntry 2 }

adCuDyLease OBJECT-TYPE
SYNTAX INTEGER
```

```
ACCESS read-only
STATUS mandatory
DESCRIPTION "Lease Time"
::= { adCurrentDynamicEntry 3 }

adCurrentStaticTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdCurrentStaticEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Current DHCP Static Pools Table"
::= { adCurrentList 2 }

adCurrentStaticEntry OBJECT-TYPE
SYNTAX AdCurrentStaticEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Current DHCP Static Pools Entry"
INDEX {ifIndex}
::= { adCurrentStaticTable 1 }

AdCurrentStaticEntry ::= SEQUENCE {
adCuStMAC MacAddress,
adCuStAssignedIP IpAddress
}

adCuStMAC OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "Binding MAC"
::= { adCurrentStaticEntry 1 }

adCuStAssignedIP OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "Assigned IP"
::= { adCurrentStaticEntry 2 }

adSiteSurveyTable OBJECT IDENTIFIER ::= { advanced 8 }

adSiteSurvey OBJECT-TYPE
SYNTAX SEQUENCE OF AdSiteSurveyEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Site Survey Table"
::= { adSiteSurveyTable 1 }

adSiteSurveyEntry OBJECT-TYPE
```

```
SYNTAX AdSiteSurveyEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Site Survey Entry"
INDEX {ifIndex}
::= { adSiteSurvey 1 }

AdSiteSurveyEntry ::= SEQUENCE {
adRefresh INTEGER
}

adRefresh OBJECT-TYPE
SYNTAX INTEGER {nothing(0), refresh(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Refresh the Result of Site Survey (0.nothing, 1.refresh)"
::= { adSiteSurveyEntry 1 }

adSiteTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdSiteEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "The Result Table of Site Survey"
::= { adSiteSurveyTable 2 }

adSiteEntry OBJECT-TYPE
SYNTAX AdSiteEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "The Result Entry of Site Survey"
INDEX {ifIndex, adSiteIndex}
::= { adSiteTable 1 }

AdSiteEntry ::= SEQUENCE {
adSiteIndex INTEGER,
adSiteBSSType OCTET STRING,
adSiteChannel INTEGER,
adSiteRSSI INTEGER,
adBSSID MacAddress,
adSiteEncryption OCTET STRING,
adSiteSSID OCTET STRING
}

adSiteIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Index of Site Survey Result"
::= { adSiteEntry 1 }
```

```
adSiteBSSType OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Type of BSS (1:infrastructure 2:Ad-hoc)"
::= { adSiteEntry 2 }

adSiteChannel OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Channel of BSS"
::= { adSiteEntry 3 }

adSiteRSSI OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "RSSI of BSS"
::= { adSiteEntry 4 }

adBSSID OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "BSSID (MAC Address) of BSS"
::= { adSiteEntry 5 }

adSiteEncryption OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Encrypt Status of BSS {WEP off(0), WEP on(1)}"
::= { adSiteEntry 6 }

adSiteSSID OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..32))
ACCESS read-only
STATUS mandatory
DESCRIPTION "SSID of BSS"
::= { adSiteEntry 7 }

tAdmin OBJECT IDENTIFIER ::= { tools 1 }

admUserName OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..12))
ACCESS read-write
STATUS mandatory
```

```
DESCRIPTION "The Administrator Username"
::= { tAdmin 1 }

admPassword OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..12))
ACCESS read-write
STATUS mandatory
DESCRIPTION "The Administrator Password"
::= { tAdmin 2 }

tSystem OBJECT IDENTIFIER ::= { tools 2 }

sysRestart OBJECT-TYPE
SYNTAX INTEGER {nothing(0), reboot(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Reboot AP(0:Nothing{Default Value}, 1.Reboot)"
::={ tSystem 1 }

sysFactoryDefault OBJECT-TYPE
SYNTAX INTEGER {nothing(0), reset(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Reset AP to factory setting(0:Nothing{Default Value}, 1:Reset)"
::={ tSystem 2 }

tUpdate OBJECT IDENTIFIER ::= { tools 3 }

uRemoteSrvIP OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION "TFTP Remote Server IP"
::={tUpdate 1}

uRemoteFileName OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS mandatory
DESCRIPTION "TFTP Remote File Name"
::={tUpdate 2}

uLocalFileName OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
STATUS mandatory
DESCRIPTION "TFTP Remote File Name"
::={tUpdate 3}
```

```
uTFTPCommand OBJECT-TYPE
SYNTAX INTEGER {connect(1), get(2), put(3), nothing(4)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "TFTP Command(1:connect, 2:get, 3:put, 4:nothing{Default Value})"
::={tUpdate 4}

uUpgradeSettingCommand OBJECT-TYPE
SYNTAX INTEGER {firmwareupdate(1), configsetting(2), configsave(3), reboot(4),
factoryreset(5), nothing(6)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Upgrade Setting
Command(1:firmwareupdate,2:configsetting,3:configsave,4:reboot,5:factoryreset,
6:nothing{Default Value})
Manager must set timeout longer than 1 (0.5) min if upgradesettingcommand is
firmwareupdate (configsave)"
::={tUpdate 5}

tMisc OBJECT IDENTIFIER ::= { tools 4 }

telnet OBJECT-TYPE
SYNTAX INTEGER {disabled(0), enabled(1)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Telnet Support Status(0:Disabled, 1:Enabled)"
::={ tMisc 1 }

timeout OBJECT-TYPE
SYNTAX INTEGER {never(0), s60(1), s180(2), s300(3), s600(4), s900(5)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Telnet Timeout Value(0:Never, 1:60sec, 2:180sec, 3:300sec,
4:600sec, 5:900sec)"
::={ tMisc 2 }

stDeviceInfo OBJECT IDENTIFIER ::= { status 1 }

stEthernet OBJECT IDENTIFIER ::= { stDeviceInfo 1 }

stFirmwareVersion OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "AP Firmware Version"
::= { stEthernet 1 }

stMACAddress OBJECT-TYPE
SYNTAX MacAddress
```

```
ACCESS read-only
STATUS mandatory
DESCRIPTION "System Mac Address"
::= { stEthernet 2 }

stGetIPFrom OBJECT-TYPE
SYNTAX INTEGER {manual(1), dynamic(2)}
ACCESS read-only
STATUS mandatory
DESCRIPTION "the IP setting manner of ethernet interface"
::= { stEthernet 3 }

stIPAddress OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "IP address of the ethernet interface"
::= { stEthernet 4 }

stSubnetMask OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "subnet mask of the ethernet interface"
::= { stEthernet 5 }

stDefaultGateway OBJECT-TYPE
SYNTAX IpAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "default gateway of the ethernet interface"
::= { stEthernet 6 }

stWirelessBandTable OBJECT-TYPE
SYNTAX SEQUENCE OF StWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Interface Table"
::= { status 2 }

stWirelessBandEntry OBJECT-TYPE
SYNTAX StWirelessBandEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Interface Entry"
INDEX {ifIndex}
::= { stWirelessBandTable 1 }

StWirelessBandEntry ::= SEQUENCE {
```

```
stSSID OCTET STRING,
stChannel INTEGER,
stSuperMode INTEGER,
stDataRate OCTET STRING,
stSecLevelAuth OCTET STRING,
stSecLevelPriv OCTET STRING
}

stSSID OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Service Set ID"
::= { stWirelessBandEntry 1 }

stChannel OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "support channel"
::= { stWirelessBandEntry 2 }

stSuperMode OBJECT-TYPE
SYNTAX INTEGER { turbo(1), static(2), dynamic(3), disabled(4)}
ACCESS read-only
STATUS mandatory
DESCRIPTION "the super mode setting (1.super with turbo, 2.super with static turbo,
3.super with dynamic turbo, 4.Disabled)"
::= { stWirelessBandEntry 3 }

stDataRate OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "the data rate of wireless interface"
::= { stWirelessBandEntry 4 }

stSecLevelAuth OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Security Level -- Authentication"
::= { stWirelessBandEntry 5 }

stSecLevelPriv OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION "Security Level -- Privacy (1.Eabled, 2.Disabled)"
```

```
::= { stWirelessBandEntry 6 }

stStats OBJECT IDENTIFIER ::= { status 3 }


stWirelessBandTrafficTable OBJECT-TYPE
SYNTAX SEQUENCE OF StWirelessBandTrafficEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Interface Table"
::= { stStats 1 }

stWirelessBandTrafficEntry OBJECT-TYPE
SYNTAX StWirelessBandTrafficEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Interface Entry"
INDEX {ifIndex}
::= { stWirelessBandTrafficTable 1 }

StWirelessBandTrafficEntry ::= SEQUENCE {
stTrSuccessRate INTEGER,
stTrRetryRate INTEGER,
stRxSuccessRate INTEGER,
stRxDuplicateRate INTEGER,
stRTSSuccessCount INTEGER,
stRTSFailureCount INTEGER,
stTrFrameCount INTEGER,
stMuticastTrFrameCount INTEGER,
stTrErrorCount INTEGER,
stTrTotalRetryCount INTEGER,
stTrMutiRetryCount INTEGER,
stRxFrameCount INTEGER,
stMuticastRxFrameCount INTEGER,
stRxFrameFCSErrorCount INTEGER,
stRxFrameDulicateCount INTEGER,
stAckRxFailureCount INTEGER,
stWEPExcludedFrameCount INTEGER,
srWEPICVErrorCount INTEGER
}

stTrSuccessRate OBJECT-TYPE
SYNTAX INTEGER (0..100)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Transmit Success Rate"
::= { stWirelessBandTrafficEntry 1 }

stTrRetryRate OBJECT-TYPE
```

```
SYNTAX INTEGER (0..100)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Transmit Retry Rate"
::= { stWirelessBandTrafficEntry 2 }

stRxSuccessRate OBJECT-TYPE
SYNTAX INTEGER (0..100)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Receive Success Rate"
::= { stWirelessBandTrafficEntry 3 }

stRxDuplicateRate OBJECT-TYPE
SYNTAX INTEGER (0..100)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Receive Duplicate Rate"
::= { stWirelessBandTrafficEntry 4 }

stRTSSuccessCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "RTS Success Count"
::= { stWirelessBandTrafficEntry 5 }

stRTSFailureCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "RTS Failure Count"
::= { stWirelessBandTrafficEntry 6 }

stTrFrameCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Transmitted Frame Count"
::= { stWirelessBandTrafficEntry 7 }

stMuticastTrFrameCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Muticast Transmitted Frame Count"
::= { stWirelessBandTrafficEntry 8 }

stTrErrorCount OBJECT-TYPE
```

```
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Transmitted Errot Count"
::= { stWirelessBandTrafficEntry 9 }

stTrTotalRetryCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Transmitted Total Retry Count"
::= { stWirelessBandTrafficEntry 10 }

stTrMutiRetryCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Transmitted Multiple Retry Count"
::= { stWirelessBandTrafficEntry 11 }

stRxFrameCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Received Frame Count"
::= { stWirelessBandTrafficEntry 12 }

stMuticastRxFrameCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Muticast Received Frame Count"
::= { stWirelessBandTrafficEntry 13 }

stRxFrameFCSErrorCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Received Frame FCS Error Count"
::= { stWirelessBandTrafficEntry 14 }

stRxFrameDulicateCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Received Frame Duplucate Count"
::= { stWirelessBandTrafficEntry 15 }

stAckRxFailureCount OBJECT-TYPE
```

```
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Ack Rcv Failure Count"
::= { stWirelessBandTrafficEntry 16 }

stWEPExcludedFrameCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "WEP Excluded Frame Count"
::= { stWirelessBandTrafficEntry 17 }

srWEPICVErrorCount OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "WEP ICV Error Count"
::= { stWirelessBandTrafficEntry 18 }
stLED OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Panel LED Status"
::={status 4}

adClientInfoTable OBJECT IDENTIFIER ::= { status 5 }

adGetClientTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdGetClientEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Client Information Table"
::= { adClientInfoTable 1 }

adGetClientEntry OBJECT-TYPE
SYNTAX AdGetClientEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Acquiring Clinet Information"
INDEX {ifIndex}
::= { adGetClientTable 1 }

AdGetClientEntry ::= SEQUENCE {
adGetClient INTEGER
}

adGetClient OBJECT-TYPE
SYNTAX INTEGER {nothing(0), acquire(1)}
```

```
ACCESS read-write
STATUS mandatory
DESCRIPTION "To acquire the Client Information (0.nothing, 1.acquire)"
::= { adGetClientEntry 1 }

adClientTable OBJECT-TYPE
SYNTAX SEQUENCE OF AdClientEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "The Result Table of Client Information"
::= { adClientInfoTable 2 }

adClientEntry OBJECT-TYPE
SYNTAX AdClientEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "The Result Entry of Client Information"
INDEX {ifIndex, adSiteIndex}
::= { adClientTable 1 }

AdClientEntry ::= SEQUENCE {
adClientIndex INTEGER,
adClientMAC MacAddress,
adClientBand INTEGER,
adClientAuth INTEGER,
adClientRSSI INTEGER,
adClientPSM INTEGER
}
adClientIndex OBJECT-TYPE
SYNTAX INTEGER
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Index of Client Information Table"
::= { adClientEntry 1 }

adClientMAC OBJECT-TYPE
SYNTAX MacAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION "MAC Address of client"
::= { adClientEntry 2 }

adClientBand OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Band of client (0.802.11A 1.802.11B 2.802.11G)"
::= { adClientEntry 3 }
```

```
adClientAuth OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Authentication status of client (0.disable 1.enable)"
::= { adClientEntry 4 }

adClientRSSI OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "RSSI of client"
::= { adClientEntry 5 }

adClientPSM OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION "Power saving status of client (0.disable 1.enable)"
::= { adClientEntry 6 }

sysfunction OBJECT-TYPE
SYNTAX SEQUENCE OF SysfunctionEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Interface Table"
::= { functionality 1 }

sysfunctionEntry OBJECT-TYPE
SYNTAX SysfunctionEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Wireless Interface Entry"
INDEX {ifIndex}
::= { sysfunction 1 }

SysfunctionEntry ::= SEQUENCE {
apmodes INTEGER,
turbomodes INTEGER,
aclnumber INTEGER,
xrsupported INTEGER,
codebase OCTET STRING,
countrycode OCTET STRING
}
apmodes OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
```

"AP Mode option (bit used: 5):
LSB: AP Client
AP Repeater
PtmP Bridge
PtP Bridge
MSB: Normal AP"
::={ sysfunctionEntry 1 }

turbomodes OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"options of Turbo Mode
1 : 4
0 : 2
"
::={ sysfunctionEntry 2 }

aclnumber OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Max Number of ACL Entries
256 : 256
16 : 16
"
::={ sysfunctionEntry 3 }

xrsupported OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"XR Function Support
1 : Supported
0 : Not Supported
"
::={ sysfunctionEntry 4 }

codebase OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..256))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Code Base"
::={ sysfunctionEntry 5 }

```
countrycode OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..256))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Country code"
::={ sysfunctionEntry 6 }
--
##########################################################################
-- HOME END--
##########################################################################

END
```