Ciência.Inovação 2010 Programa Operacional Ciência e Inovação 2010

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

U.PORTO

FEUP – IRICUP

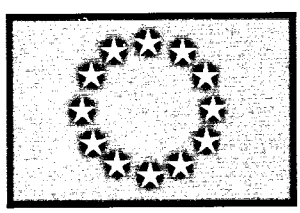
FGRW

Ferramenta de Gestão / Monitorização de Redes *Wireless*

Relatório de Estágio Curricular realizado no IRICUP

Estágio financiado no âmbito do POCI2010.

Porto, 20 de Setembro de 2006



621.3(047.3)/
LEEC
2006/CARDr

131

Ciência.Inovação 2010 Programa Operacional Ciência e Inovação 2010
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

FEUP – Faculdade de Engenharia da Universidade do Porto

IRICUP – Instituto de Recursos e Iniciativas Comuns da Universidade do Porto

LICENCIATURA EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
(Ramo Sistemas de Telecomunicações, Electrónica e Computadores)

FGRW

Ferramenta de Gestão / Monitorização de Redes *Wireless*

Relatório de Estágio Curricular realizado no IRICUP

Estágio financiado no âmbito do POCI2010.

Porto, 20 de Setembro de 2006

Aluno: Rui Miguel Ribeiro Cardoso
Orientador FEUP: Professor Doutor Manuel Alberto Pereira Ricardo
Orientador IRICUP: Engenheiro Mário Paulo Monteiro Serrão

621.3(047.3)/LEEC 2006/GAZD-2

105161

24 02 10

Resumo

Com este trabalho pretendeu-se desenvolver um conjunto de ferramentas que auxiliem a gestão de redes IEEE 802.11 incluindo, em particular, a configuração, a avaliação de desempenho, a análise de falhas e a geração de estatísticas. A ferramenta foi desenvolvida em linguagem Java 1.5, foram utilizadas duas API's: *AdventNet SNMP API* para implementar o protocolo de gestão de redes SNMP e *RRD4J* para implementar a tecnologia *RRDtool* que permite armazenar estatísticas e gerar gráficos a partir destas.

O resultado deste projecto traduziu-se numa ferramenta gráfica que permite visualizar o estado e configuração de uma rede *wireless* em tempo real, possibilita ainda a detecção de falhas e a disponibilização de estatísticas sobre a utilização da rede, nomeadamente tráfego e número de clientes associados ao longo do tempo.

Abstract

The goal of the current work was the development of supporting tools for the management of IEEE 802.11 networks, in particular their configuration, performance evaluation, fault analysis and statistical reporting. These tools were implemented using Java language and two API's: *AdventNet SNMP API* for the implementation of the network management protocol SNMP and *RRD4J* for the implementation of the *RRDtool* technology, that allows for statistical information storing and respective graphical visualization.

The final result of this project is a graphical tool that can be used for real-time visualization of the state and the configuration of a wireless network. This tool can also be used for fault detection and to visualize statistics of the network activity, more specifically, of the traffic and the progress of the number of associated clients.

Agradecimentos

Ciência.Inovação 2010 Programa Operacional Ciência e Inovação 2010
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Ao Programa Operacional Ciência e Inovação 2010 pelo financiamento do estágio.

Ao meu orientador da FEUP, Prof. Doutor Manuel Alberto Pereira Ricardo, pela disponibilidade e orientação prestada, sobretudo na elaboração do relatório.

Ao meu orientador do IRICUP, Eng. Mário Paulo Monteiro Serrão, pela disponibilidade e orientação prestada ao longo do estágio.

A toda a equipa pertencente a unidade de Infra-estruturas Tecnológicas do IRICUP pela disponibilidade e ajuda prestada ao longo do estágio.

Ao Prof. Doutor Ademar Manuel Teixeira de Aguiar pelo esclarecimento das dúvidas relacionadas com a linguagem Java.

ÍNDICE

ÍNDICE DE FIGURAS

Siglas

AP – *Access Point*
API – *Application Programming Interface*
ANSI – *American National Standards Institute*
ANS.1 – *Abstract Syntax Notification One*
BER – *Basic Encoding Rules*
DES – *Data Encryption Standard*
DHCP – *Dynamic Host Configuration Protocol*
EAP – *Extensible Authentication Protocol*
FEUP – Faculdade de Engenharia da Universidade do Porto
FGRW – Ferramenta de Gestão / Monitorização de Redes *Wireless*
IANA – *Internet Assigned Number Authority*
IEEE – *Institute of Electrical and Electronics Engineers*
IETF – *Internet Engineering Task Force*
IOS – *Internetwork Operating System*
IP – *Internet Protocol*
IRICUP - Instituto de Recursos e Iniciativas Comuns da Universidade do Porto
ISO – *International Standards Organization*
LAN – *Local Area Network*
MAC – *Media Access Control*
MD5 – *Message Digest algorithm 5*
MIB – *Management Information Base*
NMS – *Network Management System*
OID – *Object Identifier*
OSI – *Open Systems Interconnection*
PDU – *Protocol Data Unit*
RADIUS - *Remote Authentication Dial In User Service*
RFC – *Request For Comments*
RRD – *Round-Robin Database*
SMI – *Structure of the Management Information*
SMTP – *Simple Mail Transfer Protocol*
SNMP – *Simple Network Management Protocol*
SSID – *Service Set Identifier*
TCP – *Tansport Control Protocol*
USM – *User-based Security Model*
UDP – *User Datagram Protocol*
UP – Universidade do Porto
VACM – *View-based Access Control Model*
VLAN – *Virtual LAN*
Wi-Fi – *Wireless Fidelity*
WLAN – *Wireless LAN*
WEP – *Wired Equivalent Privacy*
XML – *Extensible Markup Language*

1 Introdução

1.1 Enquadramento do trabalho

O programa **e-U** - *universidade electrónica*, diz respeito a uma iniciativa lançada pelo Governo, que envolve Serviços, Conteúdos, Aplicações e Redes de Comunicações Móveis dentro e fora da Universidade para estudantes e professores do Ensino Superior, incentivando e facilitando a produção, acesso e partilha de Conhecimento.

Pretende-se através da implementação de redes sem fios, permitir a transmissão de dados em banda larga, sendo desta forma possível ter acesso a um vasto conjunto de informação e serviços relacionada com o funcionamento das instituições de ensino superior, como por exemplo sumários, artigos, trabalhos, notas, serviços. No conjunto de serviços oferecidos inclui-se também o acesso à *Internet* e a plataformas de conteúdos on-line como por exemplo *b-on*.

A Universidade do Porto - UP, através do Instituto de Recursos e Iniciativas Comuns da UP - IRICUP, participou nesta iniciativa, tendo sido instaladas infra-estruturas de redes sem fios nas suas 14 Faculdades, bem como na Escola de Gestão, Reitoria e em cinco Residências Universitárias.

A gestão das redes implementadas e a manutenção dos diferentes serviços oferecidos localmente são da responsabilidade dos respectivos Serviços de Informática de cada Unidade Orgânica, uma vez que dependem das condições específicas existentes.

Este projecto tem como objectivo o desenvolvimento de uma ferramenta de fácil utilização, sem necessidade de conhecimentos prévios profundos na área de gestão de redes, que auxilie as tarefas de gestão/monitorização da infra-estrutura de rede sem fios nas diferentes Unidades Orgânicas da UP. Na verdade, existem várias ferramentas de gestão no mercado, no entanto uma vez que são bastante dispendiosas, torna-se incomportável a sua aquisição para cada uma das Unidades Orgânicas, além disto, algumas delas exigem um conhecimento profundo de gestão de redes e do protocolo SNMP.

1.2 Objectivos do trabalho

Com este trabalho pretendeu-se desenvolver um conjunto de ferramentas que possibilitem a gestão de redes IEEE 802.11 incluindo, em particular, a configuração, a avaliação de desempenho, a análise de falhas e a geração de estatísticas.

1.3 Estrutura do relatório

Este relatório está estruturado em duas partes, uma parte de descrição de conceitos teóricos necessários para a compreensão do trabalho desenvolvido e uma segunda parte de descrição do trabalho propriamente dito.

A primeira parte está dividida em dois capítulos, um capítulo de redes *wireless* e um capítulo de gestão de redes. No capítulo de redes *wireless* é efectuada uma breve introdução à arquitectura e funcionamento destas redes, seguida da descrição da arquitectura implementada no contexto da UP e do projecto e-U. É também feita uma breve referência aos *Access Points* – AP's Cisco, uma vez que as redes a que se destina a ferramenta desenvolvida só utilizam AP's deste fabricante. O capítulo de gestão de redes é constituído por uma introdução à gestão de redes e ao protocolo utilizado na mesma.

A segunda parte do relatório consiste na descrição do trabalho efectuado, na qual são referidas as diferentes fases de desenvolvimento deste, seguindo-se uma especificação e descrição das funcionalidades implementadas. De seguida é documentada a verificação dos casos de uso e, no final é feita uma análise ao trabalho desenvolvido culminando em algumas sugestões de trabalho futuro.

Existem ainda os seguintes anexos: tecnologias utilizadas, listagem de API's livres em Java que implementam o protocolo SNMP e manual de utilização.

2 Redes *Wireless*

2.1 Introdução

As redes *wireless* também designadas por WLAN's – *Wireless LAN's,* constituem uma alternativa ás LAN's – *Local Área Network* convencionais, fornecendo aos utilizadores as mesmas funcionalidades e serviços mas acrescentando o conceito de mobilidade. Estas redes podem ser utilizadas em áreas residenciais ou de campus. Apresentam como principais características a mobilidade permitida aos seus os utilizadores, o funcionamento em espectro livre de licenciamento, a resistência a erros de transmissão e a simplicidade de gestão e utilização. Estas redes têm algumas desvantagens, das quais se destacam a menor largura de banda disponibilizada quando comparada com os débitos oferecidos nas LAN's convencionais. Além de ser de ordem inferior o débito é variável de acordo com o nível de sinal. A existência de bastantes soluções proprietárias a nível protocolar e os aspectos de segurança são também factores de desvantagem.

As redes *wireless*, podem ser infraestruturadas ou *Ad-Hoc*, neste caso interessa referenciar as primeiras, mais concretamente as redes IEEE802.11 também conhecidas por redes Wi-Fi - *Wireless Fidelity*, das quais é feita uma breve introdução na secção seguinte.

2.2 Redes IEEE802.11

2.2.1 Arquitectura

Uma rede 802.11 consiste numa rede baseada em comunicação via rádio, em que as estações clientes – **STA**'s se ligam ao *Distribution System* – **DS**, através de *Access Points* – AP's. O DS poderá estar ligado ou não a uma LAN – *Local Area Network* convencional. Os **AP**'s são emissores/transmissores, que se comportam como uma *bridge* fazendo o encaminhamento do tráfego entre STA's e DS e vice-versa.

As redes sem fios são constituídas por uma ou mais células, cada célula denomina-se *Basic Service Set* – **BSS** e consiste num conjunto de estações que operam na mesma frequência rádio. A Figura 2.1 mostra um exemplo de uma rede *wireless*.

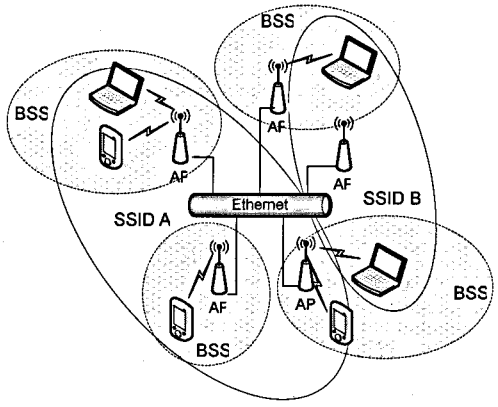


Figura 2.1 – Arquitectura de uma rede IEEE802.11

2.2.2 Normas

Existem três normas que especificam o modo de comunicação rádio nestas redes: 802.11a, 802.11b e 802.11g.

A norma **802.11b** oferece um débito máximo de 11 Mbps, opera segundo o DSSS - *Direct Sequence Spread Spectrum* a 2,4 GHz e selecciona automaticamente o débito, dependendo da qualidade do sinal recebido. Os débitos utilizados por esta norma são 1, 2, 5,5, ou 11 Mbps.

A norma **802.11a** permite débitos até 54 Mbps, opera numa gama de frequências dos 5 GHz e usa uma técnica de codificação denominada *Orthogonal Frequency Division Multiplexing* – OFDM. Para a mesma potência, os sinais a 5GHz têm cerca de metade do alcance dos sinais a 2,4 GHz e que implica uma menor área de cobertura.

A norma **802.11g** usa a codificação OFDM, podendo também usar DSSS para manter a compatibilidade com os rádios 802.11b. Deste modo atingem-se débitos brutos de 54 Mbps na banda dos 2,4 GHz. [3]

2.2.3 Segurança

Em relação à segurança das redes sem fios, é necessário distinguir entre segurança ao nível do acesso à rede através dos *Access Points* e a segurança da rede física. O primeiro é o mais problemático, uma vez que a rede passa a estar acessível a partir de qualquer local com cobertura de rádio, ou seja, os acessos não estão confinados às instalações físicas, ao contrário das redes cabladas. As transmissões podem ser captadas por qualquer receptor dentro da área de acção de um AP, o que motiva preocupações de confidencialidade e integridade.

Ao nível do acesso à rede existem dois componentes principais de segurança: autenticação e encriptação. A **autenticação** permite que apenas utilizadores autorizados

possam aceder à rede. Esta pode ser feita recorrendo à confirmação de valores que identificam o utilizador, como o endereço MAC, ou a valores conhecidos pelo utilizador, como o SSID, WEP ou *passwords*. A **encriptação** visa garantir a confidencialidade dos dados e é o processo no qual se utiliza um algoritmo matemático para cifrar uma mensagem. Os algoritmos de encriptação baseiam-se geralmente num valor, chamado chave, que permite encriptar e desencriptar os dados. [3]

SSID *broadcast*

O SSID funciona com um identificador de um segmento da rede sem fios. Permite segmentar uma rede em várias redes lógicas. É possível desactivar o *broadcast* do SSID de modo a que apenas quem o conhece se possa autenticar. [3]

Sistema de autenticação 802.1X

A norma 802.1X é um protocolo que implementa um método de controlo de acessos por interface física (porta), não se tratando de um protocolo de autenticação propriamente dito, uma vez que é apenas responsável pela translação de mensagens de um determinado algoritmo de autenticação para o formato usual de tramas numa LAN com controlo de acessos. [3]

O princípio básico de funcionamento do 802.1X, assenta na divisão do universo de rede em três entidades distintas: **Suplicante** responsável pelo pedido de acesso ao meio e pelas respostas ao autenticador. **Autenticador** é o dispositivo responsável pelo controlo de acessos ao meio, esta entidade funciona como um intermediário entre o cliente e o servidor de autenticação, pedindo informação de identidade ao cliente e negociando a validação de autenticação com o servidor, como exemplo de autenticador podemos ter um AP. **Servidor de Autenticação** é a entidade que disponibiliza o serviço de autenticação, determinando a partir das credenciais transmitidas pelo suplicante, a autorização ou não de acesso aos serviços disponibilizados numa LAN. O servidor de autenticação é transparente para o suplicante. Um exemplo de servidor de autenticação é um servidor RADIUS - *Remote Authentication Dial In User Service.*

2.3 Arquitectura de autenticação na UP

O esquema de autenticação implementado na UP está representado na Figura 2.2. De seguida é efectuada uma breve explicação da arquitectura de autenticação implementada no contexto do programa e-U.

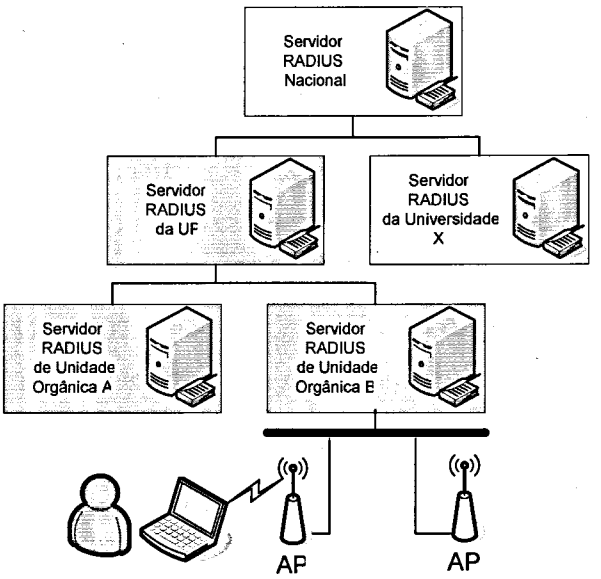


Figura 2.2 – Arquitectura de autenticação implementada na UP

Quando um cliente se associa a um AP, tenta-se autenticar fornecendo as suas credenciais, nomeadamente *username*, *password* e instituição a que pertence. Após receber as credenciais, o AP passa-as ao servidor RADIUS local e fica à espera de uma resposta acerca da validade destas. O servidor RADIUS local verifica se se trata de um utilizador da instituição local e no caso de isso se verificar, confirma ou não o *username* e *password*, no caso de se tratar de um utilizador de outra instituição, o servidor RADIUS local passa as credenciais ao servidor seguinte na cadeia hierárquica, neste caso, o da UP, o qual se encarrega de as enviar o servidor da instituição de origem do utilizador, assim que as credenciais chegam ao servidor RADIUS da instituição de origem do cliente, são verificadas e é transmitida informação relativa à sua validade, esta informação faz o percurso inverso até chegar o servidor RADIUS ligado ao AP onde o cliente se associou. No caso de as credenciais não serem válidas, o servidor RADIUS comunica ao AP que o cliente não se pode autenticar, caso contrário, o servidor comunica ao AP a VLAN à qual o cliente tem acesso. Normalmente está definida uma VLAN para clientes externos e várias VLAN's para clientes internos os

quais são diferenciados como alunos ou funcionários, podendo existir outras diferenciações. Depois de autenticado, o cliente obtém um endereço IP por DHCP - *Dynamic Host Configuration Protocol.*

2.4 AP's Cisco

2.4.1 Principais funcionalidades

As principais funcionalidades dos AP's da Cisco que interessam referenciar no contexto deste projecto são o suporte de SNMPv1, SNMPv2c e SNMPv3 e o facto de poderem serem configurados por *telnet*, existindo alguns modelos como por exemplo os 1200 que têm uma entrada RJ45 permitindo acesso directo ao *access point* para configuração, sem que este tenho de estar ligado à rede.

2.4.2 *Management Information Base* - MIB utilizada pela ferramenta desenvolvida

Nesta secção são utilizados termos específicos de gestão de redes e protocolo SNMP – *Simple Network Management Protocol*, caso o leitor não se encontre familiarizado com estes termos, sugere-se que leia o capítulo seguinte, antes de ler esta secção.

Na figura 2.3 está representada a MIB dos AP's da Cisco utilizada pela ferramenta desenvolvida. Esta MIB é constituída essencialmente por duas sub-árvores com informação de gestão: a sub-árvore *mib-2* que define informação de gestão normalizada e, a sub-árvore *ciscoMgmt* que contém informação de gestão definida pela Cisco.

Como se pode observar pela figura, são utilizados seis grupos descendentes do *Object Identifier* - OID ***mib-2***: *system, interfaces, ip, transmission, entityMIB* e *dot1Bridge*. No grupo ***system*** é possível obter a descrição, o *uptime*, o nome e a localização do AP. O grupo ***interfaces*** contém a tabela de interfaces com dados sobre as diferentes interfaces do AP, nomeadamente interface rádio, interface *ethernet* e respectivas VLAN's. Sobre cada uma das interfaces é disponibilizada a seguinte informação: descrição textual, estado operacional, estado administrativo, tráfego *in* e *out*, endereço MAC - *Media Access Control* entre outros campos. É no grupo ***ip*** que está guarda a máscara de rede, mais concretamente no OID *ipAdEntNetMask*. No OID *dot2StatsDuplexStatus* do grupo ***trasmission*** é possível obter o modo de operação da

interface *ethernet*, nomeadamente se se trata do modo *Half* ou *Full Duplex*. A marca, o modelo e o número de série do AP e da respectiva interface rádio podem ser obtidos na tabela *entPhysicalTable* presente no grupo ***entityMIB***, uma descrição mais pormenorizada do modelo da interface rádio pode ser encontrada na tabela *entLogicalTable* existente no mesmo grupo. Através do grupo ***dot1dBridge*** é possível obter o número de VLAN's configuradas no AP, este número é disponibilizado pelo OID *dot1qNumVlans*.

Na MIB privada da Cisco é possível obter a razão pela qual o AP reiniciou a última vez, no OID *whyReload* e o IP de origem da última tentativa de acesso SNMP falhada, no OID *authAddr*. O grupo ***ciscoDot11MIBObjects*** deste grupo contém informação sobre: SSID's, na tabela *cd11IfAuxSsidTable*; endereço MAC de clientes associados, na tabela *cdIfAssignedAidTable*; norma IEEE802.11, na tabela *cd11IfOperationTable*; canal, na tabela *cd11IfPhyBasicRateSet*; débitos suportados, na tabela *cd11IfSuppDataRatesPrivacyTable*. No grupo ***ciscoDot11AssociationMIB*** é possível obter informação sobre associações com este dispositivo, nomeadamente: número de clientes associados no momento, na tabela *cDot11ActiveDevicesTable*; estatísticas sobre o valor acumulado de associações, autenticações e *roamings* de e, para o AP, na tabela *cDot11AssociationStatsTable*. Na tabela *cDot11ClientConfigInfoTable* é possível encontrar informação sobre a configuração e estado dos clientes associados ao AP e, na tabela *cDot11ClientStatisticTable* é possível encontrar estatísticas acerca destes.

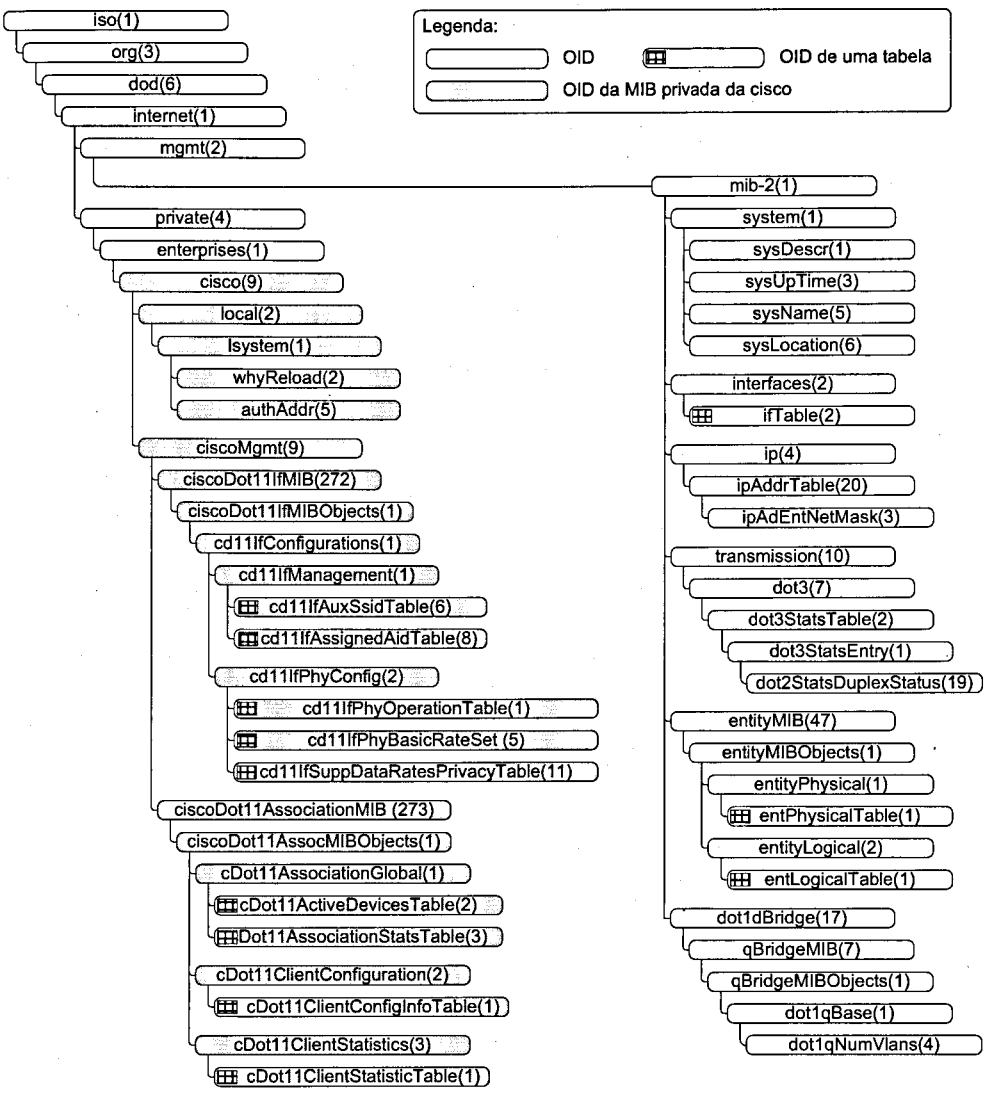


Figura 2.3 - Parte da MIB de um AP Cisco que é utilizada pela aplicação

3 Gestão de Redes

A gestão de uma rede tem como objectivo a monitorização e controlo dos sistemas de *hardware* e *software* que a compõem, garantindo que esta se encontra sempre disponível e, com qualidade de serviço para os seus utilizadores. Desta forma torna-se imprescindível monitorar alguns parâmetros como tráfego, taxas de utilização, taxa de erros entre outros; estes parâmetros deverão ser mantidos dentro de limites preestabelecidos de forma a garantir que a rede não fica congestionada.

De seguida é efectuada uma breve introdução à gestão de Redes OSI.

3.1 Áreas Funcionais

A ISO - *Internacional Organization for Standardization* definiu cinco áreas funcionais na gestão de redes: gestão de falhas, gestão de contabilizações, gestão de configurações, gestão de desempenho e gestão de segurança.

A **gestão de falhas** inclui, desde a detecção e localização de falhas, até isolamento e correcção destas.

A **gestão de contabilizações** diz respeito à contabilização da utilização dos recursos da rede, de forma a promover a sua utilização eficiente. Permite detectar eventuais gastos excessivos por parte de alguns utilizadores que limitem a utilização da rede. Pode também ser utilizada para taxar a utilização de recursos ou serviços.

A **gestão das configurações** diz respeito à configuração e escalonamento dos sistemas de rede, assim como a manutenção e actualização de *software* e *hardware* destes.

A **gestão do desempenho** visa garantir o bom funcionamento do sistema distribuído, evitando o seu congestionamento e permitindo uma utilização eficiente e com qualidade de serviço. Para isso são monitorizados alguns parâmetros como taxa de utilização e volume de tráfego.

A **gestão da segurança** – visa garantir a privacidade e integridade da informação que circula na rede, bem como prevenir e detectar ataques a esta. È implementada a custa de politicas de acesso a nós e recursos da rede, registo de eventos e analise desses registos.

Um sistema de gestão pode conter todas as áreas funcionais referidas, ou apenas algumas.

3.2 Arquitectura

Os elementos da arquitectura de gestão compreendem dispositivos geridos, uma ou mais estações de gestão e, o protocolo de gestão utilizado para troca de informação de gestão entre dispositivos geridos e estações de gestão.

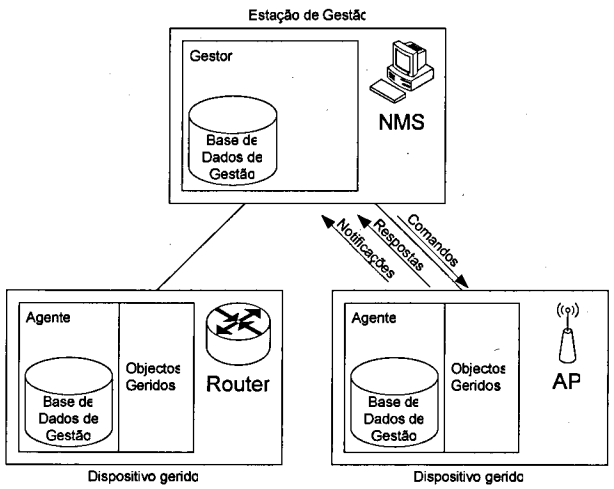


Figura 3.1 – Arquitectura de Gestão

Os **dispositivos geridos** compreendem os diversos dispositivos constituintes de uma rede, como por exemplo: *routers*, *bridges*, *hubs*, *access points*, estações de trabalho, servidores, entre outros. Cada dispositivo gerido contém um módulo de *software* denominado agente e uma base de dados local com informação de gestão. A base de dados contém informação de natureza estatística assim como diversas variáveis representativas do estado e configuração actual do dispositivo. O agente implementa o protocolo de gestão do lado do dispositivo gerido, permitindo trocar informação de gestão com o gestor residente na estação de gestão. Qualquer alteração de alguma das variáveis da base de dados irá ter repercussão no estado do dispositivo e vice-versa.

A **estação de gestão** habitualmente denominada por NMS - *Network Management System*, contém um gestor e uma base de dados de elevada capacidade. Pode existir uma ou mais NMS's numa rede, no caso de existirem várias, poderão estar organizadas hierarquicamente podendo existir dispositivos de rede que desempenhem funções de agente e gestor simultaneamente. O gestor é constituído por uma ou mais aplicações de gestão responsáveis por monitorar e controlar os diversos dispositivos geridos; é também responsável por detectar anomalias na rede, por auscultação directa, ou através da recepção de notificações dos dispositivos geridos. A base de dados guarda

informação de gestão sobre todos os dispositivos geridos pela estação.

O **protocolo de gestão** é responsável pela troca de informação de gestão entre agentes e gestores e é constituído por mensagens básicas como comandos e respectivas respostas e notificações. Os comandos são utilizados pelos gestores para ler ou escrever em variáveis das bases de dados dos agentes; as repostas são utilizadas pelos agentes para responder aos comandos dos gestores; as notificações são utilizadas pelos agentes para reportar eventos aos gestores. Um exemplo de um protocolo de gestão é o SNMP – *Simple Network Management Protocol* que é abordado em seguida.

3.3 SNMP

O protocolo SNMP – *Simple Network Management Protocol* é um protocolo da camada de aplicação situado sobre a camada de transporte UDP – *User Datagram Protocol* e permite a troca de informação de gestão entre dispositivos de rede. Define um conjunto de normas de gestão, que incluem protocolo de comunicação, formato de informação de gestão e estrutura de bases de dados. Actualmente existem três versões de SNMP: SNMPv1 definida pelos *Request for Comments* – RFC's 1155, 1157 e 1212; SNMPv2c definida pelos RFC's 1901 até 1908; SNMPv3 definida nos RFC's 2271 até 2275. A coexistência entre as três versões está detalhada no RFC 3584. As versões 1 e 2c têm muitas funcionalidades em comum, no entanto a versão 2c resulta de uma evolução da primeira, contendo novas funcionalidades tais como novas operações protocolares e novos tipos de dados. A versão 3 contém novas funcionalidades ao nível da segurança e uma nova arquitectura das entidades SNMP, nomeadamente agentes e gestores.

Componentes Básicos

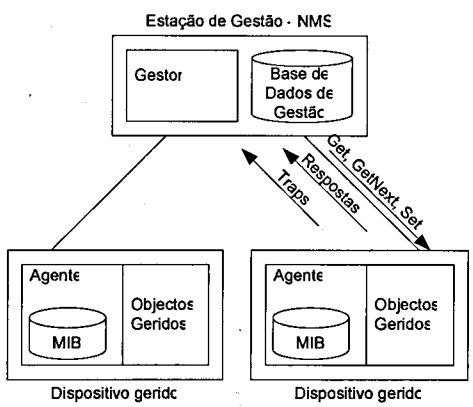


Figura 3.2 – Arquitectura SNMP

A arquitectura SNMP contém três componentes principais: uma ou mais estações de gestão habitualmente designadas por NMS – *Network Management System,* um ou mais dispositivos geridos, e um agente em cada dispositivo gerido.

Um **dispositivo gerido** é um nó da rede que contém um agente e reside na rede gerida. Como exemplos de dispositivos geridos temos: *routers*, *access points*, estações de trabalho e outros elementos de rede.

O **agente**, é um módulo de *software* residente no dispositivo gerido que guarda informação de gestão numa base de dados local cuja estrutura é definida por uma MIB – *Management Information Base*. O agente é responsável por disponibilizar ao NMS a informação guardada na base de dados local de uma forma compatível com SNMP.

O ***NMS – Network Management System*** é responsável pela gestão da rede, contém uma ou mais aplicações de gestão as quais executam operações de monitorização e controlo sobres os dispositivos geridos. O NMS tem uma base de dados local onde guarda informação sobre toda a rede.

Toda a gestão e controlo feitos pelo NMS passam por operações de leitura e escrita nas variáveis guardadas nas bases de dados locais dos dispositivos geridos.

Tipos de Comandos Básicos

Existem três operações atómicas, *Get*, *Set* e *Trap*. O *Get* é um comando de leitura usado pelo NMS nos dispositivos geridos para consulta de variáveis em operações de monitorização. O *Set* é um comando de escrita usado pelo NMS para configurar e controlar os dispositivos geridos. O *Trap* é um comando de notificação usado pelos dispositivos geridos para reportar a ocorrência de eventos ao NMS.

As operações de leitura e escrita são enviadas para a porta UDP 161 do agente e, os *traps* são enviados para a porta UDP 162 do gestor.

3.3.1 MIB – *Management Information Base*

Uma MIB é definida num ficheiro no qual é estabelecido um conjunto de objectos organizados de forma hierárquica; pode ser considerada um mapa que define a estrutura da informação guardada nas bases de dados locais dos dispositivos geridos. Essa informação é constituída por variáveis de tipo simples as quais são instâncias de objectos da MIB. Os objectos consistem em visões abstractas de um recursos reais do sistema. Assim, todos os recursos de um dispositivo são modelados em estruturas de dados representadas como objectos ou conjuntos de objectos. Para se saber o estado de

um qualquer recurso, basta ler o valor da instância correspondente guardada na base de dados, de igual forma, qualquer alteração numa instância irá reflectir-se no estado do recurso real. Diferentes objectos poderão ter diferentes permissões de leitura e de escrita. Um objecto da MIB tem especificidades como o nome, os atributos e o conjunto de operações com as quais pode ser manipulado. O nome do objecto identifica-o univocamente e designa-se por OID – *Object Identifier*. Os atributos de um objecto são: o tipo de dados que este contém e respectiva descrição. As operações de um objecto poderão ser leitura e/ou escrita.

3.3.2 OID's

Os OID's - *Object Identifiers* estão dispostos hierarquicamente em forma de árvore e identificam univocamente valores disponibilizados por um agente SNMP. Existem duas formas de representar um OID, a forma absoluta e a forma relativa. Na forma absoluta o OID pode ser representado numericamente por um ponto seguido de vários valores inteiros separados por pontos como por exemplo: *.1.3.6.1.2.1.1.1.0* ou de uma forma textual como: *.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0* . Na forma relativa um OID é representado pelo nome da MIB seguido de duas vezes dois pontos seguidos da chave textual única definida na própria MIB como se pode ver neste exemplo: *RFC1213-MIB::sysDescr.0* . Sempre que um OID termina em 0, significa que diz respeito a um valor atómico e como tal já não têm descendentes na árvore definida pela MIB, caso contrário o OID representa a família de objectos seus descendentes na estrutura hierárquica.

Hierarquia de objectos

A árvore hierárquica representada na figura 3.3 foi definida pela ISO e representa a estrutura lógica da MIB, mostra o identificador e o nome de cada objecto. Cada nó da árvore é um OID que representa a sub-árvore descendente.

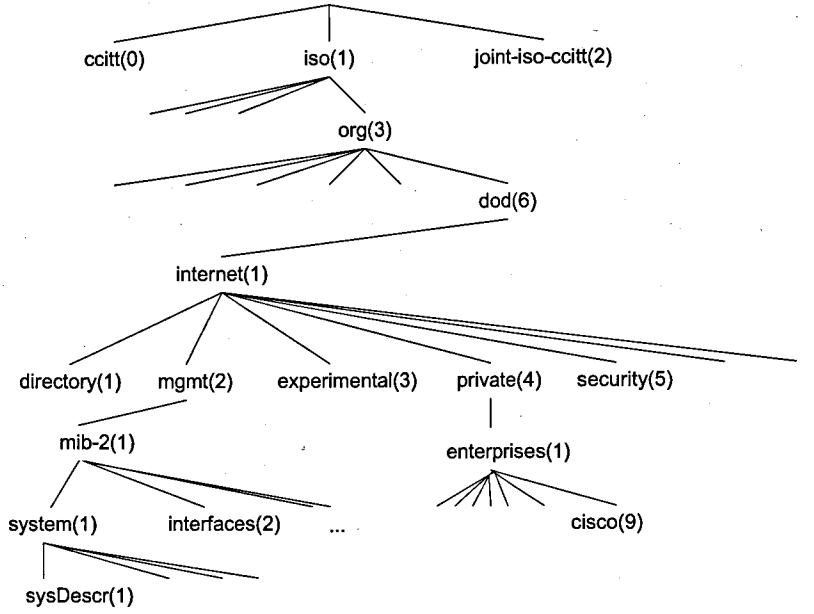


Figura 3.3 - Árvore ISO/CCITT

Relativamente à árvore representada na figura 3.3, importa referir alguns aspectos, o nó raiz não possui identificador O nó **internet(1)** foi definido pelo departamento de defesa dos EUA para a comunidade Internet e, é administrado pela IANA – *Internet Assigned Numbers Authority.* O nó **mgmt(2)** define uma sub-árvore constituída por todas as variáveis de gestão de rede normalizadas. O nó **experimental(3)** específica uma sub-árvore utilizada para experiências e pesquisa e o nó **private(4)** possibilita que organizações privadas possam adicionar os seus próprios nós. Os fabricantes usam identificadores de produtos e definições da MIB necessários na gestão dos seus produtos na sua própria parte da sub-árvore.

MIB's I e II

A MIB I é constituída pela sub-árvore:

.iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1)

Originalmente foi desenvolvida para gestão de comunicações TCP/IP na Internet, concentra-se em informação específica de TCP/IP e inclui informação como: descrição de sistema, interfaces de rede e respectivos parâmetros e estatísticas.

Com o RFC 1213 a MIB I evoluiu para a MIB II, esta fornece informações gerais de gestão sobre dispositivos geridos, permitindo obter informação como o número de pacotes transmitidos numa interface e o endereço da mesma. A MIB II é

constituída pelas seguintes sub-árvores: *system(1), interfaces(2), address translation(3), ip(4), icmp(5), tcp(6), udp(7), egp(8), oim(9), transmission(10), snmp(11).*

Definição de MIB's

Na especificação da estrutura das MIB's, são utilizadas macros que definem a informação de gestão, a custa de tipos de dados definidos na SMI. A SMI por sua vez utiliza um subconjunto da ANS.1, ao qual atribui nomes mais sugestivos de maneira a definir os tipos de dados utilizados pelo SNMP.

SMI - Structure of the Management Information

A estrutura da informação de gestão SNMP é designada por SMI. Esta estrutura organiza a informação de gestão atribuindo-lhe nomes e descrevendo-a de maneira a que seja acessível. A SMI prevê que cada objecto gerido tem de ter um nome, uma sintaxe e uma codificação. O **nome** é o OID que identifica univocamente; a **sintaxe** define o tipo de objecto, por exemplo inteiro ou cadeia de caracteres e a **codificação** descreve a forma como o objecto é serializado de maneira a permitir a sua transmissão. Por razões de simplicidade, o SNMP utiliza um subconjunto do ANS.1.

ANS.1 – Abstract Syntax Notation One

A ANS.1 é uma linguagem formal normalizada definida pela norma *ISO-8824*; é utilizada para definir sintaxes abstractas de tipos de dados. No caso do SNMP, a ANS.1 é usada na especificação da forma dos pacotes de informação e controlo designados por PDU's – *Protocol Data Units*, assim como na definição da informação de gestão básica.

Para definir os seus procedimentos, a ANS.1 utiliza: definições de tipos, valores atribuídos, declaração e utilização de macros e definições de módulos. O **tipo** classifica os dados, os quais poderão ser tipos primitivos como *INTEGER, OCTET STRING, OBJECT IDENTIFIER, NULL, SEQUENCE, SEQUENCE OF* ou tipo definidos aos quais podem ser atribuídos nomes. A **declaração** e **utilização** de macros permite estender a linguagem construindo novos tipos de dados à custa dos tipos primitivos.

```
OBJECT-TYPE::=BEGIN
      TYPE NOTATION::= "SYNTAX" type (ObjectSintax)
                                  " ACCESS" Access
                                  "STATUS" Status
      Access::=     "read-only"
                        | "read-write"
                        | "write-only"
                        | "not-Accessible"
      Status::=     "mandatory"
                        | "optional"
                        | "obsolete"
                        | "deprecated"
      Description::= value (description DisplayString)
      VALUE NOTAION::= value (VALUE ObjectName)
END
```

Estrutura de uma macro

BER – *Basic Encoding Rules*

È um conjunto de regras definidas pela norma ISO 8825-1 utilizadas para a tradução dos valor de variáveis ASN.1 para um formato que permita a sua transmissão e transferência entre os sistemas.

As mensagens SNMP são construídas à custa do tipo SEQUENCE. Cada variável de uma MIB que seja transportada numa mensagem SNMP tem de ser de tipo simples.

3.3.3 SNMPv1

SMI

A SMI da versão 1 do protocolo SNMP define os seguintes tipos de dados: *Counter, TimeTicks, Gauge, IpAddress, NetworkAddress, Opaque, integer, unsigned integer* e *MIB Tables*. *Counter* é um contador inteiro positivo de 32 bits, quando atinge o limite, volta a 0 e que apenas pode ser incrementado. *TimeTicks* é um inteiro positivo que representa o tempo em centésimos de segundo decorrido desde um determinado instante. *Gauge* é um contador inteiro de 32 bits, pode ser incrementado ou decrementado. *IpAddress* representa um endereço IPv4 de 32 bits. *NetworkAddress* representa o endereço IPv4 de 32 bits de uma rede. *Opaque* representa qualquer outros tipos ASN.1 numa *OCTET STRING*. *integer* é um inteiro com sinal, com precisão ASN.1 limitada pela precisão SMI. *unsigned integer* é um inteiro positivo, com precisão ASN.1 limitada pela precisão SMI. *MIB Tables* são estruturas de alto nível, compostas por zero ou mais linhas, que se utilizam para agrupar instâncias tabulares; permitem ler ou alterar uma linha inteira com uma única operação de leitura ou de escrita.

Operações Protocolares

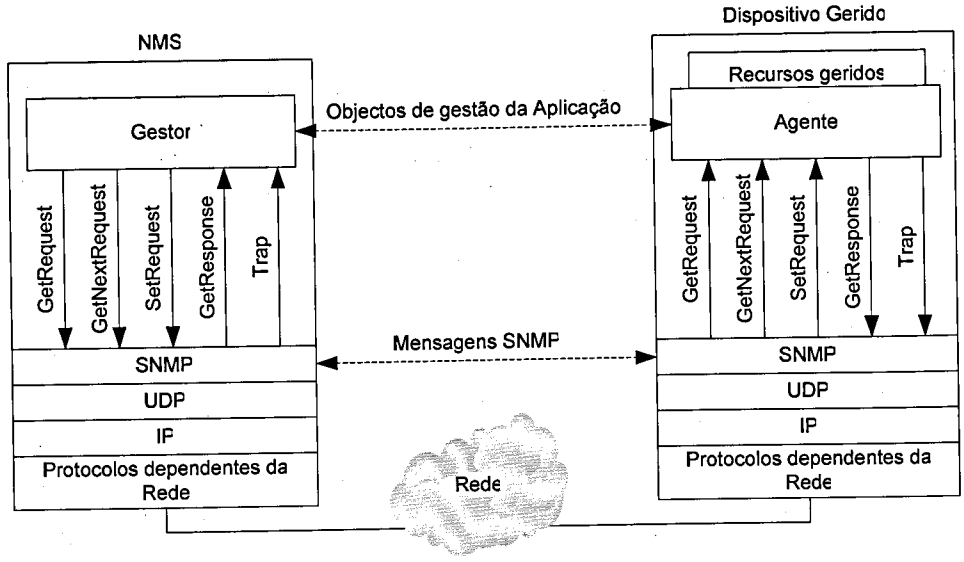


Figura 3.4 – Mensagens do protocolo SNMPv1

O SNMPv1 define as seguintes operações protocolares básicas: *Get, GetNext, Set* e *Trap*. A operação ***Get*** é usada pelo NMS para pedir o valor de uma ou mais instâncias de um agente, é constituída por uma mensagem *GetRequest* enviada pelo NMS ao agente, seguida de uma mensagem de resposta *GetResponse* enviada pelo agente ao NMS. A operação ***GetNext*** é usada pelo NMS para pedir o valor da próxima instância numa tabela ou lista de um agente, ou para descobrir a estrutura de uma MIB, é constituída por uma mensagem *GetNextRequest* enviada pelo NMS ao agente, seguida de uma mensagem de resposta *GetResponse* enviada pelo agente ao NMS. O comando ***Set*** é uma operação atómica utilizada pelo NMS para alterar o valor de uma ou mais instâncias de um agente, permite também criar novas instâncias. O comando ***Trap*** é usado pelo agente para reportar um evento ao NMS.

Existem os seguintes tipos de traps: *coldStart(0), warmStart(1), linkDown(2), linkUp(3), authenticationFailure(4), egpNeighborLoss(5), enterpriseSpecific(6).* O *trap coldStart* reporta uma reinicialização na qual todos os *Counters* e *Gauges* foram reinicializados a zero.). O *trap linkDown* reporta uma alteração para *Down* do estado de uma interface de rede. O *trap linkUp* reporta uma alteração para *Up* do estado de uma interface de rede. O *trap authenticationFailure* reporta uma falha de autenticação na *community string*. O *trap enterpriseSpecific* é usado para reportar eventos definidos pelos fabricantes.

O gestor pode descobrir a estrutura da MIB de um agente à custa da operação *walk*, esta consiste numa série de operações *GetNext* que permitirão percorrer toda a MIB.

As operações protocolares poderão transportar informação acerca de erros, no SNMPv1 estão definidos os seguintes tipos de erros: *noError(0), tooBig(1), noSuchName(2), badValue(3), readOnly(4), genErr(5).*

PDU – Protocol Data Unit

As mensagens SNMPv1 são constituídas por um cabeçalho e um PDU. O PDU contém um conjunto de campos que variam de acordo com o tipo de mensagens. A figura 3.5 mostra o formato dos PDU's SNMPv1.

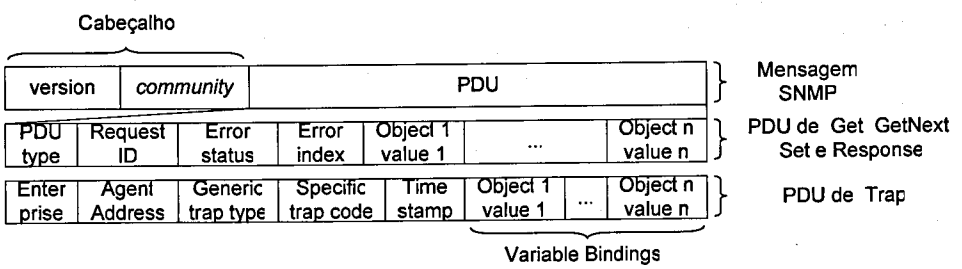


Figura 3.5 - Formato de mensagens SNMPv1

Os PDU's de mensagens do tipo *Get, GetNext, Response* e *Set* contêm os seguintes campos: **PDU type** – especifica o tipo de PDU transmitido; **Request ID** – Associa os pedidos SNMP com as Respostas; **Error status** – Indica o tipo de erro, apenas a operação Set usa este valor diferente de zero; **Error index** – Associa um erro com uma instância particular, este campo apenas é usado pelas respostas; **Variable bindings** – Constituído por pares instância-valor, serve para transporte de dados.

Os PDU's de mensagens do tipo *trap* contêm os seguintes campos: **Enterprise** – especifica o tipo de dispositivo gerido que gerou o trap; **Agent Address** – disponibiliza o endereço do dispositivo gerido que gerou o *trap*; **Generic trap type** – Indica o tipo de *trap* genérico; **Specific trap code** – Indica o tipo de t*rap* específico; **Time stamp** – Especifica o tempo decorrido entre a última reinicialização de rede e a geração do trap; **Variable bindings** – Constituído por pares instância-valor, permite relacionar instâncias particulares com o *trap* gerado.

Community String

Em todas as mensagens SNMP é enviada uma *community string* que implementa um serviço de autenticação e controlo de acesso. Em relação à autenticação, só poderão aceder aos serviços SNMP estações que conheçam a *community*. Relativamente ao controlo de acesso, o protocolo SNMP permite que sejam definidas diferentes *communities* com diferentes permissões de leitura e/ou escrita. Os agentes devem ser configurados para conhecer uma ou mais *communities* e saber o tipo de acesso de cada uma, devem ainda conhecer os destinos dos traps e respectivas *communities.*

Observações

O protocolo SNMPv1 tem como vantagem a simplicidade, no entanto apresenta as algumas limitações: numero limitado de códigos de erro, numero limitado de códigos de notificação, performance limitada, não suporta comunicação entre dois NMS's, e a segurança é reduzida, uma vez que o sistema de autenticação é simples e, os dados, assim como *community* circulam na rede sem serem encriptados.

3.3.4 SNMP v2c

O SNMPv2c surge com uma evolução do SNMPv1, também è baseado em *communities* e as mensagens são parecidas com as mensagens v1. Com a especificação do SNMPv2 foram definidas novas operações e novos tipos de dados, foram também introduzidos melhoramentos nas tabelas de OID's e na alteração dos valores destas. Além disso, passa a ser possível restringir as permissões de uma *community* a uma parte de uma MIB. Também está prevista a comunicação entre NMS, assim pode-se definir uma estrutura hierárquica de NMS e, um elemento da rede pode ser simultaneamente um dispositivo gerido e um NMS.

SMI

A SMI da versão 2 do protocolo SNMP define os seguintes tipos de SMIv1 e os seguintes novos tipos: *Counter32, Counter64, Gauge32, Gauge64, Integer32, Uinteger32, NsapAdress e BIT STRING. Counter32* é idêntico a *Counter. Counter32* é um *Counter* com 64 bits. *Gauge32* é idêntico a *Gauge. Gauge64* é um *Gauge* com 64 bits. *NsapAdress* é um endereço OSI. *BIT STRING* é uma listas de bits com significado próprio.

Operações Protocolares

No protocolo SNMPv2c estão definidas as mesmas operações que no SNMPv1 e, são definidas duas novas operações: *GetBulk* e *Inform. GetBulk* é uma operação de leitura utilizada pelo NMS para obter grandes quantidades de informação de um agente, de um forma eficiente, permitido minimizar o *overhead. Inform* é uma espécie de *trap* com confirmação podendo ser usado por um NMS para reportar eventos a outro NMS, um agente que envie um *inform* fica á espera que o NMS lhe comunique que recebeu o *trap,* se após um *timeout* especificado, o agente não receber a confirmação do *inform,* volta a enviar o *inform.*

No protocolo SNMPv2 são especificados tipos de erro mais específicos: wrongValue, wrongEncoding, wrongType,wrongLength, inconsistentValue, noAcess, notWritable, noCreation, inconsistentName, resourceUnavailable, genErr, CommitFailed, undoFailed.

PDU

Os PDU's SNMPv2 são iguais aos PDU's SNMPv1, com excepção dos *traps* que passam a ter PDU's iguais aos das restantes mensagens e, do PDU de mensagens *GetBulk,* que tem uma estrutura própria, como se pode ver na figura 3.6.

PDU type	Request ID	non-repeaters	max-repetitions	Object 1 value 1	...	Object n value n

Figura 3.6 - PDU GetBulkRequest

Segurança

Tanto o SNMPv1 como o SNMPv2c apresentam algumas falhas de segurança uma vez que não é implementada autenticação nem é utilizada encriptação. Assim sendo, o protocolo é vulnerável a seguintes situações de disfarce, modificação de informação, alteração da sequência de mensagens e divulgação de informação. No caso de disfarce: uma entidade não autorizada pode tentar fazer operações de gestão fazendo-se passar por entidade autorizada. No caso de modificação de informação: uma entidade não autorizada pode tentar alterar mensagens de entidades autorizadas resultando em operações de gestão e configuração não autorizadas. o autorizadas. Em relação à alteração da sequência e tempo das mensagens: uma entidade não autorizada reordena, atrasa, ou copia e repete mais tarde mensagens geradas por entidades autorizadas. No caso da divulgação: uma entidade não autorizada pode extrair informação de objectos

geridos ou, apreender informação monitorizando eventos de notificações entres agentes e gestores.

3.3.5 SNMP v3

A versão 3 do SNMP define novas funcionalidades ao nível da segurança e da arquitectura de agentes e gestores, permitindo a configuração remota de parâmetros SNMP. Esta versão foi concebida de forma a permitir futuras extensões, mantendo o protocolo tão simples quanto possível, permitindo implementações mínimas para redes de pequenas dimensões e ao mesmo tempo suportando funcionalidades mais complexas requeridas na gestão de redes de maiores dimensões; sempre que possível foram reutilizadas especificações existentes nas versões anteriores.

USM - *User-based Security Model*

O modelo USM foi proposto pelo RFC 2274 e define os procedimentos para implementar segurança ao nível das mensagens. Estão definidas protecções contra ataques do tipo disfarce, modificação de informação, alteração de sequência das mensagens e divulgação da informação.

O USM usa MD5- *Message Digest Algorithm* e *Secure Hash Algorithm* para garantir a integridade dos dados, indirectamente possibilita autenticação da origem dos dados e defende contra ataques por disfarce; Usa também DES - *Data Encryption Standard* para protecção contra divulgação.

VACM - *View-based Access Control Model*

O modelo VACM foi proposto pelo RFC 2275 e define elementos de procedimento para controlo do acesso à informação de gestão.

3.3.6 Conclusão

As versões v1 e v2c são mais simples, exigem menos recursos e geram menos *overhead* em relação a versão v3, no entanto possuem como principal desvantagem a falta de segurança. Assim poder-se-á assumir o compromisso de utilizar v2c para monitorização e gestão de performance reservando a versão v3 para configuração, contabilização e gestão de falhas, uma vez que estas operações são mais criticas para a segurança.

4 Descrição do trabalho

4.1 Introdução

A ferramenta de gestão desenvolvida consiste em duas aplicações, uma aplicação gráfica de gestão propriamente dita e uma aplicação de recolha de estatísticas, estas aplicações deverão ser executadas num NMS ligado à rede local de gestão dos *access points* que constituem a rede *wireless*. Na figura 4.1 está representado o diagrama de distribuição desta ferramenta, no qual se observa um directório *FGRW* situado no *file system* da máquina de gestão, este deverá conter as duas aplicações e respectivas bibliotecas.

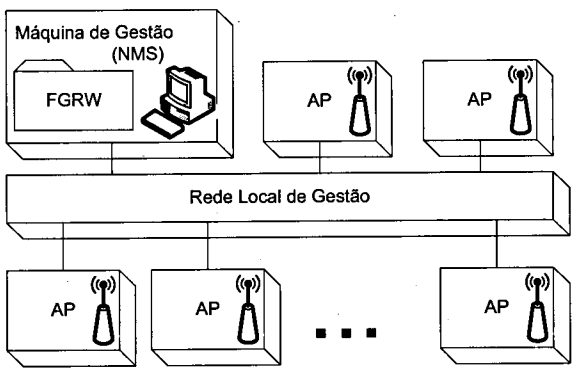


Figura 4.1 – Diagrama de distribuição.

A ferramenta foi desenvolvida na sua totalidade em linguagem Java 1.5, utilizando a ferramenta de desenvolvimento *Eclipse*. Foram utilizadas duas API's – *Application Programming Interfaces*: a *AdventNet SNMP API* e a *RRD4J*, a primeira implementa o protocolo SNMP e, a segunda consiste numa implementação em Java da ferramenta *RRDtool* utilizada para armazenar estatísticas em bases de dados e gerar gráficos a partir destas.

4.2 Fases do projecto.

Este projecto foi dividido em várias fases: fase de estudo, fase de especificação de requisitos, fase de desenvolvimento, fase de testes e por último uma fase de documentação.

A **fase de estudo** contemplou o estudo dos conceitos de gestão de redes e protocolo SNMP, estudo da linguagem Java, pesquisa de API's em Java que

implementam o protocolo SNMP, estudo da MIB dos AP's Cisco e, finalmente pesquisa de API's em Java alternativas à ferramenta *RRDtool*.

Na **fase de especificação de requisitos** foram definidas a funcionalidades a implementar descritas na secção seguinte.

Na **fase de desenvolvimento** foi desenvolvida inicialmente uma aplicação gráfica que permitiu obter informação detalhada sobre um AP da Rede, de seguida, com base nesta aplicação, desenvolveu-se um nova aplicação na qual a rede wireless já está toda representada pelo conjunto dos seus *access points*, esta aplicação já disponibiliza uma tabela de AP's com uma informação sumária sobre os estado e configuração destes, sendo possível observar informação detalhada sobre cada um deles, na parte final foram implementadas as estatísticas e os alarmes.

Na **fase de testes** foi verificado o funcionamento de todos os casos de uso definidos.

Na **fase de documentação** foi elaborado o presente relatório.

4.3 Requisitos gerais

4.3.1 Requisitos Funcionais

Pretende-se que a aplicação desenvolvida possua funcionalidades de monitorização, alarmista, e estatísticas. A monitorização deve permitir obter informações sobre a configuração e estado geral dos *access points*. Sempre que forem detectadas falhas, como por exemplo a alteração do estado de alguma das interfaces de um AP deverá ser despoletado um aviso (envio de um e-mail). A aplicação deve também disponibilizar estatísticas de tráfego referentes á utilização das interfaces físicas (*ethernet* e rádio) e também das diferentes redes virtuais (VLANs) associadas a essas interfaces físicas. Deverão também ser disponibilizadas estatísticas acerca do número de clientes associados a cada *access point*.

4.3.2 Requisitos Não Funcionais

A ferramenta desenvolvida deve ter uma interface gráfica simples e intuitiva; gerar apenas o tráfego estritamente necessário e sempre que possível distribuído ao longo do tempo e, deve também ser robusta, sendo imune à inserção de dados incorrectos no formato ou conteúdo e prevendo a ocorrência de falhas durante a transferência de informação via SNMP.

4.3.3 Modelo de Casos de Uso

Antes de descrever o modelo de casos de uso, importa apresentar o conceito de *caso de uso*. Trata-se de uma funcionalidade de um sistema com valor para o seu utilizador, neste caso o sistema é uma aplicação de software. Assim, o modelo de casos de uso permite visualizar a ferramenta desenvolvida, na perspectiva do utilizador que, neste caso, é o *Gestor de Rede.* Observando a figura 4.2, pode obter-se uma visão geral sobre os pacotes de casos de uso. Os mais importantes são os pacotes de *Monitorização, Estatísticas* e *Alarmes,* são estes grupos de casos de uso que concretizam os objectivos do trabalho. Os pacotes *Configurar Aplicação* e *Editar Lista de AP's* dizem respeito, tal como o próprio nome indica, à configuração da aplicação e lista de *access points* que constituem a rede *wireless.* Por fim referencia-se o pacote *Ferramentas SNMP* que contem algumas funcionalidades adicionais constituídas por operações do protocolo SNMP.

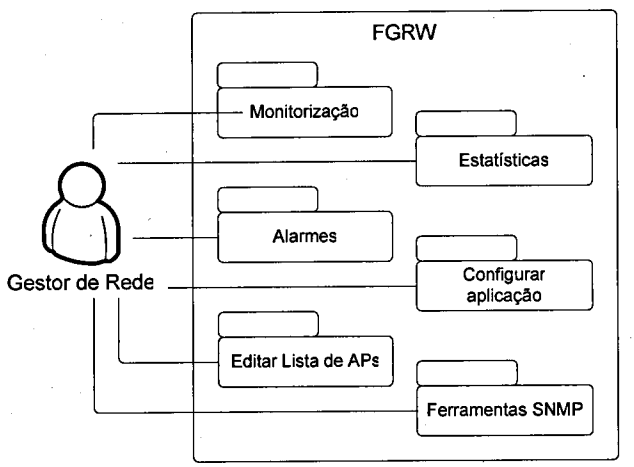


Figura 4.2 - Diagrama de casos de uso, *visão geral.*

Monitorização

Este pacote de casos de uso permite monitorizar o estado e configuração actual de uma rede *wireless.* Neste pacote existem dois casos de usos principais, *Monitorização da Rede* e *Monitorização de detalhes de AP.*

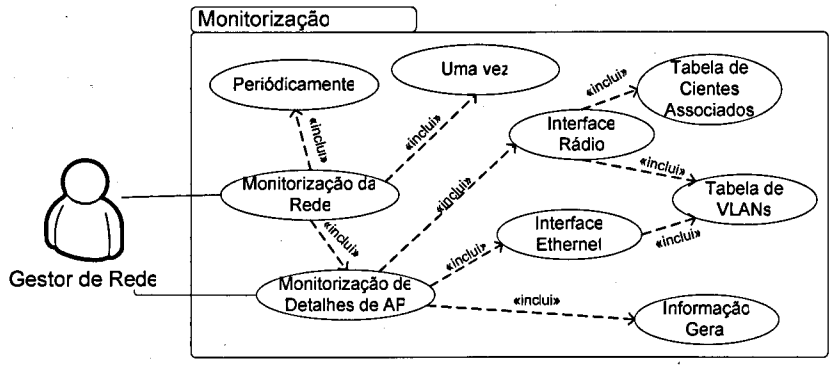


Figura 4.3 - Diagrama de casos de uso, *Monitorização.*

O caso de uso *Monitorização da Rede* permite ter uma visão global da rede, concretizada numa tabela gráfica com um sumário do estado e configuração de cada um dos AP's que constituem a rede *wireless.* A informação incluída no sumário de um *access point* contém os seguintes campos: endereço IP – *Internet Protocol*, endereço MAC da interface *Ethernet*, endereço MAC da interface Rádio, legenda, nome, localização, *upTime*, versão de IOS, estado da interface Rádio, canal rádio, norma IEEE802.11, número de clientes associados. Com excepção do endereço IP, todos os campos atrás referidos são opcionais, isto é, poderão estar ou não visíveis na tabela gráfica. A informação contida no sumário dos AP's é obtida na MIB de cada dispositivo respectivo, com excepção da legenda, que é um campo opcional definido na aplicação e com vista a facilitar uma identificação mais intuitiva de cada um dos *access points.*

Estão previstas diferentes formas de monitorizar a rede, desta forma é possível monitorar a rede uma vez efectuando um rastreio geral aos AP's para visualizar o seu estado e configuração; outra forma de monitorar a rede consiste em rastreios automáticos e periódicos à rede, observando-se a informação obtida destes na tabela gráfica, neste caso estão ainda previstas duas possibilidades: a primeira consiste em pedir informação ao conjunto dos dispositivos a cada rastreio, na segunda possibilidade em cada rastreio efectuado é apenas solicitada informação a um dispositivo alternadamente. A primeira solução tem como vantagem a visualização de informação completamente actualizada e, como desvantagem a geração de um *burst* de tráfego de gestão mais elevado; Na segunda hipótese os *bursts* são menores, no entanto ao visualizarmos a aplicação gráfica, temos que considerar que apenas uma das linhas/dispositivo está actualizada em cada momento.

O caso de uso *Monitorização de Detalhes de AP* permite o acesso a uma informação mais detalhada sobre um *access point* em particular. Essa informação está dividida em três grupos: *Descrição Geral, Interface Ethernet* e *Interface Rádio.* Em *Descrição geral* é disponibilizada informação acerca da configuração do dispositivo. Nos grupos de *Interfaces* é mostrada informação detalhada sobre cada uma das interfaces de rede do AP, incluindo uma tabela com as VLAN's definidas, na *Interface Rádio* é apresentada uma tabela de clientes associados ao dispositivo em questão e à semelhança da tabela de sumários referida anteriormente, esta é também constituída por um conjunto de campos opcionais. Como exemplo de campos da tabela de clientes associados temos: endereço IP, endereço MAC, VLAN, *upTime*, débitos suportados, débito corrente, potência de sinal, qualidade de sinal, estado da associação, tipo de dispositivo, tráfego em octetos e em pacotes, entre outros.

Estatísticas

Este pacote de casos de uso visa a disponibilização de gráficos que permitam o acesso ao histórico do tráfego nas interfaces de um AP e também do número de clientes associados. A gama temporal do histórico poderá ir desde uma hora até um ano, sendo que no segundo caso, a precisão de valores representados será menor que no primeiro.

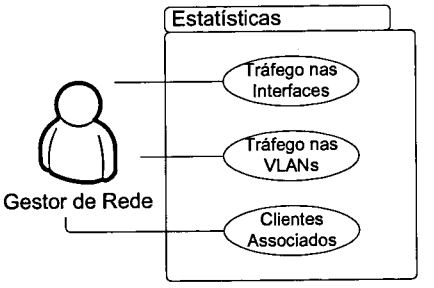


Figura 4.4 - Diagrama de casos de uso, *Estatísticas.*

Existem dois tipos de gráficos, gráficos de **tráfego** e gráficos de **clientes associados.** Os gráficos de tráfego, dizem respeito ao número de octetos que flúem numa determinada interface de rede, esta informação está dividida em duas categorias: tráfego *in* e tráfego *out,* estas duas categorias são representadas no mesmo gráfico, uma acima do eixo das abcissas e a outra abaixo deste. As interfaces de rede sobre as quais é disponibilizada informação são as interfaces *ethernet,* rádio e respectivas VLANS ou sub interfaces. Os gráficos de clientes associados mostram a variação ao longo do tempo

do número de clientes associados a um determinado AP . Em qualquer gráfico é possível obter representação de valores médios e/ou valores máximos.

A gama temporal observada em qualquer gráfico está dividida nas seguintes categorias: hora, dia, semana, mês e ano. Para qualquer uma destas categorias pode-se indicar como referência a data de início ou de fim, com precisão até aos minutos, em alternativa é disponibilizada uma referência automática de fim coincidente com a data e hora actual.

As VLANS monitorizadas nas interfaces rádio e *ethernet* são determinadas no momento em que são inseridos os dados de um AP na aplicação, nesta altura, a aplicação vai processar a tabela de interfaces da MIB do *dispositivo* correspondente de forma a saber quais são a VLAN's que estão definidas neste.

Alarmes

É este pacote de casos de uso que vai permitir a detecção de falhas. Neste caso, as falhas detectadas dizem respeito a passagem do estado de uma interface de rede a *down*. Esta detecção é realizada de duas formas: pela espera de *traps* enviados pelos AP's ou por auscultação directa, na qual se detecte que um dispositivo deixa de responder, o que poderá indicar que a sua interface *ethernet* está *down*. Assim, a detecção de um dos eventos atrás referidos, é sinalizada de acordo com a configuração da aplicação. Essa sinalização é feita enviando um *e-mail* ao gestor de rede, e/ou emitindo um aviso gráfico e/ou fazendo o seu registo num ficheiro de texto.

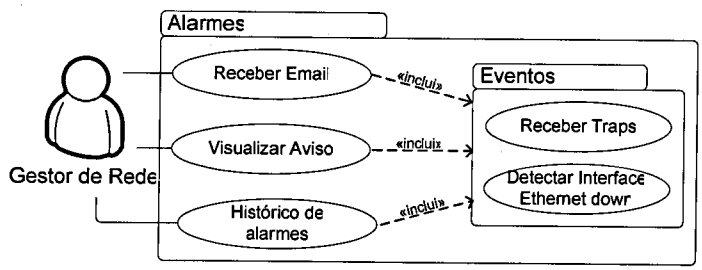


Figura 4.5 - Diagrama de casos de uso, *Alarmes*.

Configurar Aplicação

A ferramenta desenvolvida permite configurar os seguintes parâmetros: conteúdo da tabela de sumários de AP's, conteúdo da tabela de clientes associados a um dispositivo, opções de monitorização, alarmes e parâmetros SNMP.

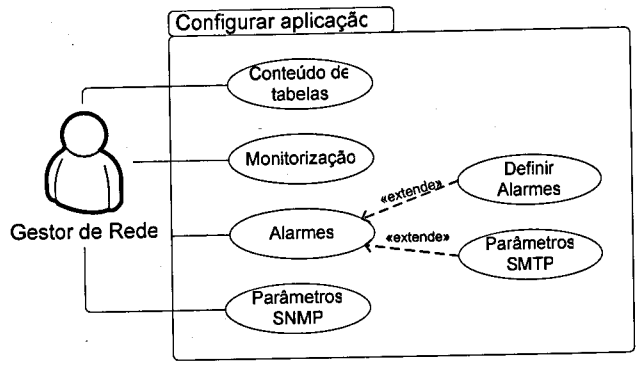


Figura 4.6 - Diagrama de casos de uso, *Configurar aplicação.*

Em relação ao conteúdo das tabelas gráficas, é possível indicar os campos que se pretendem visualizar a partir de um conjunto de campos possíveis.

É também possível configurar um conjunto de parâmetros SNMP que serão utilizados como parâmetros por defeito, na adição e edição de elementos da lista de AP's.

Relativamente a configurações de monitorização, é possível definir o intervalo de tempo entres monitorizações e a forma de se realização destas, a qual pode consistir na monitorização de todos os dispositivos ou monitorização de AP's alternadamente.

Nas configurações de alarmes, é possível definir os tipos de alarmes activos, assim como o tipo de sinalização de cada um deles. É ainda possível configurar os parâmetros SMTP de forma a permitir o envio de *e-mails* caso esta opção seja indicada para sinalização de um ou mais tipos de alarmes.

Editar Lista de AP's

A rede *wireless* é definida por um conjunto de *access points*, estes, são definidos numa lista de AP's contendo os parâmetros do protocolo SNMP e opções de monitorização definidas de cada um deles. As opções de monitorização consistem em informação sobre os campos sobres os quais deverão ser disponibilizadas estatísticas, e ainda um campo a indicar se determinado dispositivo deve ou não ser monitorizado. Esta última opção é útil para quando por qualquer razão, um AP for retirado da rede, não ser necessário remover os seus dados da lista. A informação sobre os campos sobres os quais deverão ser disponibilizadas estatísticas é determinada sempre que se adiciona um dispositivo, quando isso acontece, a aplicação determina quais as VLAN's existentes no AP e guarda os respectivos OID's de tráfego.

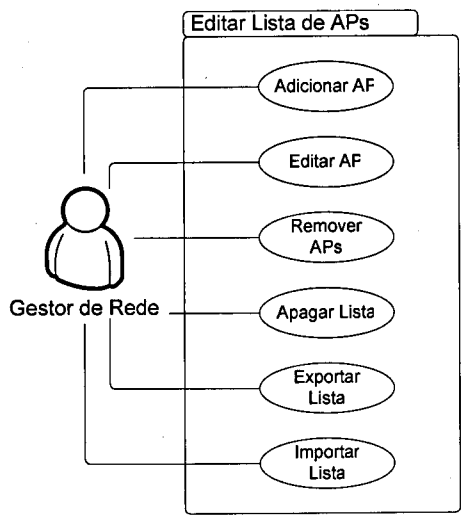


Figura 4.7 - Diagrama de casos de uso, *Editar Lista de AP's*.

Ferramentas SNMP

Estão definidas três ferramentas SNMP que permitem obter o conteúdo de qualquer OID presente na MIB de um dispositivo correspondente a um AP da lista. As operações *Get* e *GetNext* permitem obter o conteúdo de um OID particular, enquanto que a operação *Walk* permite obter o conteúdo de todos os OID da MIB.

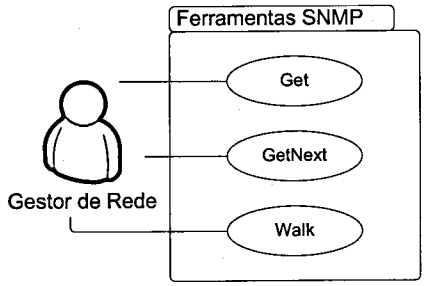


Figura 4.8 - Diagrama de casos de uso, *Ferramentas SNMP*.

4.4 Arquitectura

A ferramenta desenvolvida consiste em duas aplicações, uma aplicação gráfica na qual estão implementados os casos de usos atrás descritos e, uma aplicação sem interface gráfica que tem como finalidade a recolha de dados estatísticos, com os quais vão ser gerados os gráficos mostrados na aplicação gráfica.

4.4.1 Arquitectura Lógica

Aplicação Gráfica

Na figura 4.9 é disponibilizada uma visão geral sobre a arquitectura da aplicação gráfica. São utilizadas três camadas de software, uma camada de comunicação, uma camada de lógica e uma camada de interface gráfica para a interacção com o utilizador.

Na camada de comunicação é implementado o protocolo SNMP, usado para obter informação dos AP's e o protocolo SMTP, utilizado para enviar *e-mails* de alarme.

Na camada lógica ou de aplicação está o núcleo da aplicação, este, é definido pela classe *Controlo* que centraliza toda informação utilizada na execução da aplicação. A classe *Controlo* contém as classes dos pacotes: *configuracao*, *listaDeAps* e *detalhesDeAp*, usa ainda serviços prestados pelas classes do pacote *alarmes*. O pacote *configuracao* contém as classes responsáveis por definir e armazenas todas as configurações da aplicação, o pacote *listaDeAps* é responsável por gerir e manter a lista de AP's, o pacote *detalhesDeAp* define, gere e mantém um conjunto de informação detalhada sobre um dispositivo em particular, o pacote *alarmes* é responsável pela sinalização dos alarmes ocorridos e o pacote *rrd* implementa o arquivo e consulta de estatísticas para disponibilização em gráficos.

A camada de interface com o utilizador é constituída pela classe *JFrame_Fgrw* e respectivos diálogos. Um desses diálogos utiliza o pacote *rrd* da camada lógica para gerar os gráficos estatísticos.

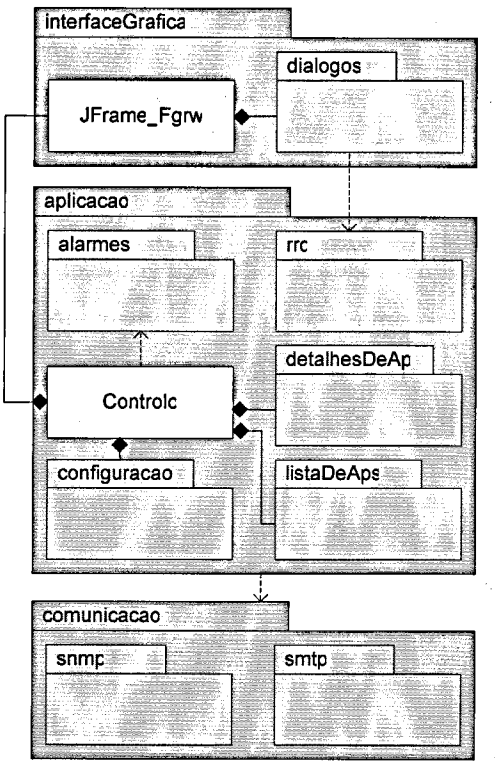


Figura 4.9 – Visão geral do diagrama de classes da aplicação gráfica

Aplicação de recolha de estatísticas

Esta aplicação está dividida em três camadas de *software*: camada de comunicação, camada lógica e camada de interface com o utilizador.

A camada de comunicação é utilizada pela classe *ControloRRD* para, obter via SNMP os valores dos OID's sobre o quais se pretendem efectuar estatísticas.

A camada lógica é constituída pela classe *ControloRRD* que contém uma lista de AP's gerida pelo pacote *listaDeAps*. A classe *ControloRRD* utiliza serviços prestados pelo pacote de classes *rrd*, nomeadamente criar e actualizar os ficheiros que contêm as bases de dados com as estatísticas.

A interface com o utilizador limita-se a mostrar informação sobre a última recolha estatística efectuada e esperar que o utilizador introduza um comando para terminar a aplicação.

A interacção entre a aplicação gráfica e a aplicação de recolha de estatísticas é feita com base no ficheiro *lista.xml*. Na aplicação gráfica é definida a lista de AP's com as respectivas opções de monitorização e quando a aplicação gráfica é fechada, exporta-se automaticamente a sua lista de AP's para o mencionado. Ao iniciar-se aplicação de

recolha de estatísticas é importada a lista de AP's do ficheiro referido e de acordo com as definições presentes no mesmo, inicia-se o processo de recolha estatísticas dos *access points* da lista.

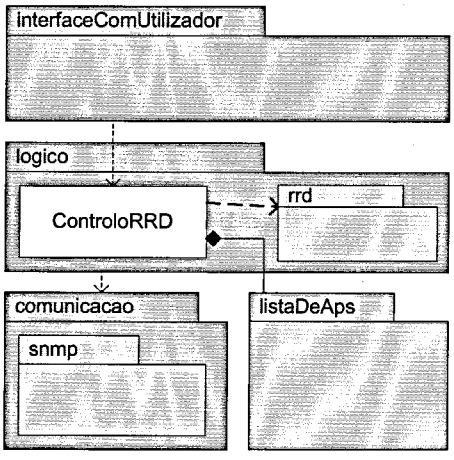


Figura 4.10 - Visão geral do diagrama de classes da aplicação de recolha de estatísticas

Diagrama de Classes

Na Figura 4.11 está representado o diagrama de classes detalhado que define a aplicação gráfica e a aplicação de recolha de estatísticas. Mais uma vez, todo o conjunto de classes está dividido em três camadas, uma camada de interface gráfica, uma camada lógica com a aplicação propriamente dita, um conjunto de ferramentas utilizadas pelas várias camadas e uma camada de comunicação.

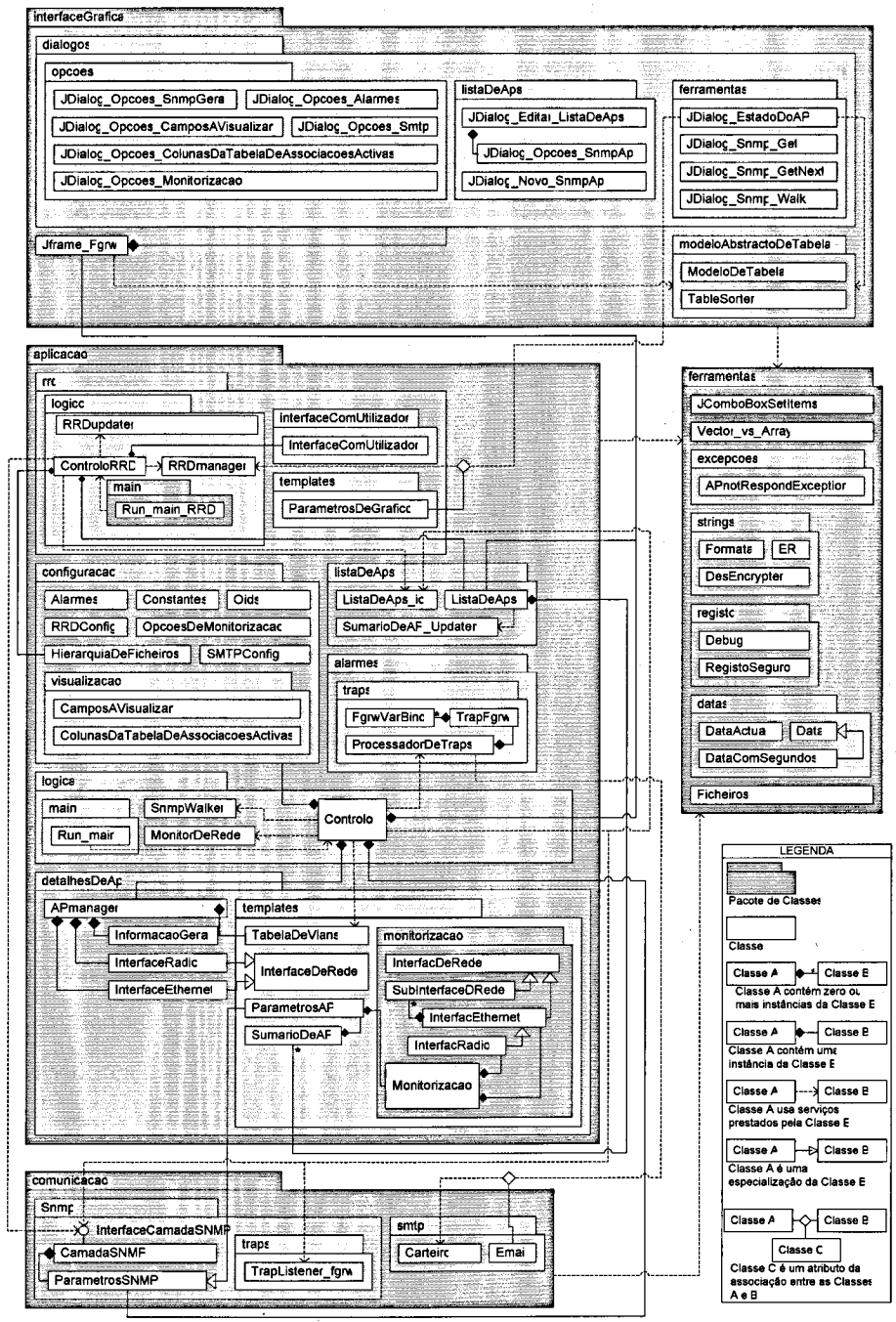


Figura 4.11 – Diagrama de classes – visão detalhada

Descreve-se de seguida os principais pacotes de classes e classes definidos:

O pacote ***interfaceGrafica*** é composto pelas classes gráficas e pelo pacote *modeloAbstractoDeTabela*. A classe ***JFrame_Fgrw*** define a janela principal da aplicação e tem como atributos o conjunto dos diálogos que constituem o pacote *dialogos*. O pacote ***dialogos*** que por sua vez contém três pacotes agrupa o conjunto dos diálogos de acordo com a sua função: o pacote ***opcoes*** é constituído por todos os

diálogos que permitem configurar a aplicação, o pacote ***listaDeAps*** é constituído pelo conjunto dos diálogos que permitem editar a lista de AP's, o pacote ***ferramentas*** é constituído pelo diálogo *JDialog_EstadoDoAP* e pelos diálogos que permitem utilizar as operações básicas do protocolo SNMP. O diálogo ***JDialog_EstadoDoAP*** é o diálogo mais complexo, é este que permite visualizar os detalhes de um *access point*, inclusive os gráficos com as estatísticas, este diálogo utiliza a classe *RRDmanager* para gerar os gráficos. O pacote ***modeloAbstractoDeTabela*** contém a classe ***ModeloDeTabela*** utilizada na definição de conteúdo de tabelas gráficas e a classe ***TableSorter*** que permite que as tabelas gráficas sejam ordenáveis quando se clica no título de uma das suas colunas.

O pacote ***ferramentas*** fornece serviços às três camadas de *software*. A classe ***JComboBoxSetItems*** é utilizada para definir os campos de uma *ComboBox* gráfica, a classe ***Vector_vs_Array*** é utilizada para converter *Vectors* de *Strings* em *arrays* de *Strings,* a classe ***Ficheiros*** fornece serviços de manipulação de ficheiros e directórios tais como verificar se determinado ficheiro existe, apagar ficheiro ou criar directórios. O pacote ***excepcoes*** contém a classe ***APnotRespondException*** utilizada sempre que um AP não responde ou deixa de responder. O pacote *strings* fornece serviços de manipulação de *Strings*: a classe ***Formata*** define métodos de formatação de *Strings* utilizados para formatar alguns valores obtidos nas MIB's dos AP's, a classe ***DesEncrypter*** permite encriptar e desencriptar *Strings*, é utilizada para encriptar a *community* nos ficheiros *XML* que contêm a lista de AP's, a classe ***ER*** define serviços de expressões regulares, permite testar formatos de *Strings* e é bastante utilizada pelos diálogos para filtrar a informação introduzida nos diversos formulários. O pacote ***datas*** fornece serviços de manipulação de datas: a classe ***Data*** permite representar datas com precisão até aos minutos e tem métodos para as incrementar e decrementar, a classe ***DataComSegundos*** é idêntica a *Data* mas, tem precisão até aos segundos, a classe ***DataActual*** permite obter a data actual.

O pacote ***registo*** fornece serviços de *debug* e escrita em ficheiros de texto: a classe ***Debug*** estabelece diferentes níveis de *debug*, e é utilizada para desenvolvimento, através desta classe é possível alterar o grau do detalhe da informação de *debug* de todas as outras classes sem ter de alterar o respectivo código, a classe ***Registo*** fornece um serviço seguro de escrita em ficheiros de texto, permitindo uma abstracção dos detalhes

da manipulação destes, este serviço diz-se seguro e lento porque o ficheiro é aberto e fechado em cada operação de escrita.

O pacote ***aplicacao*** contém toda a lógica das aplicações.

O pacote ***rrd*** contém todas a classes que utilizam directamente a tecnologia RRD, neste pacote, apenas as classes *RRDmanager* e *ParametrosDeGrafico* são usadas pela aplicação gráfica, as restantes classes são usadas apenas pela aplicação de recolha de estatísticas. A classe ***InterfaceComUtilizador*** implementa a interface com o utilizador na aplicação de recolha de estatísticas, a classe ***ParametrosDeGrafico*** utilizada pelo diálogo *JDialog_EstadoDoAP* para especificar os parâmetros dos gráficos a gerar a partir dos ficheiros RRD, a classe ***RRDupdater*** é responsável por criar e actualizar os ficheiros RRD de acordo com as definições presentes nas lista de AP's importada do ficheiro *lista.xml*, a classe ***RRDmanager*** é a única classe que interage com a API *RRD4J*, é esta classe que implementa os métodos que permitem criar e actualizar ficheiros RRD e gerar gráficos a partir destes, a classe ***ControloRRD*** é o núcleo da aplicação de recolha de estatísticas e a classe ***Run_main_RRD*** contém o método *main()* da aplicação de recolha de estatísticas.

O pacote ***configuracao*** contém as classes que definem e guardam as configurações da aplicação gráfica: a classe ***Constantes*** define as principais constantes utilizadas pela aplicação gráfica e a classe ***HierarquiaDeFicheiros*** define a hierarquia de ficheiros utilizada na pasta de trabalho da ferramenta.

O pacote ***listaDeAps*** contém as classes usadas para manipular a lista de AP's: a classe ***ListaDeAps*** contém a lista de AP's propriamente dita, implementa diversos métodos que permitem adicionar, remover e alterar elementos, assim como obter listas de IP's de todos os *access points* ou apenas daqueles que têm monitorização activa, a classe ***ListaDeAps_io*** implementa a exportação e importação do conteúdo da lista de AP's para formato XML, a classe ***SumarioDeAP_Updater*** implemente um *thread* utilizado para a actualizar o sumário de um AP da lista com informação pedida ao dispositivo correspondente.

O pacote ***alarmes*** é constituído pelas classes que implementam a alarmística: a classe ***TrapFgrw*** define uma estrutura de dados utilizada para representar o conteúdo dos *traps* recebidos, a classe ***FgrwVarBind*** define uma estrutura de dados para representar os campos variáveis de um *trap* recebido e a classe ***ProcessadorDeTraps*** é

responsável por processar os traps recebidos, sinalizando-os de acordo com as configurações definidas na classe *Alarmes* do pacote *configuracao*.

O pacote ***logica*** contém o núcleo da aplicação gráfica: a classe ***Run_main*** contém o método *main()* da aplicação gráfica, a classe ***MonitorDeRede*** implementa a monitorização periódica da rede *wireless*, monitorizando os dispositivos definidos na lista de AP's, a classe ***SnmpWalker*** implementa um *walk* SNMP com auxílio da operação SNMP *GetNext* e a classe ***Controlo*** centraliza toda a informação da aplicação gráfica.

O pacote ***detalhesDeAp*** determina as estruturas de dados utilizadas para definir e representar informação de um AP particular. a classe ***APmanager*** é responsável por obter via SNMP a informação detalhada de um *access point* , essa informação é guardada e disponibilizada ao diálogo *JDialog_EstadoDoAP*, a classe ***TabelaDeVlans*** é utilizada pela classe *APmanager* para visualizar as VLANS definidas e respectivo tráfego em octetos; esta classe também é utilizada para obter os OID's das VLAN's sobre os quais são disponibilizadas estatísticas, a classe ***InformacaoGeral*** guarda informação genérica de um dispositivo, a classe ***InterfaceRadio*** gere a informação relativa à interface rádio, a classe ***InterfaceEthernet*** gere a informação relativa a interface *ethernet* e a classe ***InterfaceDeRede*** define o tipo de informação relativa às duas interfaces de rede de um *access point*.

O pacote ***monitorizacao*** define uma estrutura de dados para representar as configurações de estatísticas de um AP, essas configurações incluem os OID's sobre os quais devem ser guardados dados estatísticos e identificações textuais dos mesmos. Na verdade, um nome mais intuitivo para o pacote *monitorizacao* e a classe *Monitorizacao* seria "estatisticas". A classe ***ParametrosAP*** define as configurações de um AP da lista de AP's, nomeadamente: parâmetros SNMP, e opções de monitorização e estatísticas, a classe ***SumarioDeAP*** contém as configurações de um AP e informação sumária do dispositivo correspondente a disponibilizar na tabela de *access points*.

O pacote ***comunicacao*** contem as classes que implementam os protocolos SNMP e SMTP. O pacote ***smtp*** contém as classes que possibilitam o envio de *e-mails*: a classe **Carteiro** é um cliente SMTP que envia *e-mails* definidos pela classe ***Email***.

O pacote ***snmp*** implementa serviços SNMP: a classe ***TrapListener_fgrw*** implementa um *thread* responsável por receber *traps*, a classe ***ParametrosSNMP*** define uma estrutura de dados com os parâmetros do protocolo SNMP, a interface

InterfaceCamadaSNMP define os serviços SNMP usados pela aplicação, a classe ***CamadaSNMP*** implementa os serviços definidos pela interface *InterfaceCamadaSNMP*, basicamente esta classe faz a ponte entre a interface referida e a API SNMP utilizada.

4.4.2 Arquitectura Física

Os componentes da ferramenta desenvolvida estão representados na figura 4.12. Pela figura podemos observar o conteúdo da pasta *FGRW*, onde estão as duas aplicações e respectivos recursos. Descreve-se de seguida os diversos componentes representados: ***jars*** – contém o ficheiro *FGRW.jar* que corresponde à aplicação gráfica e o ficheiro *RRDupdater.jar* que corresponde à aplicação de recolha de estatísticas; ***RRD4J*** – contém os *jars* da API RRD4J; ***AdventNet*** – contém os *jars* da AdventNet SNMP API.; ***MIBs*** – contém as MIB's; ***logs*** – nesta pasta são guardados os registos dos alarmes e da actualização dos RRD's; ***walks*** – nesta pasta são guardados os resultados dos *walks;* ***RRDs*** – contém os ficheiros RRD's com as estatísticas; ***temp*** – é nesta pasta que são guardados os ficheiros temporários gerados pela aplicação, estes dizem respeito às imagens dos gráficos; ***FGRWwindows.jar*** – aplicação que executa um comando Windows para executar o ficheiro *FGRW.jar*; ***FGRWlinux.jar*** – aplicação que executa um comando Linux para executar o ficheiro *FGRW.jar; **lista.xml*** – ficheiro XML onde está definida a lista de AP's.

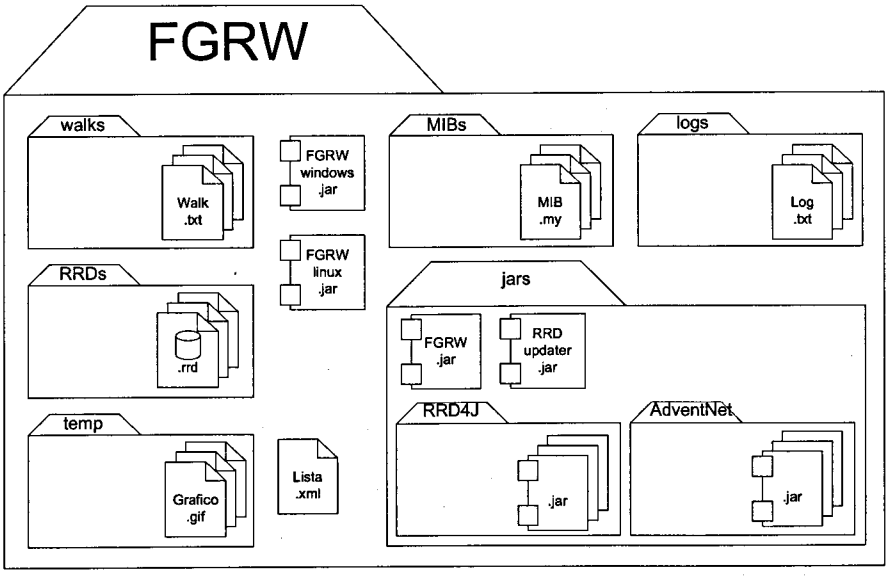


Figura 4.12 – Diagrama de Componentes.

4.5 Principais Decisões de Desenvolvimento

As principais decisões que marcaram a arquitectura e modo de funcionamento das duas aplicações desenvolvidas são descritas de seguida.

4.5.1 Escolha de tecnologias

Java

A linguagem de programação escolhida foi o Java, esta escolha deve-se ao facto de se tratar de uma linguagem de alto nível orientada a objectos e independente do sistema operativo. Esta última característica foi a mais relevante, uma vez que as características anteriores também se aplicam a outras alternativas consideradas como C++ ou C#. O perl e o php também eram possibilidades que facilitariam uma possível implementação de uma interface *web* para a ferramenta.

AdventNet SNMP API

Após uma pesquisa sobre as API's de SNMP livres para Java, constatou-se que a AdventNet era a mais referenciada. Esta API tem uma versão *free* e uma versão *Professional*, a primeira é uma versão limitada da segunda. A versão *free*, apesar de limitada, possui os requisitos considerados necessários para a aplicação desenvolvida, nomeadamente implementa operações *Get*, suporta SNMP v1, v2c e v3, e permite a recepção de *traps*, tem ainda a vantagem de possuir funcionalidades de alto nível como *GetTable* e *GetColumn* que permitem obter tabelas ou partes de tabelas.

XML – Extensible Markup Language

O XML é um formato *standard*, aberto e multiplataforma, utilizado para descrição, armazenamento e troca de dados.

A lista de AP's utilizada pela ferramenta pode ser exportada para um ficheiro em formato XML, podendo posteriormente ser importada a partir deste. A escolha do formato XML deve-se ao facto de se tratar de um formato textual normalizado que, além das características já referidas, pode ser editado com um simples editor de texto e visualizado em qualquer *browser*.

RRD4J

A tecnologia mais indicada para gerir as estatísticas seria a ferramenta RRDtool, no entanto esta, não se encontra disponível para linguagem Java, assim foi necessário encontrar uma alternativa, essa alternativa foi a API *RRD4J* que consiste uma implementação em Java das funcionalidades da ferramenta RRDtool.

4.5.2 Funcionalidades

Usabilidade

De modo a facilitar a utilização da aplicação, tomaram-se alguns cuidados relativamente à sua usabilidade entre os quais: em todos os diálogos de configurações existem **valores predefinidos**; nos diálogos de selecção de várias opções existem **botões para seleccionar todas as opções ou nenhuma**; em todos os diálogos em que é necessário introduzir informação em caixas de texto, é **filtrado o formato e conteúdo da informação introduzida**. No caso de esta informação não ser válida, é emitido um diálogo a identificar a ocorrência e a respectiva justificação, nestes casos só é possível sair do diálogo depois de estarem todos os campos preenchidos com valores válidos, ou então pode-se cancelar o diálogo ficando em vigor as configurações anteriores à abertura do diálogo. Por simplicidade, apenas é possível estar activo **um diálogo de cada vez** e sempre que são efectuadas **alterações na aplicação, a informação disponibilizada é automaticamente actualizada**.

Embora apenas seja necessário definir os parâmetros SNMP para cada um dos AP's, foram definidos parâmetros SNMP gerais que funcionam como parâmetros predefinidos cada vez que se edita ou adiciona um *Access Point* na lista de AP's.

A lista de AP's é guardada automaticamente no ficheiro *lista.xml* ao encerrar a aplicação e sempre que este ficheiro exista, é carregado automaticamente ao executar a aplicação.

Alarmes

Os alarmes mais importantes definidos para a aplicação dizem respeito à detecção de mudança de estado de alguma das interfaces de rede de um AP, estes alarmes são implementados com recurso ao envio de *traps* por parte dos dispositivos e desta forma a aplicação limita-se aguardar a recepção de *traps*. Esta abordagem funciona correctamente para o caso de uma alteração do estado da interface rádio, no caso da interface *ethernet*, isso só se verifica no caso de um alteração de estado para *up*.

Assim, a detecção de alteração do estado da interface *ethernet* de um AP para *down* é feita na operação de monitorização regular da rede, na qual se se detectar que um AP deixa de responder, é sinalizado um alarme.

SNMP

Optou-se por implementar as versões SNMPv1 e v2c, deixando a implementação de v3 para trabalho futuro pese embora já se tenha previsto essa possível implementação, e como tal os formulários e as estruturas de dados de parâmetros SNMP já têm lugar para os parâmetros SNMPv3.

Arquitectura Lógica

A arquitectura lógica da ferramenta está dividida em **camadas**, cada uma das quais contendo pacotes de classes agrupadas de acordo com a sua natureza ou função. Este tipo de arquitectura visa tornar a ferramenta modular e escalável possibilitando a reutilização do código escrito no seu todo ou em parte.

Importa ainda referir o facto de ser ter utilizado uma camada de *software* para fazer a interface entre a aplicação e a API que implementa o protocolo SNMP, assim, se por alguma razão se optar por utilizar outra API basta alterar as classes *CamadaSNMP* e *TrapListener_fgrw*.

Arquitectura Física

Optou-se por dividir a ferramenta em duas aplicações independentes, a aplicação gráfica de interacção com o utilizador, a qual poderá ou não estar sempre em execução e uma outra aplicação de recolha de estatísticas, esta aplicação deverá estar sempre em execução por forma a manter as bases de dados com as estatísticas actualizadas. A comunicação entre as duas aplicações é feita com base no ficheiro *lista.xml* com a lista de AP's e respectivos OID's sobre os quais se pretendem guardar estatísticas.

Optou-se por definir um ficheiro *RRD-Round Robin DataBase* para cada AP em detrimento de um ficheiro para todos os AP's, esta opção visa possibilitar a inserção e remoção de elementos na lista de AP's, uma vez que a estrutura de um ficheiro RRD depois de criada não pode ser alterada.

4.5.3 Eficiência

Para minimizar o tráfego de gestão gerado, apenas é pedida a informação estritamente necessária aos *access points*, nomeadamente, para a construção das tabelas

gráficas só é pedida informação para as colunas activas no momento. Paralelamente, sempre que é necessário solicitar o conteúdo de vários OID's a um dispositivo, estes são agrupados no mesmo pedido de forma a minimizar-se o *overhead* do protocolo SNMP.

Com o intuito de atenuar os *bursts* gerados, existe a possibilidade de na monitorização da rede, apenas ser monitorizado um AP em cada monitorização periódica, o qual vai alternando. Da mesma forma, na aplicação de recolha de estatísticas, os pedidos aos AP's são repartidos ao longo do tempo.

Com vista a obter uma resposta mais rápida por parte da aplicação, sempre que é necessário actualizar a tabela de AP's na sua totalidade é lançado um *thread* por cada elemento da lista, se não se optasse pela utilização de *threads*, seria necessário pedir a informação a um AP de cada vez, ficando todo o processo atrasado se um dispositivo demorasse mais responder.

4.5.4 Escrita de Código

Ao longo do desenvolvimento de todo código existiu a preocupação de respeitar as *boas práticas* de programação em Java de forma a facilitar a leitura e percepção do código escrito, nomeadamente a utilização de: letras maiúsculas na primeira letra de nomes de classes; letras minúsculas na primeira letra de nomes de pacotes de Classes, nomes instâncias e nomes de variáveis; em nomes de constantes apenas se utilizam letras maiúsculas; nomes longos em vez de abreviaturas.

5 Avaliação do trabalho

Para validar a ferramenta desenvolvida, testaram-se os casos de uso descritos anteriormente. Estes testes foram realizados com as aplicações a serem executadas numa máquina com acesso à rede de gestão dos *access points*. Inicialmente executou-se a aplicação gráfica, definiram-se os dispositivos que constituem a rede *wireless* (no caso a infra-estrutura wireless do IRICUP), exportou-se a lista de AP's para o ficheiro *lista.xml* e executou-se a aplicação de recolha de estatísticas. Em seguida são documentados os resultados obtidos para cada caso de uso.

5.1 Editar Lista de AP's

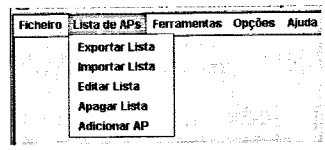


Figura 5.1 – Menu *Lista de AP's*.

Para definir ou editar a lista de AP's bastou executar a aplicação gráfica e aceder ao item *Editar Lista* do menu *Lista de APs*. Foi então aberto um diálogo idêntico ao representado na Figura 5.2, neste diálogo clicou-se no botão *Adicionar AP* e em seguida surgiu o diálogo representado na Figura 5.3, no qual se definem os parâmetros do *access point*. Depois de preenchidos todos os dados do formulário com valores válidos, clicou-se no botão *Ok* e o AP ficou definido na lista, aparecendo o respectivo IP no diálogo da Figura 5.2, a partir daí esse elemento, assim como outros que estivessem definidos poderiam ser alterados ou removidos. Seleccionou-se então um AP da lista e clicou-se no botão *Editar AP*, após esta acção surgiu o diálogo representado na Figura 5.4, em seguida o procedimento é idêntico ao descrito para adicionar um AP.

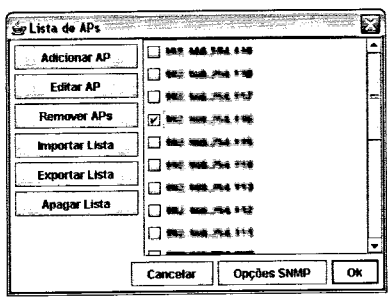


Figura 5.2 – Diálogo de edição de lista de AP's.

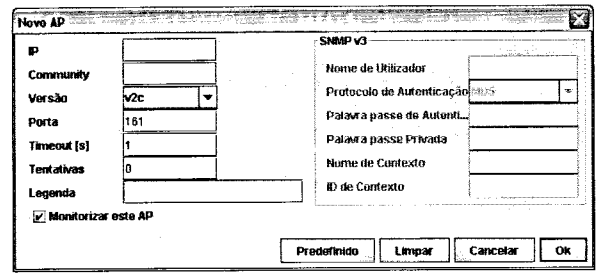


Figura 5.3 – Diálogo de adição de novo AP.

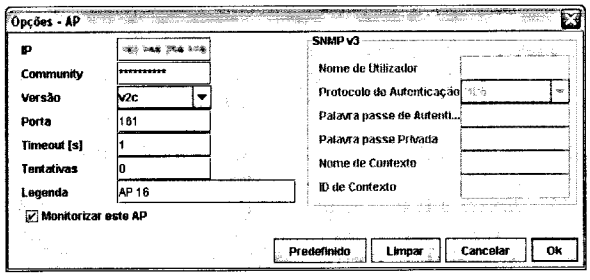


Figura 5.4 – Diálogo de edição de AP existente.

5.2 Monitorização de Rede

Para verificar este caso de uso, executou-se a aplicação gráfica com elementos definidos na lista, em seguida clicou-se no botão *Monitorizar.* A Figura 5.5 mostra o resultado obtido. Este resultado representa a concretização do caso de uso *Monitorização da Rede Periodicamente,* no qual a informação visualizada na tabela de *access points* é actualizada periodicamente. Da mesma forma, para testar o caso de uso *Monitorização da Rede Uma vez* , premiu-se o botão *Rastreio de Rede* em vez do botão *Monitorizar* e obteve-se o mesmo resultado gráfico.

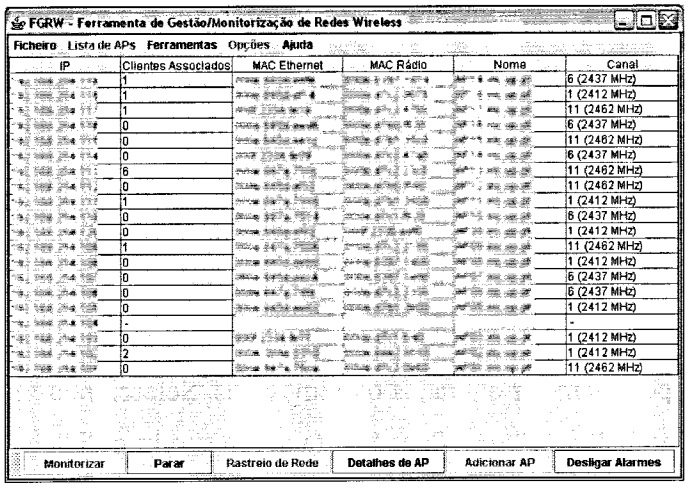


Figura 5.5 – Concretização de caso de uso *Monitorização da Rede Periodicamente*

5.3 Monitorização de Detalhes de AP

Para validar este caso de uso, executou-se a aplicação gráfica com AP's definidos na lista, clicou-se duas vezes numa das linhas da tabela representada na figura anterior. Em seguida surgiu o diálogo representado na Figura 5.6 no qual é disponibilizada diversa informação sobre um *access point,* para visualizar mais informação bastou experimentar as diversas abas presentes no diálogo, estão representados alguns exemplos destas nas figuras seguintes.

Para se visualizar informação sobre outro dispositivo, bastou selecciona-lo na *Combo Box* presente no fundo do diálogo.

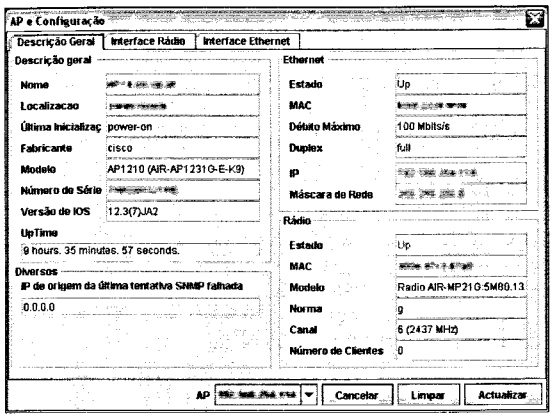


Figura 5.6– Concretização de caso de uso *Detalhes de AP - Informação Geral*

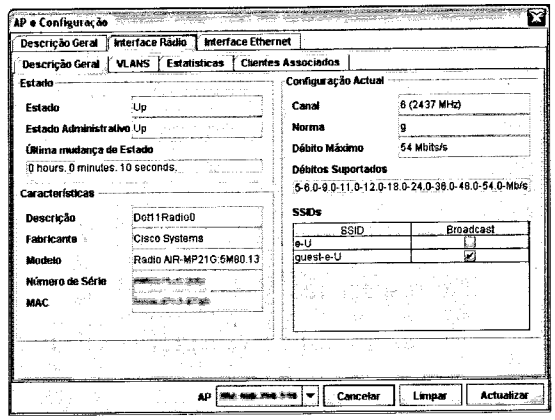


Figura 5.7 – Concretização de caso de uso *Detalhes de AP – Interface Rádio*

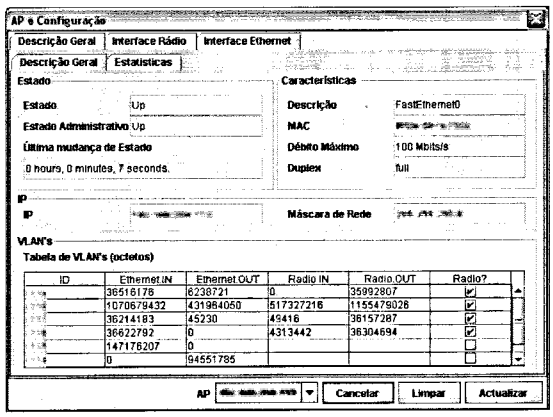


Figura 5.8 – Concretização de caso de uso *Detalhes de AP – Interface Ethernet – Tabela de VLAN's*

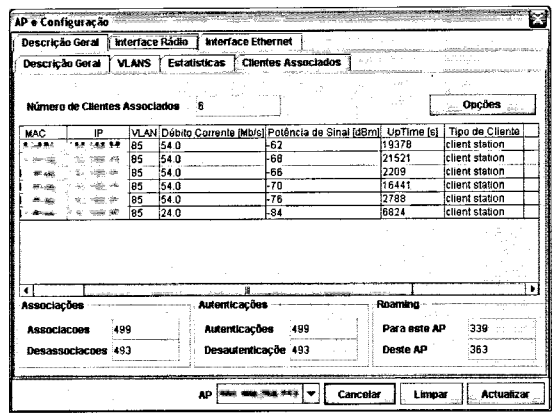


Figura 5.9 – Concretização de caso de uso *Detalhes de AP – Interface Rádio- Tabela de Clientes Associados*

5.4 Estatísticas

Para verificar este caso de uso executou-se a aplicação gráfica com alguns elementos com monitorização activa na lista, encerrou-se a aplicação para que a lista fosse guardada no ficheiro *lista.xml*, é importante referir que em alternativa poder-se-ia ter exportado manualmente a lista para o mesmo ficheiro. De seguida executou-se a aplicação de recolha de estatísticas. Esta começou por criar um ficheiro com uma base de dados RRD para cada dispositivo, posteriormente foi actualizando essas bases de dados de 2 em 2 minutos e ao final de uma hora de recolha, já existiam amostras suficientes que permitiam visualizar um gráfico completo. Para aceder aos gráficos

executou-se a aplicação gráfica e acedeu-se ao diálogo de *Detalhes de AP*, clicou-se na aba *Estatísticas* da aba *Interface Rádio* ou *Interface Ethernet*, como resultado desta operação obtivemos os gráficos presentes nas figuras seguintes. Em cada gráfico está disponível a visualização da média e do máximo da grandeza representada, ou apenas um destes; é também possível alterar a escala temporal para *Dia, Semana, Mês ou Ano* podendo especificar-se um dos extremos do gráfico com precisão até aos minutos.

No caso da interface Rádio estavam disponíveis os gráficos de tráfego na interface e respectivas VLAN's e ainda, o gráfico de clientes associados. Na interface *ethernet* estavam disponíveis gráficos de tráfego na interface e nas VLAN's.

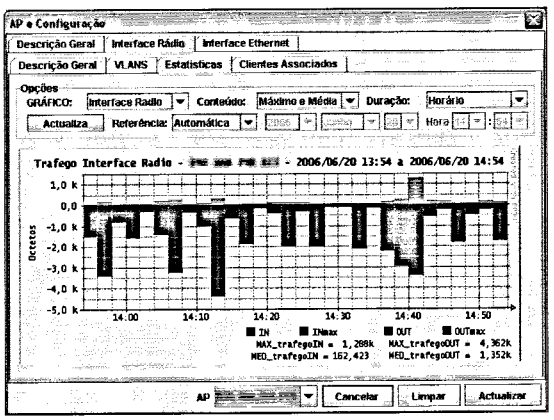


Figura 5.10 – Concretização do caso de uso *Estatísticas – Tráfego na interface rádio*

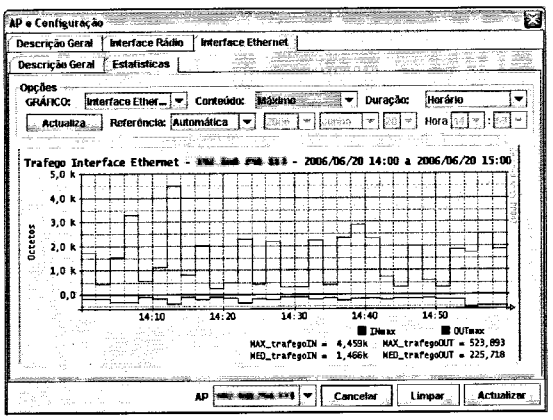


Figura 5.11– Concretização do caso de uso *Estatísticas – Tráfego na interface ethernet*

5.5 Alarmes

Para testar este caso de uso, configuraram-se as opções relativas aos Alarmes no menu *Opções*. De seguida clicou-se no botão *Ligar Alarmes* da janela principal, a partir desta acção a aplicação ficou a aguardar a recepção de *traps*. Para permitir o envio de *traps* por parte dos dispositivos, foi necessário efectuar a configuração respectiva que consiste basicamente na definição do endereço IP da máquina colectora onde está simultaneamente a correr a aplicação gráfica. Para forçar o envio de um *trap* por parte de um AP, acedeu-se ao mesmo por *telnet* e alterou-se o estado da interface rádio. As figuras seguintes mostram a sinalização de um *trap* recebido após a alteração do estado da interface rádio de um AP para *up*.

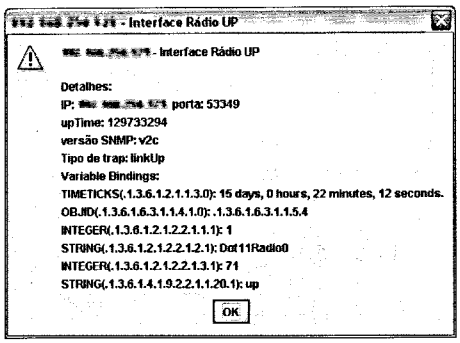


Figura 5.12 – Diálogo de sinalização da recepção de um *trap*.

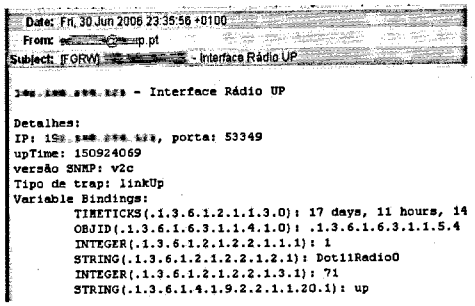


Figura 5.13 – *E-mail* de sinalização da recepção de um *trap*.

5.6 Configurar Aplicação

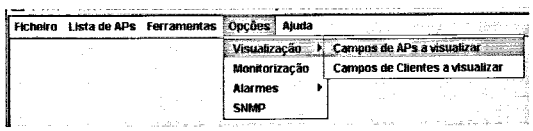


Figura 5.14 – Menu Opções / Visualização.

Para testar este grupo de casos de uso, executou-se a aplicação gráfica e experimentaram-se a diversas opções presentes no menu *Opções* da janela principal da aplicação. Todas as alterações efectuadas repercutiram-se no funcionamento da aplicação. As figuras seguintes mostram as diversas possibilidades de configuração da aplicação implementadas.

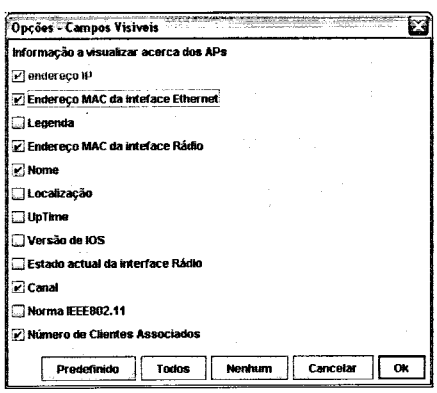


Figura 5.15 – Diálogo de configuração dos campos visíveis na tabela de AP's da janela principal.

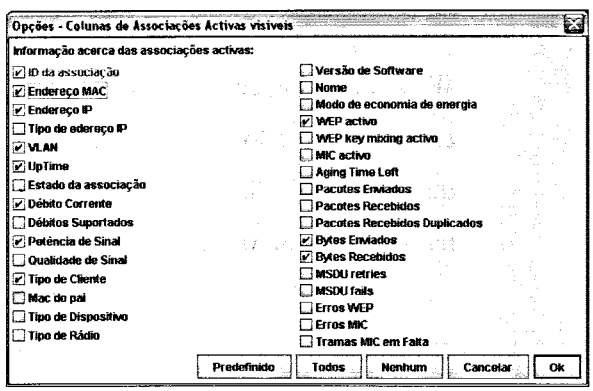


Figura 5.16 – Diálogo de configuração dos campos a visualizar na tabela de clientes associados do diálogo de detalhes de AP.

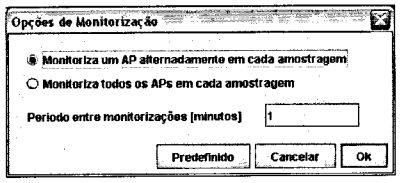


Figura 5.17 – Diálogo de configuração das opções de Monitorização.

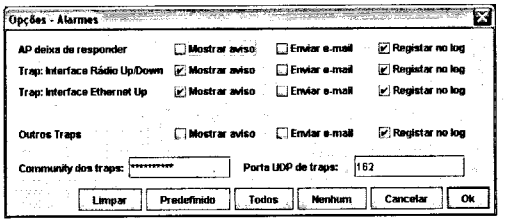


Figura 5.18 – Diálogo de configuração de alarmes.

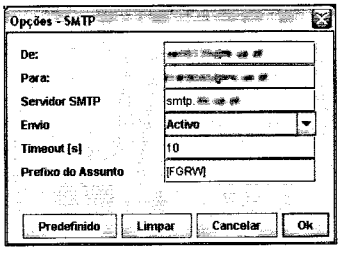


Figura 5.19 – Diálogo de configuração de parâmetros SMTP para sinalização de alarmes.

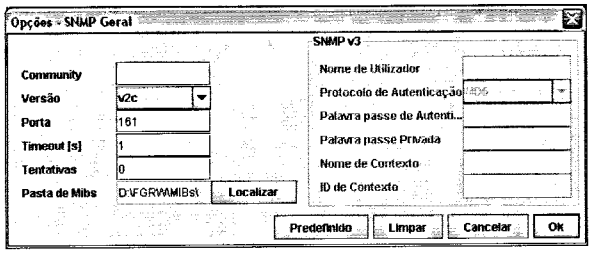


Figura 5.20 - Diálogo de configuração dos parâmetros SNMP gerais da aplicação

5.7 Ferramentas SNMP

Para verificar estes casos de uso executou-se a aplicação gráfica e experimentaram-se os três itens do sub menu *SNMP* do menu *Ferramentas.* No caso de das ferramentas *Get* ou *GetNext,* escolheu-se o AP sobre o qual se pretendia obter informação, especificou-se um OID numérico e clicou-se no botão *Get* ou *GetNext*, após estas acções obtiveram-se os conteúdos dos OID's especificados. No caso da ferramenta *Walk*, bastou escolher o AP pretendido e clicar no botão *Walk*, o resultado desta acção está representado nas Figuras 5.24 e 5.25, na primeira pode-se visualizar o diálogo em que é disponibilizada informação com o número de OID's percorridos e o OID actual, na segunda está representada uma parte do ficheiro de texto resultante da operação.

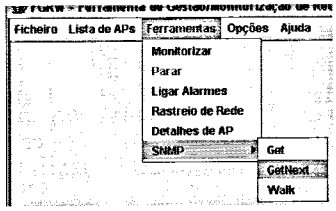


Figura 5.21 - Menu *Ferramentas.*

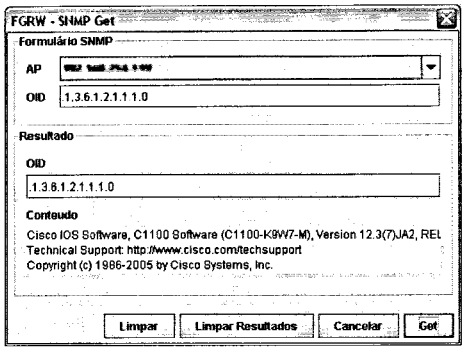


Figura 5.22 – Diálogo da ferramenta *SNMP Get.*

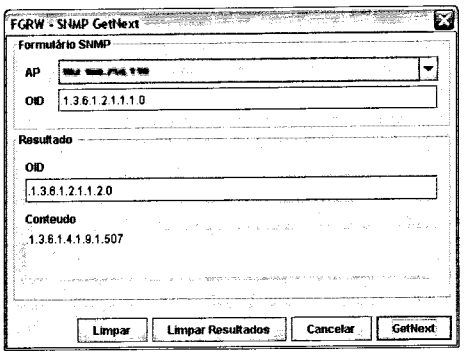


Figura 5.23– Diálogo da ferramenta *SNMP GetNext.*

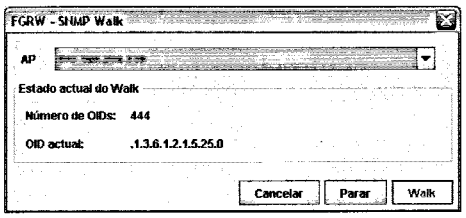


Figura 5.24 – Diálogo da ferramenta *SNMP Walk.*

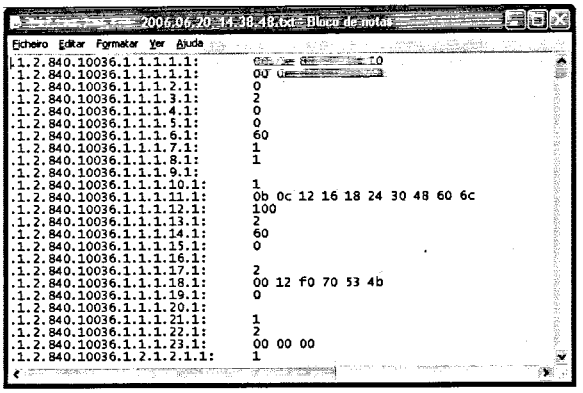


Figura 5.25 – Ficheiro de texto com o *resultado de um Walk.*

6 Conclusões

6.1 Revisão do trabalho desenvolvido

Todos os casos de uso foram implementados com sucesso. As funcionalidades desenvolvidas abrangem os diversos objectivos propostos, uma vez que esta ferramenta permite obter uma visão geral do estado e configuração da rede e, simultaneamente é possível aceder a informação mais detalhada sobre qualquer AP, tudo isto em tempo real. A ferramenta facilita a gestão de falhas, uma vez que implementa a sua detecção e sinalização, permitindo ainda o registo de um histórico destas. As funcionalidades relativas às estatísticas permitem ter uma visão sobre a utilização da rede, nomeadamente do tráfego transportado e do número de utilizadores facilitando a gestão do desempenho da rede.

Uma das vantagens desta ferramenta, é a simplicidade de utilização, a qual não exige conhecimentos profundos de SNMP ou de gestão de redes. Com esta ferramenta, a monitorização dos vários aspectos da rede fica à distância de alguns cliques. A definição dos *access points* que constituem a rede *wireless* também é simples, sobretudo se forem utilizadas as mesmas opções SNMP em todos os AP's. Neste caso, basta configurar as opções gerais SNMP e depois introduzir o IP de cada AP, a partir daqui a ferramenta está pronta a ser utilizada, note-se que a indicação dos AP's que constituem a rede só é necessária a primeira vez que se executa a aplicação.

Outra vantagem desta ferramenta, decorrente do seu desenvolvimento integral em Java é o facto de poder ser executada em diferentes sistemas operativos.

Ao utilizar a aplicação dever-se-á ter o cuidado de utilizar a *community* de leitura, uma vez que só estão implementadas as versões 1 e 2c do protocolo SNMP e, como tal, a *community* circula à vista pela rede nos pacotes SNMP, embora estes pacotes só circulem na rede de gestão, o que por si já melhora a segurança, no entanto será uma boa prática não usar a *community* de escrita.

6.2 Resultados / contribuições relevantes

6.2.1 Monitorização

A monitorização do estado e configuração actual de uma rede *wireless* é a funcionalidade mais desenvolvida desta ferramenta, uma vez que permite monitorizar a rede de diversas formas e com diferentes graus de detalhe. A forma de monitorização da

rede, pode ser configurada pelo utilizador da aplicação, ao qual cabe definir um compromisso entre obter uma informação actualizada da rede *wireless* e ao mesmo tempo causar o menor impacto possível na rede de local com tráfego de gestão.

6.2.2 Alarmes

A detecção de falhas e sua sinalização já estão implementadas. Já é possível configurar diferentes tipos de sinalização para diferentes tipos de alarmes. Contudo o número de tipos de alarmes definidos é reduzido. A detecção de uma passagem a *down* do estado da interface *ethernet* de um AP já é implementada pela detecção da ausência de resposta deste a um pedido SNMP, esta implementação apesar de detectar quando um *access point* deixa de responder, é susceptível de gerar falsos alarmes uma vez que basta que seja alterada a *community* de um *access point* para que este alarme seja activado, a solução para este problema passará pela implementação de uma ferramenta de *ping* que teste a ligação com o dispositivo antes de activar o alarme mencionado.

Uma outra questão a melhorar é o facto de a detecção de alarmes ser feita pela aplicação gráfica a qual nem sempre estará a correr, o ideal será passar esta funcionalidade para a aplicação de recolha de estatísticas uma vez que esta terá de estar sempre em execução.

6.2.3 Estatísticas

As funcionalidades relativas às estatísticas já disponibilizam informação sobre a utilização da rede, informação essa que poderá ser consultada em diferentes escalas temporais que, podem ir desde uma hora até um ano, podendo ser especificado o início ou final desses intervalos. Existem algumas lacunas existentes, nomeadamente o facto de os campos sobre os quais são guardadas estatísticas, apenas serem definidos quando se adiciona um AP à lista e, depois não podem ser alterados, outros inconvenientes são: o facto de aplicação de recolha de estatísticas, apenas consultar a *lista.xml* quando é iniciada e ainda o facto de este ficheiro apenas ser gerado quando se fecha a aplicação gráfica ou então tem de se exportar manualmente, para colmatar estas falhas: a aplicação gráfica deverá exportar a lista para o ficheiro xml sempre que esta for alterada, por sua vez a aplicação de recolha de estatísticas deverá consultar o ficheiro *lista.xml* antes de cada monitorização.

6.3 Trabalho futuro

Relativamente à **gestão de falhas**, sugere-se: que a recepção dos *traps* seja implementada na aplicação de recolha de estatísticas, seja efectuada uma diferenciação de novos tipos de *traps* e implementada uma ferramenta de *ping* para confirmação da ausência de resposta por parte de um *access point*. Deverá também ser implementada uma funcionalidade de recolha de *logs* gerados pelos AP's, os quais deverão ser guardados numa base de dados de forma a permitir pesquisas complexas.

Em relação á **informação disponibilizada** pela aplicação gráfica, sugere-se a diferenciação de número de clientes associados a um AP e número de clientes autenticados no mesmo, deverá ser também disponibilizada informação sobre o número de clientes associados por VLAN.

No que diz respeito ás **estatísticas**, sugere-se a implementação de um *access point virtual* que represente o conjunto de todos os dispositivos de forma a disponibilizar uma tabela de clientes associados em toda a rede e estatísticas resultantes da contribuições de todos os AP´s. Sugere-se ainda a adição de estatísticas sobre o número de clientes por VLAN e tempos de sessão por utilizador. Deverá também ser possível configurar os parâmetros das estatísticas, tais como o intervalo de tempo entre amostragens.

Relativamente à **configuração da aplicação**, é sugerida a exportação de todas as configurações para um ficheiro sempre que a aplicação é encerrada e a posterior importação quando esta for executada. Actualmente esta funcionalidade só é implementada para a lista de *access points*. Uma funcionalidade extra será a possibilidade de adição de novos campos à tabela de AP's inserindo os respectivos OID's.

Bibliografia

[1] STALLINGS, William – **SNMP, SNMPv2, SNMPv3, and RMON 1 and 2**
3rd ed. Canada: Addison-Wesley 1999. ISBN 0-201-48534-6

[2] SCHILLER, Jochen – **Mobile Communications**
1st ed. London: Addison-Wesley 2000. ISBN 0-201-39836-2

[3] CUNHA, António; LEITÃO, Jorge André; SERRÃO, Mário – **Segurança em Redes sem Fios**
2003/2004

[4] GUIDO, Luís – **Campus Virtuais, Arquitectura de Roaming Nacional**
Versão 1.0 : FCCN 22/10/2000

[5] RICARDO, Manuel Alberto Pereira – **LANs sem fios**
Apontamentos da cadeira *Comunicações Móveis* da LEEC
Porto: FEUP 2005/2006

[6] NEVES, João – **Gestão de Redes**
Apontamentos da cadeira *Planeamento e Gestão de Redes* do MRSC
Porto: FEUP 2005/2006

[7] NEVES, João – **SNMPv1**
Apontamentos da cadeira *Planeamento e Gestão de Redes* do MRSC
Porto: FEUP 2005/2006

[8] NEVES, João – **A Informação de Gestão**
Apontamentos da cadeira *Planeamento e Gestão de Redes* do MRSC
Porto: FEUP 2005/2006

[9] OLIVEIRA, Raul – **Gestão de Redes**
Apontamentos da cadeira *Arquitecturas de Redes e Serviços* da LEIC
Porto: FEUP 2005/2006

[10] **Cisco IOS Software Configuration Guide for Cisco Aironet Access Points**
Cisco IOS Release 12.3(7)JA USA: Cisco Systems 08/2005

[11] **SNMP Link**: [Em Linha]
Disponível em. *http://www.snmplink.org/*
[Consultado em 03/07/2006]

[12] **Simple Web**: [Em Linha]
Disponível em. *http://www.simpleweb.org/*
[Consultado em 03/07/2006]

[13] **Simple Network Management Protocol:** [Em Linha]
Disponível em. *http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm*
[Consultado em 03/07/2006]

[14] **Lessons About SNMP**: [Em Linha]
Disponível em. *http://www.et.put.poznan.pl/snmp/main/mainmenu.html*
[Consultado em 03/07/2006]

[15] **Java**: [Em Linha]
Disponível em. *http://java.sun.com/*
[Consultado em 03/07/2006]

[16] **Eclipse**: [Em Linha]
Disponível em. *http://www.eclipse.org/*
[Consultado em 03/07/2006]

APÊNDICES

A - Tecnologias

AdventNet SNMP API

A tecnologia *AdventNet SNMP API* é uma API em Java que implementa o protocolo SNMP. Suporta de SNMPv1, SNMPv2c e SNMPv3. Tem como principais vantagens as funcionalidades de alto nível que permitem uma abstracção dos detalhes do protocolo SNMP. Existe uma versão *Professional* e uma versão *free* que é uma versão limitada da primeira. (*http://snmp.adventnet.com/*)

RRDtool

RRDtool é uma ferramenta que facilita o arquivo e visualização de estatísticas. As principais vantagens desta tecnologia são o facto de utilizar bases de dados RRD de tamanho fixo, estas bases de dados são configuradas quando criadas, a partir dai basta inserir amostras periodicamente. Para obter gráficos de estatísticas apenas é necessário especificar a gama temporal dos mesmos sendo opcional a especificação de outros parâmetros. (*http://oss.oetiker.ch/rrdtool/*)

RRD4J

A tecnologia *RRD4J* é um API em Java que implementa as funcionalidades da ferramenta *RRDtool.* (*https://rrd4j.dev.java.net/*)

B - API's de SNMP para Java

Neste apêndice são referenciadas implementações em Java do protocolo SNMP livres para uso não comercial. As referências aqui encontradas englobam API e/ou projectos *open source*.

- **AdventNet SNMP API 4**
  http://snmp.adventnet.com/
- **SNMP4J**
  http://www.snmp4j.org/
- **Westhawk's Java SNMP stack**
  http://snmp.westhawk.co.uk/
- **Net-SNMPj**
  http://netsnmpj.sourceforge.net/
- **jManage**
  http://www.jmanage.org/
- **Jasmin Project**
  http://www.ibr.cs.tu-bs.de/projects/jasmin/
- **Debian-source java-snmp**
  http://packages.debian.org/testing/source/java-snmp
- **JMIB Browser**
  http://www.dwipal.com/mibbrowser.htm
- **open NMS**
  http://www.opennms.org
- **Simple SNMP Library**
  http://www.ccs.neu.edu/home/guol/snmp/README.html
- **DMH Software**
  http://www.dmhsoftware.com/snmp.html

C - Manual de Utilização

1 Utilização

Este manual serve de apoio a utilização da aplicação.

1.1 Aplicação gráfica de Monitorização

1.1.1 Iniciar Aplicação

Para iniciar a aplicação no *windows* basta aceder à pasta *FGRW* e clicar duas vezes no ficheiro *FGRWwindows.jar*. Após esta operação deverá aparecer a aplicação gráfica com um aparência idêntica à da Figura 13.

Para iniciar a aplicação em *Linux* é necessário abrir uma linha de comandos, aceder à pasta *FGRW* e executar o comando `java -jar FGRWlinux.jar`. Após esta operação deverá aparecer a aplicação gráfica com um aparência idêntica à da Figura 6.1

.

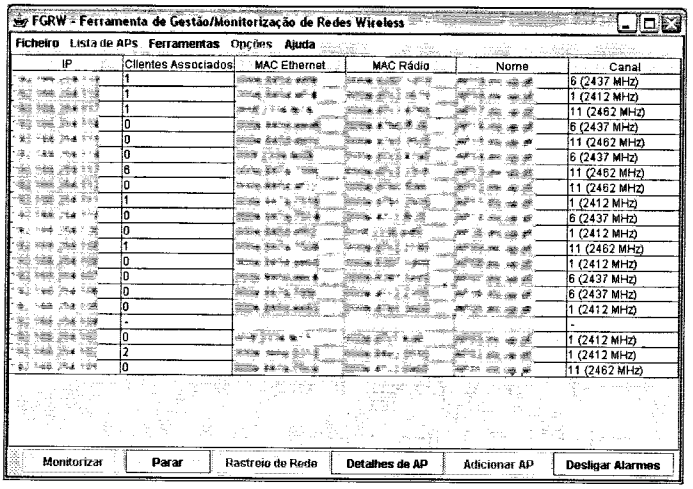


Figura 6.1 – Janela Principal

1.1.2 Configurar Aplicação

Todas as opções de configurações poderão ser acedidas através do menu *Opções* da janela principal. Em seguida apresenta-se uma descrição mais detalha sobre a forma de aceder a cada uma dessas opções.

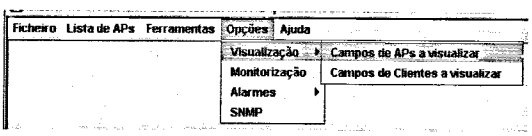


Figura 6.2 - Menu Opções / Visualização

Opções de Monitorização

Efectuar a seguinte sequência:

- Menu *Opções* / Opção *Monitorização*

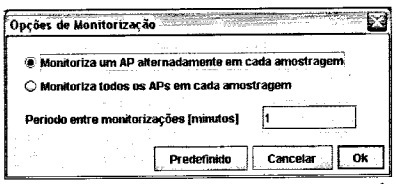


Figura 6.3 – Diálogo de edição de opções de monitorização.

Opções SNMP

Efectuar uma das seguintes sequências:

- Menu *Opções* / Opção *SNMP*
- Menu *Lista de APs* / Opção *Editar Lista de APs* / Botão *Opções SNMP*
- Botão *Editar Lista de APs* / Botão *Opções SNMP*

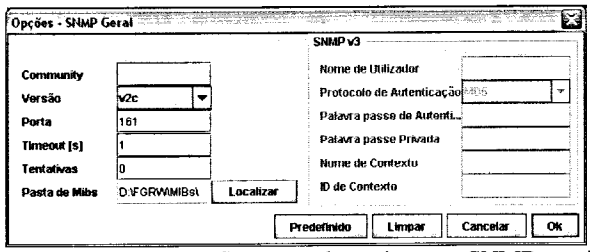


Figura 6.4 - Diálogo de configuração de parâmetros SNMP gerais.

Opções de Visualização

Colunas a visualizar na tabela de APs na janela principal

Efectuar a seguinte sequência:

- Menu *Opções* / Opção *Campos de APs a visualizar*

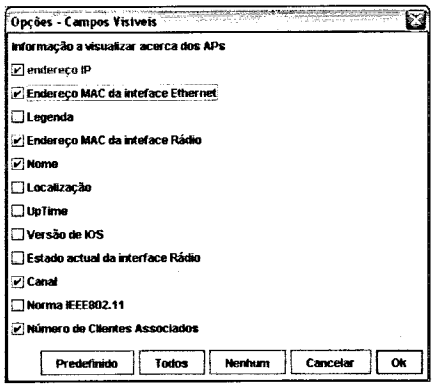


Figura 6.5 – Diálogo de configuração de opções de campos a na tabela de AP's.

Colunas a visualizar na tabela de associações activas do diálogo de detalhes de um AP.

Efectuar a seguinte sequência:

- Menu *Opções* / Opção *Campos de Clientes a visualizar*
- Diálogo de *Detalhes de AP* / Aba *Interface Rádio* / Aba *Clientes Associados* / Botão *Opções*

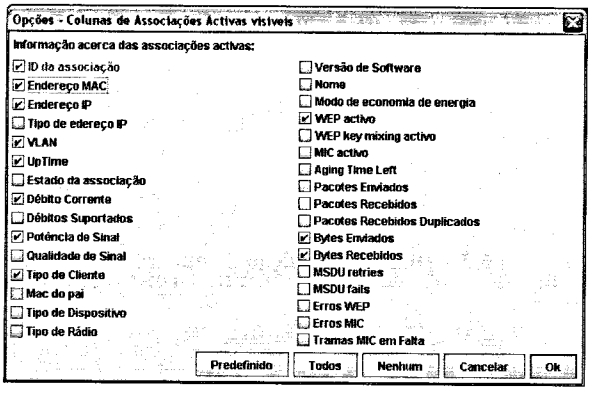


Figura 6.6 – Diálogo configuração de opções de campos a visualizar na tabela de clientes associados.

Alarmes

Para configurar os alarmes basta aceder ao sub-menu *Alarmes* do menu *Opções*.

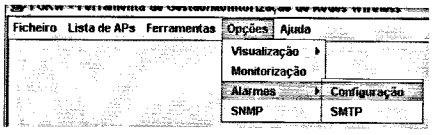


Figura 6.7 – Menu Alarmes.

Configuração de alarmes.

Efectuar a seguinte sequência:

- Menu *Opções* / Opção *Alarmes* / Opção *Configuração*

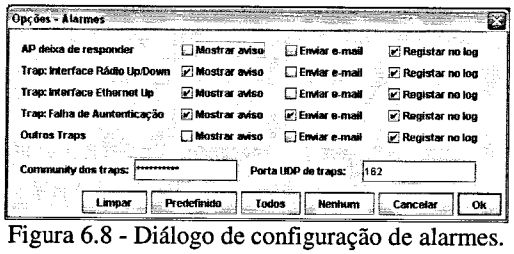


Figura 6.8 - Diálogo de configuração de alarmes.

Configuração de parâmetros SMTP.

Efectuar a seguinte sequência:

- Menu *Opções* / Opção *Alarmes* / Opção *SMTP*

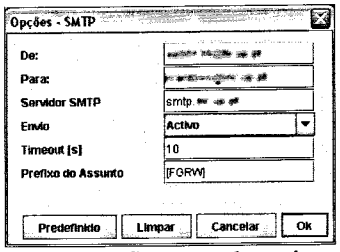


Figura 6.9 - Diálogo de configuração de parâmetros SMTP.

1.1.3 Editar Lista de APs

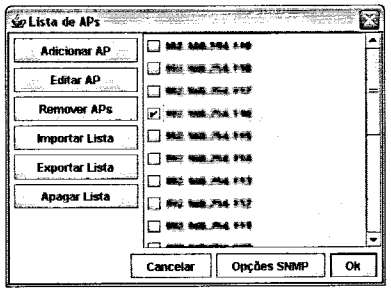


Figura 6.10 - Diálogo de edição da lista de APs.

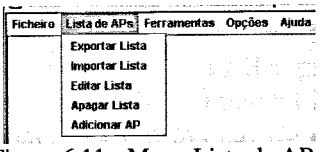


Figura 6.11 - Menu Lista de APs.

Adicionar AP

Efectuar uma das seguintes sequências:

- Botão *Adicionar AP*
- Botão *Editar Lista de APs* / Botão *Adicionar AP*
- Menu *Lista de APs* / Opção *Editar Lista de APs* / Botão *Adicionar AP*

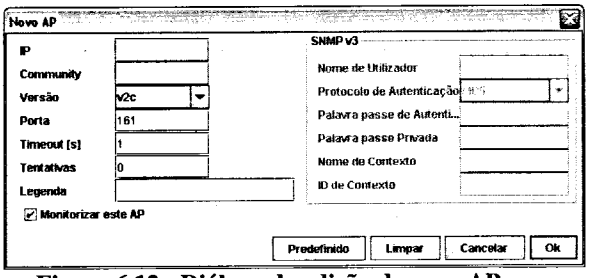


Figura 6.12 - Diálogo de adição de novo AP.

Editar AP

Efectuar uma das seguintes sequências:

- Menu *Lista de APs* / Opção *Editar Lista de APs* / seleccionar AP / Botão *Editar* AP
- Menu *Lista de APs* / Opção *Editar Lista de APs* / seleccionar AP / Botão *Editar* AP

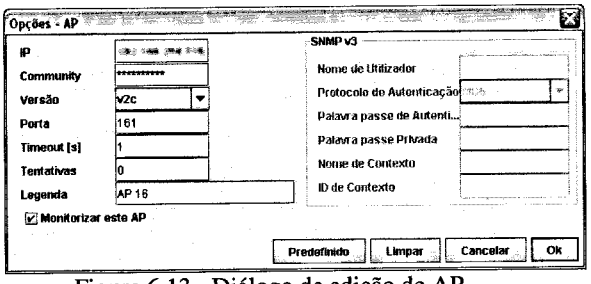


Figura 6.13 - Diálogo de edição de AP.

Remover APs

Efectuar uma das seguintes sequências:

- Botão *Editar Lista de APs* / seleccionar AP's / Botão *Remover APs*
- Menu *Lista de APs* / Opção *Editar Lista de APs* / seleccionar AP's / Botão *Remover APs*

Apagar lista de APs

Efectuar uma das seguintes sequências:

- Botão *Editar Lista de APs* / Botão *Apagar Lista*
- Menu *Lista de APs* / Opção *Editar Lista de APs* / Botão *Apagar Lista*

Importar lista

Efectuar uma das seguintes sequências:

- Botão *Editar Lista de APs* / Botão *Importar Lista*
- Menu *Lista de APs* / Opção *Importar Lista*
- Menu *Lista de APs* / Opção Editar Lista de APs / Botão *Importar Lista*
- Menu *Ficheiro* / Opção *Importar Lista de APs*

Exportar lista

Efectuar uma das seguintes sequências:

- Botão *Editar Lista de APs* / Botão *Exportar Lista*
- Menu *Lista de APs* / Opção *Exportar Lista*
- Menu *Lista de APs* / Opção *Editar Lista de APs* / Botão *Exportar Lista*
- Menu *Ficheiro* / Opção *Exportar Lista de APs*

1.1.4 Monitorizar Rede

Monitorizar uma vez

Efectuar uma das seguintes sequências:

- Botão *Rastreio de Rede*
- Menu *Ferramentas* / Opção *Rastreio de Rede*

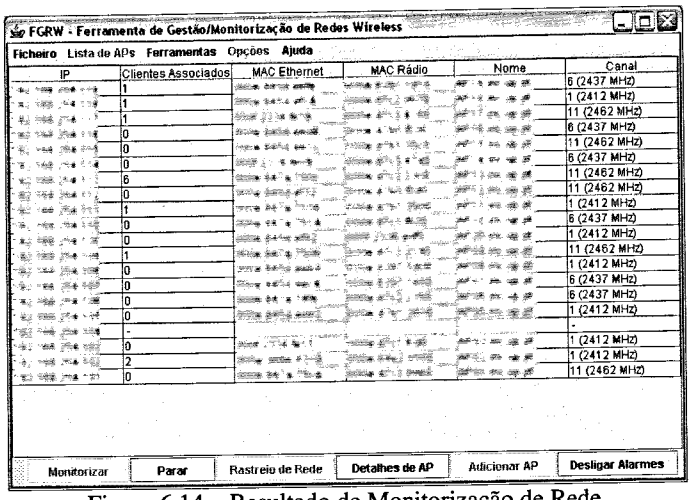


Figura 6.14 – Resultado de Monitorização de Rede.

Monitorizar Rede periodicamente

Iniciar Monitorização

Efectuar uma das seguintes sequências:

- Botão *Monitorizar*
- Menu *Ferramentas* / Opção *Monitorizar*

Parar Monitorização

Efectuar uma das seguintes sequências:

- Botão *Parar*
- Menu *Ferramentas* / Opção *Parar*

Ligar alarmes

Efectuar uma das seguintes sequências:

- Botão *Ligar Alarmes*
- Menu *Ferramentas* / Opção *Ligar Alarmes*

Desligar alarmes

Efectuar uma das seguintes sequências:

- Botão *Desligar Alarmes*
- Menu *Ferramentas* / Opção *Desligar Alarmes*

Visualizar detalhes de AP

Efectuar uma das seguintes sequências:

- Duplo clique numa das linhas da tabela de AP's
- Botão *Monitorizar AP*
- Menu *Ferramentas* / Opção *Monitorizar AP*

NOTA: nos dois últimos casos, se estiver alguma linha da tabela de AP's seleccionada, o AP correspondente vai ser monitorizado, se não, pode-se escolher o AP a monitorizar na *ComboBox* do diálogo que aparecerá em seguida.

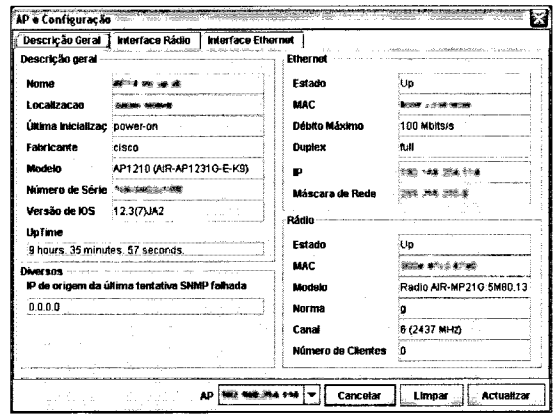


Figura 6.15 – Diálogo de detalhes de AP.

Visualizar estatísticas de AP

Visualizar estatísticas da interface ethernet

Efectuar uma das seguintes sequências:

- Duplo clique numa das linhas da tabela de AP's / Aba *Ethernet* / Aba *Estatísticas*
- Botão *Monitorizar AP* / Aba *Ethernet* / Aba *Estatísticas*
- Menu *Ferramentas* / Opção *Monitorizar AP*/ Aba *Ethernet* / Aba *Estatísticas*

Visualizar estatísticas da interface rádio

Efectuar uma das seguintes sequências:

- Duplo clique numa das linhas da tabela de AP's / Aba *Rádio* / Aba *Estatísticas*
- Botão *Monitorizar AP* / Aba *Rádio* / Aba *Estatísticas*
- Menu *Ferramentas* / Opção *Monitorizar AP*/ Aba *Rádio* / Aba *Estatísticas*

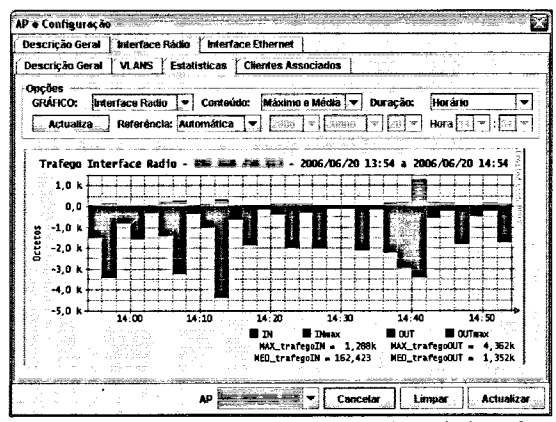


Figura 6.16– Diálogo de detalhes de AP – Estatísticas da interface Rádio.

1.1.4 Utilizar Ferramentas SNMP

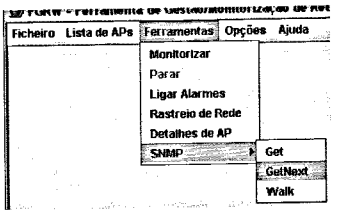


Figura 6.17 – Menu *Ferramentas.*

Get

Efectuar a seguinte sequência:

- Menu *Ferramentas* / Opção *SNMP* / Opção *Get*

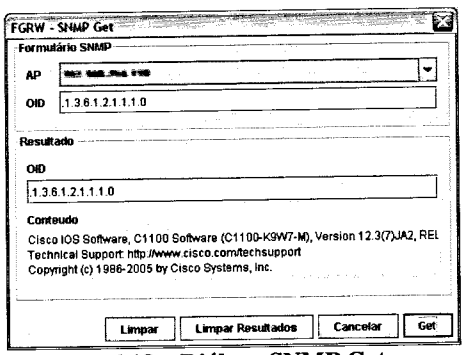


Figura 6.18 – Diálogo SNMP Get.

GetNext

Efectuar a seguinte sequência:

- Menu *Ferramentas* / Opção *SNMP* / Opção *GetNext*

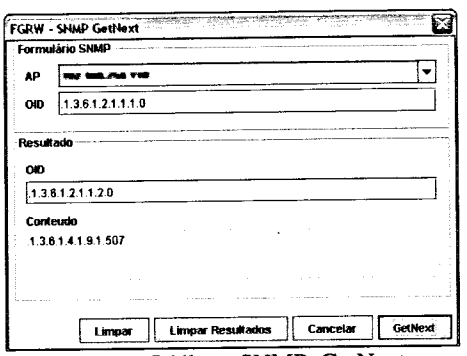


Figura 6.19 – Diálogo SNMP GetNext.

Walk

Efectuar a seguinte sequência:

- Menu *Ferramentas* / Opção *SNMP* / Opção *Walk*

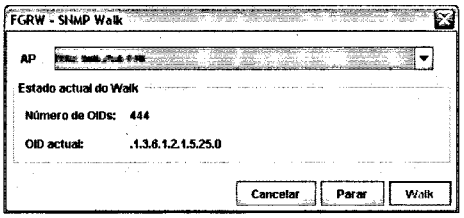


Figura 6.20 – Diálogo SNMP Walk.

1.2 Aplicação de Recolha de Estatísticas

1.2.1 Iniciar aplicação

Antes de executar esta aplicação e necessário executar a aplicação gráfica, definir os *access points* que constituem a rede *wireless* e sair da aplicação, para que a lista AP's seja exportada para o ficheiro *lista.xml*.

Para executar esta aplicação basta abrir uma consola, aceder á pasta *FGRW* e executar o comando definido no ficheiro LEIA-ME.txt que se encontra nessa pasta.

1.2.2 Parar aplicação

Para parar a aplicação basta escrever 'sair' na consola e pressionar a tecla 'Enter'.

2 Instalação

Para que a aplicação corra, é necessário ter o Java 1.5 instalado, para confirmar basta executar o comando `java -version` numa consola.

Para instalar a ferramenta, basta descompactar o ficheiro zip que contém. Após esta acção convém garantir que a pasta *jars* tenha permissões de leitura e de execução e que as restantes pastas tenham permissões de leitura e escrita.

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000105164