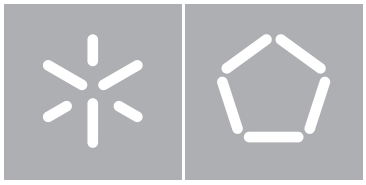
Universidade do Minho
Escola de Engenharia

Tiago Fontes Carvalho Duque da Silva

Gestão Remota para Pontos de Acesso de Redes Sem Fios

Outubro de 2013



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Tiago Fontes Carvalho Duque da Silva

Gestão Remota para Pontos de Acesso de Redes Sem Fios

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de
Professor Bruno Dias

Outubro de 2013

ANEXO 3:

DECLARAÇÃO

Nome: Tiago Fontes Carvalho Duque da Silva

Endereço electrónico: tiago.fcduque@gmail.com

Título dissertação: Gestão Remota para Pontos de Acesso de Redes Sem Fios

Orientador: Professor Bruno Dias

Ano de conclusão: 2013

Designação do Mestrado ou do Ramo de Conhecimento do Doutoramento: Mestrado em Engenharia Informática

---

Universidade do Minho, 30/10/2013

Agradecimentos

Desejo agradecer ao meu orientador Professor Bruno Dias, pela disponibilidade, atenção dispensada e dedicação.

Também agradeço a todos os professores, particularmente os do Departamento de Informática, pelo conhecimento transmitido ao longo da licenciatura e em especial do mestrado.

Agradeço também à minha família, em particular aos meus pais, pelo seu apoio e paciência, e também aos meus irmãos, tanto pelo apoio prestado como pela sabedoria que me foi passada.

Finalmente, agradeço aos meus amigos pela ajuda, bons momentos e boa disposição que proporcionaram ao longo deste trabalho.

Resumo

No âmbito de uma rede de um provedor de internet sem fios, faz todo o sentido afirmar que é essencial a existência de um sistema de monitorização com capacidades de acesso remoto e funcionalidades automatizadas. Desta forma, consegue-se reduzir a carga nos administradores da rede, bem como melhorar o tempo de resposta a vários eventos, tais como perda de rendimento da rede e aumento de colisões. Procura-se também que este sistema tenha baixas percentagens de uso da largura de banda. Para atingir esta finalidade, recorre-se a tecnologias normalizadas facilmente disponibilizadas como o SNMP ou NETCONF. Depois de um breve estudo comparativo entre as tecnologias referidas, serão analisadas em detalhe as MIBs mais relevantes relativamente a pontos de acesso sem fios. A existência de nodos escondidos, pela sua importância na degradação da largura de banda de redes sem fios, foi estudada em particular. Um dos algoritmos mais relevantes para a mitigação deste problema utiliza dinamicamente o mecanismo RTS/CTS através da monitorização de parâmetros, tais como o número de retransmissões e número de tramas com erros, activando-o tendo em conta os valores dos parâmetros monitorizados, evitando a introdução de overhead na rede devido ao seu uso desnecessário. Tal algoritmo foi introduzido na aplicação de gestão implementada e testada, sendo que os resultados obtidos não permitiram concluir da relevante bondade deste mecanismo quando aplicado somente do lado ponto de acesso.

Abstract

On the scope of an Internet service provider's wireless network, it makes all sense to declare that it is essential the existence of a monitoring system with remote access and automated capabilities. With such system it is possible to reduce the network administrators workload, as well as improve the response time to several events, like loss of throughput of the network or increasing collisions. Ideally that system would have low percentages of bandwidth usage. To achieve this, standardized technologies like SNMP or NETCONF will be used. Then it will take place a brief comparative study between those two technologies, followed by a detailed analysis of the most relevant MIBs present on wireless access points. The existence of hidden nodes, for its importance in bandwidth degradation in a wireless network, was studied in particular. One of the most relevant algorithms used for mitigation of this problem dynamically uses the RTS/CTS mechanism through parameter monitoring, such as number of retransmissions and number of frames with errors, activating the mechanism accordingly with the monitored parameters, avoiding overhead addition to the network caused by the unnecessary utilization of the mechanism. Such algorithm was introduced on the implemented and tested management application, though the obtained results didn't allow to achieve any conclusions relatively to the mechanism when applied only to the access point's end.

Sumário

Lista de Abreviaturas **ix**

Lista de Figuras **xiii**

Lista de Tabelas **xv**

1 Introdução **1**

1.1 **Organização do Documento** . . . 3

1.2 **Objectivos** . . . 3

2 Estado da Arte **5**

2.1 **Trabalhos Relacionados** . . . 5

2.2 **Protocolos de Controlo de Pontos de Acesso Sem Fios** . . . 8

2.3 **Switches/Controladores WLAN** . . . 9

2.4 ***Simple Network Management Protocol*** . . . 10

2.4.1 **Arquitectura** . . . 10

2.4.2 **Primitivas** . . . 11

2.4.3 **MIBs** . . . 13

2.4.4 **Versões** . . . 15

2.5 **Network Configuration Protocol** . . . 16

2.6 **Estudo Comparativo SNMP VS NETCONF** . . . 18

2.7 **Estudo de MIBS de Pontos de Acesso IEEE 802.11** . . . 23

3 Aplicação de Gestão - Resolução do Problema de Nodos Escondidos **27**

3.1 **O Problema dos Nodos Escondidos** . . . 27

3.2 **Estudos da Área e Soluções Propostas** . . . 30

3.3 **Algoritmo** . . . 32

3.4 **Implementação da Aplicação** . . . 38

3.5 **Detalhes da Implementação** . . . 41

4 Testes e Resultados **51**

5 Conclusões **61**
5.1 Trabalho Futuro 62

Referências Bibliográficas **65**

A Network Configuration Protocol **69**
A.1 **Operações** 71
A.2 **Aptidões (Capabilities)** 77
A.3 **Monitorização e outras considerações** 79
A.4 **Propostas em estado experimental** 84

B Estudo de MIBs de Pontos de Acesso IEEE 802.11 **87**
B.1 **IEEE802dot11-MIB** 87
B.2 **CISCO-DOT11-IF-MIB** 105
B.3 **CISCO-DOT11-ASSOCIATION-MIB** 119

Lista de Abreviaturas

ACK - Acknowledgement
ADSL - Asymmetric Digital Subscriber Line
AES - Advanced Encryption Standard
AID - Association ID
AP - Access Point
API - Application Programming Interface
ARIMA - Auto-Regressive Integrated Moving Average
ASCII - American Standard Code for Information Interchange
ASN.1 - Abstract Syntax Notation One
BEEP - Blocks Extensible Exchange Protocol
BER - Basic Encoding Rules
BSS - Basic Service Set
CAPWAP - Control and Provisioning of Wireless Access Points
CCA - Clear Channel Acessment
CCK - Complementary Code Keying
CCM - Counter with CBC-MAC
CCMP - CCM Mode Protocol
CFP - Contention Free Period
CKIP - Cisco Per Packet Key
CLI - Command Line Interface
CMIC - Cisco MMH Multi-Modal Hashing MIC Message Integrity Check
CP - Contention Period
CPU - Central Processing Unit
CR-MAC - Channel Reservation MAC
CRC - Cyclic Redundancy Check
CRUDX - Create Read Update Delete eXec
CS - Carrier Sense
CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance
CTS - Clear To Send
DCF - Distributed Coordination Function
DSSS - Direct Sequence Spread Spectrum
DTIM - Delivery Traffic Indication Message

DTLS - Datagram Transport Layer Security
EAP - Extensible Authentication Protocol
ED - Energy Detection
EHCC - Extended Hyperbolic Congruence Codes
ERP - Extended Rate PHY
ETSI - European Telecommunications Standards Institute
FCC - Federal Communications Commission
FCS - Frame Check Sequences
FHSS - Frequency Hopping Spread Spectrum
FLA - Fast Link Adaptation
FTP - File Transfer Protocol
HCC - Hyperbolic Congruence Codes
HRCS - High-Rate Carrier Sense
HRDSSS - High-Rate Direct Sequence Spread Spectrum
HTTP - Hypertext Transfer Protocol
HTTPS - Hypertext Transfer Protocol Secure
IAB - Internet Architecture Board
ICV - Integrity Check Value
IEC - International Electrotechnical Commission
IEEE - Institute of Electrical and Electronics Engineers
IETF - Internet Engineering Task Force
IOS - Internetwork Operating System
IP - Internet Protocol
IR - Infra-Red
ISO - International Organization for Standardization
LME - Layer Management Entity
LWAPP - Lightweight Access Point Protocol
MAC - Media Access Control
MDSU - MAC Service Data Unit
MIB - Management Information Base
MIC - Message Integrity Check
MODI - Model-based self Diagnosis
MOK - M-ary Orthogonal Keying
MPDU - MAC Protocol Data Unit
MTU - Maximum Transmission Unit
NACM - NETCONF Access Control Model
NETCONF - Network Configuration Protocol
NMS - Network Management System
OFDM - Orthogonal Frequency Division Multiplexing
OID - Object Identifier

ONF - Open Networking Foundation
OSI - Open Systems Interconnection
OUI - Organizationally Unique Identifier
PBCC - Packet Binary Convolutional Code
PC - Personal Computer
PCF - Point Coordination Function
PDU - Protocol Data Unit
PHY - Physical
PLCP - PHY Layer Convergence Procedure
PLME - Physical Layer Management Entity
PMD - Physical Medium Dependent
QoS - Quality of Service
RADIUS - Remote Access Dial In User Service
RF - Radio Frequency
RFC - Request For Comment
RoD-A - RTS on Demand using Acknowledgement
RoD-C - RTS on Demand using Collision rate
RoD-E - RTS on Demand using Error
RoD-R - RTS on Demand using Retransmission
RPC - Remote Procedure Call
RRM - Radio Resource Management
RSSI - Received Signal Strength Indication
RTID - Resource Type ID
RTS - Request To Send
RTS/CTS - Request To Send/Clear To Send
SBM - Structural Behavioral Model
SDN - Software Defined Networks
SIFS - Short Interframe Space
SMI - Structure of Management Information
SNAP - SubNetwork Access Protocol
SNMP - Simple Neywork Management Protocol
SOAP - Simple Object Access Protocol
SP - Sub-Period
SSH - Secure Shell
SSID - Service Set ID
STA - Station
TLS - Transport Layer Security
TTP - Trusted Third Party
TU - Time Unit
UDP - User Datagram Protocol

URL - Uniform Resource Locator
VLAN - Virtual Local Access Network
VoIP - Voice over IP
WEP - Wired Equivalent Privacy
WLAN - Wireless Local Area Network
WPA/WPA2 - WiFi Protected Access
XML - eXtensible Markup Language

Lista de Figuras

2.1 Comparação dos protocolos: Tempo de operação. Imagem retirada de [Hedstrom et al., 2011] . . . 21
2.2 Comparação dos protocolos: Utilização de largura de banda. Imagem retirada de [Hedstrom et al., 2011] . . . 22
2.3 Comparação dos protocolos: Número de transacções. Imagem retirada de [Hedstrom et al., 2011] . . . 22

3.1 Nodos escondidos e colisão. Imagem adaptada do artigo [Boroumand et al., 2012] . . . 28
3.2 Mecanismo RTS/CTS. Imagem retirada de [Gast, 2002] . . . 29
3.3 Diagrama de fluxo do RoD-A. Imagem adaptada do artigo [Chen and Vukovic, 2007] . . . 34
3.4 Diagrama de fluxo do RoD-R. Imagem adaptada do artigo [Chen and Vukovic, 2007] . . . 36
3.5 Diagrama de fluxo do RoD-C. Imagem adaptada do artigo [Chen and Vukovic, 2007] . . . 37
3.6 Diagrama geral do funcionamento do programa. . . . 43
3.7 Variáveis do objecto Data. . . . 43
3.8 Esquema do programa com pormenores das threads. . . . 44
3.9 Interface gráfica da aplicação de gestão. . . . 47
3.10 Execução do programa sem activação do algoritmo. . . . 48
3.11 Execução do programa com activação do algoritmo. . . . 48
3.12 Execução do programa com recomendação de activação do algoritmo, mas não o aplica. . . . 49

4.1 Gráficos de alguns dos testes efectuados. . . . 54
4.2 Testes com 6 a 10 *iPads*. (Imagem adaptada do estudo "Wi-Fi Stress Test") [Parsons, 2013] . . . 55
4.3 Testes com 11 a 15 *iPads*. (Imagem adaptada do estudo "Wi-Fi Stress Test") [Parsons, 2013] . . . 56
4.4 Testes com 16 a 20 *iPads*. (Imagem adaptada do estudo "Wi-Fi Stress Test") [Parsons, 2013] . . . 56

A.1 Camadas conceptuais do protocolo NETCONF 69
A.2 Diagrama da árvore monitorização NETCONF 80

Lista de Tabelas

4.1 Tempos de transferência em minutos, com e sem RTS/CTS. Sem nodos escondidos. 57
4.2 Tempos de transferência em minutos, com e sem RTS/CTS. Com nodos escondidos. 58

Capítulo 1

Introdução

Nos dias que correm, o acesso à Internet é, sem dúvida alguma, um serviço que se tem mostrado indispensável na vida de grande parte da população mundial. Exemplos desse facto são o crescente uso do correio electrónico, comércio online, redes sociais, motores de pesquisa, partilha de ficheiros, entre muitas outras aplicações.

Existe então a necessidade de apresentar soluções aos clientes para que estes possam tirar partido de todas estas funcionalidades, seja em casa, no trabalho ou em qualquer outro local. As soluções de acesso à Internet existentes são na sua maioria fornecidas por tecnologias fixas como cabo, ADSL e fibra. Apesar das suas excelentes prestações, estes métodos de acesso apenas podem ser utilizados numa zona restrita, normalmente em casa ou num escritório. Para o cliente obter mobilidade nas suas comunicações, pode recorrer a serviços de operadores móveis como 3G. Porém, estes serviços não são capazes de fornecer largura de banda suficiente em muitas situações.

Perante este cenário, começa a tornar-se evidente o compromisso entre a mobilidade e a velocidade do serviço. Neste contexto, faz sentido a existência de um provedor de serviços de Internet sem fios. Com este tipo de serviço o cliente não necessita de equipamento específico no local associado ao contrato, como acontece com os métodos fixos. Em vez disso basta estar numa zona em que haja cobertura, uma área metropolitana por exemplo, e efectuar o processo de *login* no seu computador ou *smartphone*. Uma das características mais importantes duma rede deste tipo é a sua capacidade de *roaming*. Isto permite que um cliente transite fisicamente de um ponto de acesso para outro, sem que haja interrupção das comunicações. Entra-se então no domínio das Redes de Área Metropolitana.

Quando um operador pondera a implementação de um serviço e toda a sua infra-estrutura entram em jogo vários factores económicos e financeiros. O seu interesse principal será a contenção e controlo dos custos de instalação e operação da rede. Um dos aspectos a ter em atenção, para além do capital aplicado em equipamento, é a contratação de técnicos e ad-

ministradores de rede. Como se sabe, estes elementos são essenciais à correcta configuração e bom funcionamento do serviço. Uma maneira de reduzir gastos neste campo, bem como tempo empregue nessas actividades, será atribuir capacidades de configuração automatizada e monitorização remota ao equipamento do operador. Com isto evitam-se demoras na resolução de problemas de configurações, bem como melhoras no tempo de resposta a várias situações de monitorização, como uso impróprio, desastres ou falhas do equipamento.

Para atingir este objectivo é indispensável que os fabricantes usem tecnologias normalizadas no equipamento destinado ao operador. Desta forma o operador tem maior facilidade e liberdade em desenvolver aplicações de gestão e monitorização da rede, eliminando a dependência de normas e tecnologias proprietárias impostas por muitos fabricantes, situação conhecida por *vendor lock-in* ou *proprietary lock-in*.

Uma das tecnologias para gestão de equipamento de rede é o SNMP que, para além de ser usualmente disponibilizada pelos fabricantes, é também a tecnologia base mais utilizada para implementação de aplicações de gestão. É usada maioritariamente para monitorizar condições que necessitem de atenção operacional. Há que ter também em consideração outras tecnologias que, apesar de não serem usadas em tão grande escala, se têm apresentado como alternativas ou complementos, como é o caso do NETCONF, que fornece mecanismos para a instalação e manipulação de configurações em equipamento de rede.

O presente trabalho assumiu como objectivo estudar o problema da gestão de pontos de acesso sem fios (normalmente realizada com acessos web) mediante a utilização de tecnologias próprias para gestão de redes. O ponto de partida foi que a tecnologia preferencial seria o SNMP. Mesmo assim, para o estudo deste problema mostrou-se importante desenvolver um estudo comparativo das tecnologias disponíveis: SNMP e NETCONF. Foram estudadas em detalhe MIBs SNMP cujo conhecimento obtido foi aplicado no desenvolvimento de uma aplicação de gestão. Tal aplicação tem em vista a detecção e mitigação do problema dos nodos escondidos.

Este problema afecta o rendimento e eficiência de uma rede, sendo uma das causas o aumento de colisões de pacotes. Por essa razão foi desenvolvido trabalho no sentido de solucionar ou melhorar as consequências resultantes da existência de nodos escondidos numa rede *Wi-Fi*. Na abordagem ao problema, foi escolhida como estratégia a utilização dinâmica do mecanismo RTS/CTS. Este é activado ou desactivado conforme os valores de erros e retransmissões monitorizados na aplicação de gestão implementada.

Após o desenvolvimento da aplicação de gestão, foram realizados testes onde foi simulada a situação de nodos escondidos. Estes testes permitiram concluir que a utilização dinâmica do mecanismo RTS/CTS traz melhorias ao rendimento da rede, nomeadamente em relação

a tempos de transferência. Contudo não é possível aplicar na totalidade a aplicação desenvolvida visto apenas ser possível monitorizar erros e retransmissões por SNMP no lado dos pontos de acesso. Isto acontece devido à falta de suporte de MIBs mais avançadas nas placas de rede dos dispositivos dos clientes. Pela mesma razão não é possível activar ou desactivar o mecanismo RTS/CTS automaticamente através de SNMP alterando o valor do objecto *dot11RTSThreshold*.

1.1 Organização do Documento

Após a introdução deste documento, são expostos os objectivos que foram inicialmente propostos para esta dissertação. Em seguida, na secção do estado da arte, são abordados vários aspectos relacionados com as tecnologias existentes nesta área. Destas destacam-se o protocolo SNMP e NETCONF e os protocolos de controlo de pontos de acesso sem fios, o LWAPP (Lightweight Access Point Protocol) e o CAPWAP (Control and Provisioning of Wireless Access Points) seguido de algumas considerações sobre *comutadores* e controladores VLAN. São também apresentados resultados sobre o estudo comparativo entre o protocolo SNMP e o protocolo NETCONF. Como preparação à implementação da aplicação proposta neste trabalho, são analisadas em detalhe as MIBs geralmente suportadas pelos pontos de acesso sem fios. A secção seguinte é dedicada aos pormenores de implementação da aplicação de gestão desenvolvida, que foca a sua atenção no problema dos nodos escondidos, seguida de uma secção sobre os testes efectuados e discussão dos resultados. Finalmente são apresentadas as conclusões deste trabalho.

1.2 Objectivos

No início dos trabalhos de dissertação foram estabelecidos os seguintes objectivos concretos:

1. Estudar trabalhos relevantes de investigação e desenvolvimento na área de gestão remota e automatizada de redes da norma IEEE 802.11.

2. Um estudo comparativo sobre as duas principais tecnologias para gestão de redes, o SNMP e o NETCONF e que são normalmente suportados em pontos de acesso a redes *Wi-Fi*, com vista à escolha da tecnologia mais adequada para a implementação da aplicação de gestão.

3. Estudar as MIBs normalizadas mais importantes para a gestão de pontos de acesso, de forma a obter um conhecimento mais profundo sobre os objectos SNMP disponíveis para a criação de uma aplicação de configuração e monitorização.

4. Focar o processo da gestão de pontos de acesso *Wi-Fi* no problema específico de detectar e limitar os efeitos adversos do fenómeno de nós escondidos.

5. Definir uma estratégia para tentar resolver o problema de nodos escondidos, e implementar um protótipo para gerir pontos de acesso sem fios. Deveriam ser efectuados testes para determinar se a solução escolhida ajuda a diminuir as consequências causadas pelo problema de nodos escondidos.

6. Concluir sobre a adequabilidade do uso do SNMP como tecnologia de gestão para pontos de acesso *Wi-Fi*.

Capítulo 2

Estado da Arte

Neste capítulo são apresentados alguns trabalhos recentes relacionados com a área. Também é feita uma descrição das tecnologias existentes, nomeadamente, o SNMP para gestão de redes sem fios e serviços de comunicação. Posteriormente são apresentados os estudos sobre SNMP e NETCONF, seguindo-se de um estudo comparativo de ambos os protocolos. No fim é efectuado um estudo sobre MIBs disponibilizadas para pontos de acesso da norma IEEE 802.11.

2.1 Trabalhos Relacionados

De acordo com o estudo desenvolvido por [Raghavendra et al., 2008], a gestão de uma rede sem fios de grande escala, apresenta numerosos desafios. A resolução de problemas relacionados com o acesso sem fios neste tipo de redes requer a compreensão de um conjunto de métricas ou parâmetros, bem como dados de monitorização da rede. Actualmente, as soluções existentes recolhem grandes quantidades de dados e utilizam uma porção significativa de largura de banda e poder de processamento para transferência e análise dos mesmos. Como consequência, estas soluções não são escaláveis e não são efectuadas em tempo real. Sendo assim, é apresentada uma abordagem multi-camada que controla a granularidade da recolha de dados baseada em eventos observados na rede. Desta forma, pode-se reduzir a largura de banda usada para este fim e possibilitar a gestão automatizada em tempo real.

Na proposta de [Zhao, 2008] é apresentada uma abordagem ao diagnóstico automatizado de falha de ligações. Esta proposta baseia-se no uso das probabilidades de falha de uma ligação ou nó da rede. O sistema constrói dinamicamente uma tabela de *ranking* das ligações e nós conforme a sua probabilidade de falhar. São depois efectuadas verificações conforme essa tabela, ou seja, quanto maior a probabilidade de uma falha, maior é a frequência de verificações. Com isto evita-se a frequente verificação de nós ou ligações que não costumam falhar, poupando largura de banda e aumentando a precisão da detecção das falhas.

O trabalho desenvolvido por [Chadha, 2004] põe em evidência que para proporcionar a um sistema de gestão de redes características de automatização é necessário que este seja capaz de reagir a eventos, tanto em relação à monitorização da rede, como em relação à reconfiguração da rede. Uma das formas de dotar o sistema com essa capacidade é através do uso de políticas pré-determinadas. Essas políticas são criadas pelo administrador da rede. Uma vez criadas e armazenadas no sistema de gestão, são automaticamente aplicadas. Com isto, avança-se mais um passo a caminho de uma rede completamente automatizada, eliminando a necessidade da constante atenção de um administrador ou equipa de administradores.

Segundo um estudo da área de investigação de gestão de redes automatizada, mais propriamente em relação à tecnologia NETCONF, desenvolvido por [Weinstein et al., 2011] é afirmado que tem havido progressos significativos na área. No entanto, os estudos têm-se focado em dispositivos de rede específicos, e não na rede como um todo. Com este aspecto em vista, um modelo robusto que mapeie as propriedades agregadas de uma rede para entidades NETCONF específicas aos dispositivos, permitiria a um provedor de serviços abstrair o desenho da rede da configuração dos dispositivos. Assim, aumenta-se a capacidade de automatização da configuração da rede no que diz respeito a agregação de políticas, topologia da rede e requisitos de serviços.

Já no campo da segurança, o trabalho de [Yaacob et al., 2010] propõem um sistema que, pode prever situações de potenciais ataques a rede emitindo avisos atempadamente, para que o administrador ou até mesmo um sistema de gestão automatizado possa tomar as medidas necessárias. É sugerida uma abordagem através do uso de uma técnica chamada *Auto-Regressive Integrated Moving Average* (ARIMA). Foram obtidos resultados positivos, e com a continuação do trabalho desenvolvido pode chegar-se a uma solução de segurança automatizada.

Na área de diagnóstico de falhas em redes sem fios, foi desenvolvido por [Yan and Chen, 2009] um trabalho muito interessante, o Model-based self Diagnosis (MODI), que focaliza a sua atenção na detecção e localização de falhas. A solução usa o Modelo Estrutural e Comportamental (*Structural Behavioral Model* - SBM) que é construído tendo em conta o modo operacional das especificações do protocolo sem fios e também estatísticas comportamentais. Foram construídos e implementados pontos de acesso sem fios embebidos com o MODI, que têm a capacidade de detectar tanto ataques de segurança como problemas de performance na rede. Estes pontos de acesso têm também como funcionalidade diagnosticar problemas de outros pontos de acesso, como os causados pela mobilidade dos dispositivos. Este modelo é rápido e eficaz, com pouco *overhead*.

Os autores [Biri and Afifi, 2008] tomam como ponto de partida do seu trabalho a seguinte premissa. Dada a necessidade de um provedor de serviços de internet sem fios tornar a sua

rede segura, estabelecer uma ligação entre um ponto de acesso e o dispositivo do cliente é um grande desafio. Com este ponto em vista, foi estudado um protocolo inter-camada de emparelhamento de dispositivos numa rede sem fios. Para além disso, foi também proposto um mecanismo que força a segurança da comunicação através da encriptação do endereço MAC em cada *frame*.

Como é afirmado no estudo de [Zhu et al., 2007] e respectivos colaboradores, uma das características importantes de uma rede sem fios é a sua capacidade de *roaming*. No entanto, com o aparecimento de provedores de serviços de internet sem fios, surge um novo aspecto a considerar, o *roaming* entre diferentes provedores. Com este aspecto em vista, foi proposta uma arquitectura de *roaming* inter-provedor baseada em *Trusted Third Party* (TTP). Esta abordagem tem como objectivo melhorar e preservar a privacidade e identidade dos utilizadores entre provedores distintos, bem como reduzir o tamanho da base de dados central para minimizar possíveis abusos do serviço. É também introduzido um esquema de *billing*, para resolver problemas relacionados com o *roaming*. Para além disto, é proposto um esquema de autenticação inter-provedor para suportar uma transição contínua e suave.

No trabalho realizado por [Li and Chen, 2006] fala-se do uso de SNMP (Simple Network Management Protocol) na construção de sistemas de gestão de redes, onde existe trabalho realizado que tem em vista a sua aplicação a redes sem fios. Nesse estudo é discutida a melhor forma de melhorar sistemas de gestão existentes para se adequarem aos requisitos de uma rede sem fios. O sistema tem as seguintes funcionalidades: descoberta automática de dispositivos, associação entre dispositivos móveis e pontos de acesso, recolha de dados para monitorização do desempenho da rede, controlo de acesso através do ponto de acesso e notificação de mudanças na rede.

Ainda dentro da temática do SNMP e gestão de redes, no estudo de [Yen and Yeh, 2006] é feita uma abordagem à distribuição de carga em redes sem fios usando SNMP. Tendo em atenção o facto de que as estações sem fios (dispositivos do utilizador) não têm qualquer critério de selecção de ponto de acesso no momento de associação, a carga de tráfego não é partilhada justamente entre os pontos de acesso disponíveis. Abordagens anteriores precisavam de modificar o comportamento dos pontos de acesso ou negociar largura de banda, no entanto, isto não é prático devido à dificuldade de aplicação a pontos de acesso já existentes. A nova proposta traz uma abordagem à camada aplicacional, onde um servidor dedicado recolhe informação relativa à distribuição de carga entre os pontos de acesso usando SNMP. Isto traz vantagens, nomeadamente no facto de poder ser aplicada em pontos de acesso *standard* disponíveis no mercado.

2.2 Protocolos de Controlo de Pontos de Acesso Sem Fios

O LWAPP (*Lightweight Access Point Protocol*) é um protocolo que tem como função controlar múltiplos pontos de acesso sem fios, permitindo reduzir o tempo empregue na configuração, monitorização ou resolução de problemas numa rede de grandes dimensões. O sistema é instalado num servidor central que recolhe dados de dispositivos de marcas e funcionalidades diferentes. Este servidor pode controlar um determinado grupo de dispositivos e aplicar configurações simultaneamente. Sendo um protocolo proprietário (*Airespace/Cisco*), os sistemas LWAPP competem com outros mecanismos de empresas como a *Meru Networks* e *Aruba Networks*. O protocolo é definido de forma a ser independente de tecnologia da camada 2, no entanto, *binding* para 802.11 é fornecido para utilização em redes sem fios.

Tem como objectivos a centralização das funções de *bridging*, encaminhamento, autenticação e cumprimento de políticas numa rede sem fios. Opcionalmente, o sistema de controlo de acesso pode fornecer encriptação centralizada do tráfego do utilizador. Com isto, é possível atingir custos reduzidos e maior eficiência na aplicação destas funcionalidades. Também como objectivo vem a deslocação da carga de processamento dos pontos de acesso para o sistema de controlo, aumentando os recursos disponíveis nos pontos de acesso. Finalmente, procura-se também fornecer encapsulação genérica e um mecanismo de transporte, de forma a ser aplicado a outros tipos de pontos de acesso no futuro. [Calhoun et al., 2010]

Baseado no protocolo anterior o LWAPP, o CAPWAP (*Control and Provisioning of Wireless Access Points*) é também um protocolo para controlo de múltiplos pontos de acesso sem fios. As especificações deste protocolo estão presentes no RFC 5415 [Calhoun et al., 2009b] e o *binding* com o IEEE 802.11 é descrito no RFC 5416 [Calhoun et al., 2009a]. O protocolo focaliza-se na facilidade de uso através de um sistema central de gestão, aumento de segurança com decisão de políticas centralizada, melhorias no campo da mobilidade e fornecendo mecanismos de balanceamento de carga entre pontos de acesso.

Este protocolo tem objectivos idênticos ao LWAPP. Tem em vista a centralização das funções de autenticação e cumprimento de políticas de uma rede sem fios. O sistema de controlo de acesso fornece também centralização das funções de *bridging*, encaminhamento e encriptação do tráfego do utilizador. Tenta retirar a carga de processamento dos pontos de acesso, deixando apenas aplicações de tempo crítico de controlo de acesso, fazendo uso eficiente do poder de computação disponível nos pontos de acesso que estão sujeitos a pressões de custo neste aspecto. É fornecido um protocolo extensível que não está limitado a uma tecnologia sem fios específica. [Calhoun et al., 2009b]

2.3 Switches/Controladores WLAN

Nesta secção é feita uma breve análise aos dispositivos conhecidos por Controladores WLAN ou *Switches* WLAN. Estes dispositivos têm como função gerir e controlar múltiplos pontos de acesso sem fios.

Em geral, nas redes em fios são implantados três tipos de arquitectura, Arquitectura Autónoma, Arquitectura Centralizada e Arquitectura Distribuída. [Sridhar, 2006] Na Arquitectura Autónoma, os pontos de acesso implementam na totalidade as funcionalidades da norma IEEE 802.11. Cada ponto de acesso pode ser gerido como uma entidade separada na rede. Os pontos de acesso presentes nesta arquitectura são conhecidos por *Fat AP*. No decorrer dos primeiros anos da implantação de uma rede sem fios, a maioria dos pontos de acesso eram *Fat APs*, mas com o decorrer dos anos, as arquitecturas centralizadas ganharam bastante popularidade.

A vantagem principal de uma Arquitectura Centralizada é a capacidade de controlo de múltiplos pontos de acesso de forma estruturada e hierárquica. Este tipo de arquitectura hierárquica envolve a introdução de um controlador/*switch* (também conhecido por Controlador de Acesso) responsável pela configuração, controlo e gestão de vários pontos de acesso. As funções da norma 802.11 são partilhadas entre os pontos de acesso e os controladores. Com isto consegue-se a deslocação de carga de processamento dos pontos de acesso para os controladores, sendo os pontos de acesso presentes nesta arquitectura conhecidos por *Thin APs*. Esta será a arquitectura adoptada neste trabalho.

Na arquitectura distribuída, os vários pontos de acesso podem formar redes distribuídas com outros pontos de acesso através de ligações com ou sem fios. Uma rede *mesh* de pontos de acesso é o exemplo de tal arquitectura.

Tendo em conta que no trabalho será utilizada uma Arquitectura Centralizada, de seguida são apresentados alguns aspectos sobre os pontos de acesso *Thin APs* bem como os seus controladores. Tal como o seu nome indica, com os *Thin APs* pretende-se a redução da complexidade dos pontos de acesso. Uma das motivações para esta redução é a localização dos pontos de acesso. Na maioria das empresas os pontos de acesso são montados de forma a fornecer conectividade e sinal óptimos para as estações do cliente, no entanto colocados em locais de difícil acesso. Em ambientes como armazéns isto ainda é mais evidente. Por esta razão, os gestores de rede preferem instalar os pontos de acesso de uma vez só e não ter de realizar manutenção complexa. Este tipo de pontos de acesso são também conhecidos por "antenas inteligentes", visto que a sua principal função é receber e transmitir tráfego, transportando os dados para um controlador onde são processados antes de serem comutados para a rede principal. Em relação a este transporte, os pontos de acesso não efectuam

qualquer tipo de encriptação como WEP ou WPA/WPA2. Esta tarefa é efectuada apenas no controlador. Na altura de uma introdução de novos mecanismos de segurança como WPA2 (não retrocompatível com WEP), em vez de ter que fazer o *upgrade* de *firmware* nos pontos de acesso ou até mesmo subsitui-los, apenas é necessário fazer esta operação no controlador. [Sridhar, 2006]

O próximo componente crítico é o já referido anteriormente Controlador de Acesso. Para o funcionamento deste componente é usado o CAPWAP que é responsável por vários aspectos:

- Descoberta e selecção de um controlador por parte de um ponto de acesso.
- *Download* do *firmware* do ponto de acesso pelo controlador.
- Capacidades de negociação entre o controlador e o ponto de acesso.
- Autenticação Mútua entre o controlador e o ponto de acesso.
- Troca de estatísticas de configuração e estado entre o controlador e o ponto de acesso.
- Mapeamento de QoS (*Quality of Service*) por todos os segmentos de rede (com ou sem fios).

Para além disto, apesar de o protocolo CAPWAP não definir explicitamente todos os detalhes, o controlador realiza funções tais como *Radio Resource Management (RRM)* e detecção de pontos de acesso ilegais, bem como gestão de mobilidade. [Sridhar, 2006]

2.4 *Simple Network Management Protocol*

Nesta secção é estudado o protocolo SNMP com algum detalhe. Inicialmente são feitas considerações gerais do protocolo e é analisada a sua arquitectura. Depois são apresentadas as primitivas que permitem a interacção entre agente e gestor. Um aspeto importante a abordar é o que são as MIBs SNMP e como funcionam, também isso é estudado nesta secção. Por fim são apresentadas as versões existentes do protocolo e as suas respectivas diferenças.

2.4.1 Arquitectura

A arquitectura SNMP é utilizada para gestão de dispositivos e serviços em redes IP. Serve para monitorizar e configurar a rede e o equipamento presente, procurando alertar

para condições que requeiram atenção operacional ou reconfigurando os recursos geridos automaticamente. A sua arquitectura é bastante simples, baseada num modelo agente/gestor, sendo composta por três elementos. Estes elementos são o agente, o gestor (*manager*) e a base de dados de gestão de informação. O agente está normalmente presente nos dispositivos a monitorizar, funcionando como uma interface entre o dispositivo gerido e o gestor. A sua função é implementar a instrumentação suportada pelo dispositivo gerido, disponibilizando ao gestor informação de gestão. Por sua vez, o processo gestor na aplicação de gestão monitoriza e controla os dispositivos, servindo de interface entre o sistema de gestão e administrador da rede. A aplicação de gestão pede informação aos agentes com um determinado intervalo de tempo (processo chamado de *polling*), comparando os valores recebidos com os valores referência. Pode ainda receber os alertas enviados directamente pelo agente, bem como alterar parâmetros de funcionamento e configuração do dispositivo. Apesar de normalmente se usar uma arquitectura centralizada com um só gestor a monitorizar todos os dispositivos, podem existir múltiplos gestores no sistema, cada um com uma área de acção diferente ou para criar redundância no sistema.

Tanto o agente como o gestor, utilizam uma MIB (*Management Information Base*) e um conjunto de primitivas relativamente pequeno para realizar a troca de informação. Estes dois aspectos são explorados em mais detalhe nas próximas secções. O protocolo SNMP está presente na camada aplicacional do modelo TCP/IP, usando UDP como protocolo de transporte. As portas usadas por omissão na comunicação deste protocolo são a 161 para comunicação entre o gestor e o agente (*polling*) e a 162 para comunicação no sentido agente-gestor, no caso de envio de alertas não solicitados para o gestor. Quando o SNMP é utilizado com TLS (*Transport Layer Security*) ou DTLS (*Datagram Transport Layer Security*), o gestor recebe os pedidos na porta 10161 e as notificações são enviadas para a porta 10162. [Mauro and Schmidt, 2005]

A simplicidade deste protocolo potencia uma utilização largamente difundida. Uma das causas desta simplicidade é o reduzido número de primitivas utilizadas. Outra das causas é a utilização de uma ligação de comunicação não orientada à conexão e não supervisionada. É considerado robusto devido à independência entre gestores e agentes, ou seja, se um agente falha o gestor continua a funcionar e vice-versa. Outra das vantagens é a sua flexibilidade, sendo usado em inúmeros dispositivos. Por outro lado é uma tecnologia pouco escalável por causa do modelo centralizado.

2.4.2 Primitivas

Conforme referido na secção anterior, o SNMP disponibiliza um pequeno conjunto de primitivas. Na primeira versão (SNMPv1) do protocolo, são especificadas as cinco primitivas básicas para interacção entre o agente e o gestor, sendo estas *GetRequest, SetRequest, Get-*

NextRequest, Response e *Trap*. Duas primitivas adicionais, *GetBulkRequest* e *InformRequest* foram adicionadas na segunda versão (SNMPv2) e mantidas na terceira versão (SNMPv3). De seguida são apresentados as primitivas:

- *GetRequest*: Primitiva iniciada pelo gestor para obter informação presente no agente, tipicamente o valor de uma variável ou lista de variáveis. As variáveis ou objectos de gestão são identificadas com um OID (*Object Identification*), sendo enviado um par OID/valor com o campo valor vazio. É devolvido um *Response* com os valores actualizados por parte do agente.

- *GetNextRequest*: Esta primitiva também iniciada pelo gestor, é utilizada para descobrir variáveis disponíveis e os seus valores. Devolve um *Response* com o par OID/valor da próxima variável na MIB. A MIB de um agente pode ser percorrida iterativamente com o uso de *GetNextRequest*.

- *SetRequest*: Primitiva enviada no sentido gestor-agente, com a finalidade de alterar o valor de uma variável ou lista de variáveis no agente. É enviado um par OID/valor, em que o campo valor contém o valor a modificar no agente. É devolvido um *Response* com o valor actualizado da variável para efeitos de confirmação de sucesso da operação.

- *Response*: Primitiva enviada pelo agente, devolve o par OID/valor conforme os pedidos efectuados (*GetRequest, SetRequest, GetNextRequest, GetBulkRequest e InformRequest*). A indicação de erros é especificada nos campos *error-status* e *error-index*. Apesar de ser utilizada como resposta aos *gets* e *sets*, esta primitiva era conhecida apenas por *GetResponse* no SNMPv1.

- *Trap*: A funcionalidade desta primitiva é a notificação assíncrona partindo do agente para o gestor. É enviado o valor de *sysUpTime* (tempo de funcionamento), um OID identificador do tipo de *Trap*, e outros valores opcionais. O endereço destino é determinado através de variáveis de configuração do *trap* presentes na MIB. Na segunda versão do protocolo foi substituído por *Notification*.

- *GetBulkRequest*: Esta primitiva surgiu como a alternativa optimizada ao *GetNextRequest*. É enviado pelo gestor em vez de múltiplas iterações de *GetNextRequest*. O agente retorna um *Reponse* com múltiplos pares OID/valor. É possível controlar o comportamento da resposta com os campos de eliminação de repetições e repetições máximas.

- *InformRequest*: Esta primitiva, permite a notificação assíncrona com confirmação de gestor para gestor. É utilizado para permitir hierarquizar gestores.

2.4.3 MIBs

Durante o processo de monitorização, os principais intervenientes, os agentes e os gestores comunicam entre si através de troca de mensagens do estado de um dado dispositivo. Exemplos desta troca de mensagens podem ser, quando um gestor pede ao agente informação relativa ao número de pacotes recebidos numa dada interface de rede do dispositivo. Da mesma forma, um agente pode enviar um alerta para informar o gestor que detectou um erro numa porta específica do dispositivo. Para os dispositivos comunicarem correctamente entre si, tem de haver uma linguagem comum, um modelo de Informação. Deste modo, são definidos os termos para identificar um tipo específico de informação bem como identificadores de uma instância desse tipo de informação. É necessário então organizar toda esta informação relativa à entidade que está a ser gerida. Isso é conseguido através de uma MIB que represente conceitualmente a informação, através da sua abstracção em objectos de gestão.

Uma MIB no entanto, não deve ser confundida com uma base de dados. Uma base de dados guarda informação sobre o mundo real num sistema de ficheiros. A MIB está "ligada"ao mundo real e apenas fornece uma visão sobre ele através de um paradigma denominado de instrumentação. Outro exemplo destas diferenças é a quantidade de dados armazenados. Ao passo que uma base de dados contém grandes volumes de dados sobre um mesmo aspecto, tendo poucas tabelas com muitas entradas cada uma, uma MIB contém muitos tipos diferentes de informação, mas com poucas instâncias de cada um.

Os aspectos físicos ou até lógicos do dispositivo são referidos como recursos reais ou recursos geridos, de forma a serem distinguidos da abstracção de gestão, os objectos geridos. Uma MIB, para além destes objectos específicos, pode também conter informação de como estes objectos se relacionam, por exemplo para o cálculo da percentagem de largura de banda.

São agora apresentados em mais detalhe os constituintes da MIB, os objectos geridos ou objectos SNMP, nomeadamente os seus identificadores, os OID (*Object Identifier*). Os OIDs são estruturados segundo o padrão de uma árvore hierárquica e consistem basicamente numa sequência de números, sendo que cada número representa um nó na árvore. Tomando como exemplo o objecto que representa o tipo de uma interface de rede. O OID deste objecto é 1.3.6.1.2.1.2.2.1.3.

Os tipos de informação de gestão são muito variados, por isso, convém distingui-las em diferentes categorias. Estas categorias são Informação de Estado, Informação de Configuração Física, Informação de Configuração Lógica e Informação Histórica. [Clemm, 2006]

Informação de Estado. É a informação relativa ao estado actual dos recursos físicos e lógicos, juntamente com dados operacionais. Inclui informação que permite verificar o fun-

cionamento correcto do dispositivo, como por exemplo, quais as condições de alerta e a sua gravidade ou há quanto tempo o sistema está ligado. Também inclui informação sobre a performance do dispositivo, como contagem de pacotes e ligações, carga de CPU e utilização de largura de banda e memória. As aplicações de gestão não a podem alterar, apenas adquirir. Esta informação, pela sua natureza, está sujeita a mudanças rápidas e frequentes, por esta razão, as aplicações de gestão tomam a decisão de não a armazenar, mas sim adquiri-la do dispositivo (agente) quando necessário.

Informação de Configuração Física. É a informação sobre como o dispositivo gerido está fisicamente configurado. Inclui informação sobre o tipo de dispositivo, placas e portas disponíveis, números de série e endereços MAC. Tal como a informação de estado, também esta informação "pertence"ao dispositivo, sendo que as aplicações de gestão não a podem alterar, apenas adquirir. No entanto, a informação de configuração física muda com pouca ou mesmo nenhuma frequência. Por esta razão, as aplicações de gestão guardam esta informação em vez de a pedir repetidamente ao agente, melhorando a sua eficiência.

Informação de Configuração Lógica. Esta informação é referente a vários parâmetros de configuração e recursos de configuração lógica do dispositivo como endereços IP, números de telefone ou interfaces lógicas. Ao contrário de outras categorias, esta informação pode ser alterada pelas aplicações de gestão e administradores com autorização apropriada. Assim, as aplicações de gestão guardam esta informação, sabendo que a informação não muda a não ser que os administradores a mudem. A informação de configuração lógica pode ainda ser subdividida em Informação de Configuração de Inicialização e Informação de Configuração Transitória. A Informação de Configuração de Inicialização deve permanecer inalterada para que o dispositivo possa manter a informação depois de reiniciar. Por outro lado, a Informação de Configuração Transitória não permanece no dispositivo e pode ser perdida ou voltar para os valores por omissão se o dispositivo precisar de ser reiniciado.

Informação Histórica. Esta informação inclui sumários de performance do estado como contagem de pacotes em intervalos de 15 minutos nas últimas 24 horas. Inclui ainda registos de vários tipos de eventos como, por exemplo, tentativas de acesso remoto. A Informação Histórica diferencia-se dos outros tipos de informação porque não mostra a visão actual dos recursos. Com isto, não deve ser armazenada na MIB mas sim armazenada num ficheiro de *log* no dispositivo. A sua finalidade é retirar carga sobre o sistema de gestão, tendo este apenas que adquirir os dados em bruto, em vez de adquirir incrementalmente em intervalos frequentes.

2.4.4 Versões

O SNMPv1 foi o primeiro protocolo introduzido e ainda é amplamente usado. Esta versão implementa as operações *GetRequest, SetRequest, GetNextRequest, GetResponse, e Trap*. A segurança nesta versão é baseada em "*community strings*"que são transmitidas em cada mensagem, funcionando como uma palavra-passe. Esta palavra não é encriptada e a segurança fornecida é, obviamente, muito fraca.

O SNMPv2 introduziu a capacidade de transmitir definições de MIBS SMIv2. Além disso, como já referido anteriormente, introduziu também as operações *GetBulkRequest, InformRequest* e *Notification*, bem como respostas de erro melhoradas. Para além disto, foram definidos formatos de PDU de forma que a mesma estrutura de PDU pudesse ser usada para qualquer operação, facilitando o processamento das mensagens SNMP, além de poder ser incluída informação para mecanismos de segurança, nomeadamente para implementação de autenticação e confidencialidade. No entanto, não foi estabelecida uma obrigatoriedade de utilização dos mecanismos de segurança, daí ter surgido a designação de SNMPv2c, por ser possível continuar a usar *community strings* da versão 1.

Finalmente, o SNMPv3 é a mais recente introdução do protocolo, tendo como principais preocupações as melhorias no campo da segurança tornando obrigatório o uso dos mecanismos de segurança introduzidos no SNMPv2, isto é, encriptação de mensagens de gestão e autenticação dos utilizadores, permitindo determinar que foi um gestor autorizado que gerou a mensagem e que esta não foi adulterada. Inclui também uma arquitectura padrão e modularizada para implementações de um agente SNMP. Com tudo isto, o SNMPv3 tornou-se num protocolo mais poderoso e mais complexo do que a versão original, permitindo a implementação de ferramentas de gestão mais eficientes.

Os sistemas de comunicações continuam a crescer em número e a evoluir rapidamente. Como se pode verificar pelo trabalho realizado neste documento, a área de gestão de redes tem apresentado desafios cada vez mais importantes e mais difíceis de resolver. Para se conseguir um sistema de gestão de redes eficaz, com o mínimo ou nenhumas falhas, seguro e fiável, é necessário ter em conta inúmeros factores. São exigidos conhecimentos de várias áreas, nomeadamente, comunicações, análise de redes, bases de dados, sistemas distribuídos, inteligência artificial e factores humanos e económicos. [Callon et al., 2009]. Neste sentido, ainda existe muito a fazer neste trabalho, mais pesquisa e amadurecimento de conceitos, e reflexão sobre o melhor método a usar para conseguir um sistema de gestão de rede com capacidades de acesso remoto e automatização.

2.5 Network Configuration Protocol

Nesta secção é analisado o protocolo NETCONF, começando por uma introdução da história do protocolo, seguindo-se de uma abordagem mais técnica sobre o funcionamento, operações disponíveis e características que distinguem este protocolo do SNMP. Para finalizar, é efectuado um levantamento do trabalho mais recente dentro da área relacionada com o protocolo NETCONF, bem como algumas propostas em estado experimental.

Ao longo deste capítulo, os termos cliente ou servidor, gestor ou agente são usados para descrever os elementos de um sistema de gestão de rede. Desta forma, em SSH, o cliente abre a ligação e o servidor escuta passivamente a ligação. Em NETCONF, o gestor envia comandos remotos (RPC) e o agente responde a esses comandos. Assim, o cliente é equivalente ao gestor e o servidor é equivalente ao agente. Também a palavra em inglês *capability*, referente a uma funcionalidade do NETCONF, é traduzida e utilizada ao longo do documento como aptidão.

De certa forma, pode-se dizer que o NETCONF é uma tecnologia concorrente ao SNMP para gestão de configuração. A tecnologia está ainda em fase de desenvolvimento não consegue substituir completamente as funcionalidades do SNMP nos equipamentos de rede utilizados na actualidade. Isto é justificado na medida em que, desde o momento da sua concepção, o NETCONF foi construído para ser um protocolo de instalação, manipulação e remoção de configurações em equipamentos de rede de forma automatizada, algo que na altura o SNMP não fazia de maneira eficaz. Como já evidenciado, o NETCONF é um protocolo de configuração de redes, sendo baseado em XML (*eXtensible Markup Language*). Esta linguagem é utilizada pela sua facilidade de representar e modelar a estrutura de dados das configurações. Também como argumento a favor do uso de XML, há o facto de ser uma tecnologia usada amplamente, tendo à disposição um grande número de ferramentas para o desenvolvimento. Como vantagem subsequente os fabricantes de equipamento de rede podem criar o seu próprio formato, visto que o XML é um padrão universal.

Nos finais dos anos 80 o IETF criou o SNMP, tendo sido durante muito tempo o protocolo de gestão de rede adoptado por muitas empresas. Nos finais dos anos 90, começou-se a reparar que o SNMP afinal não estava a ser usado para configurar equipamento mas apenas para monitorização da rede. Em 2002, o *Internet Architecture Board* e membros da comunidade de gestão de redes do IETF reuniram-se para discutir a situação. Como resultado desse trabalho documentado no RFC 3535[1], verificou-se que as operadoras usavam principalmente CLI (Command Line Interfaces) para configurar o seu equipamento, cujas características agradavam muito às operadoras, nomeadamente o facto de ser baseado em texto, ao contrario da utilização de BER (Basic Encoding Rules) por parte do SNMP. Com isto, a maior

[1] Overview of the 2002 IAB Network Management Workshop

parte dos fabricantes não forneciam a opção ou liberdade de configurar totalmente o equipamento via SNMP. No interesse das operadoras, a criação de scripts para gestão do seu equipamento era muito vantajoso, no entanto constatou-se que o CLI tinha algumas falhas, sendo uma delas a natureza imprevisível do *output*. O seu conteúdo e formatação estava sujeito a mudanças imprevisíveis e falta de homogeneização. Nesta mesma altura, a Juniper Networks abordava esta situação com gestão de rede baseada em XML. O IETF tomou conhecimento deste método e a Juniper Networks partilhou-o com a comunidade de desenvolvimento. Estes acontecimentos levaram à criação de um protocolo, o NETCONF.

No RFC geral do NETCONF mais recente, o RFC 6241[2] são cobertos os aspectos principais do protocolo e algumas considerações de segurança. Um desses aspectos é relacionado com os requisitos do protocolo de transporte, concluindo que o uso de SSH é obrigatório na implementação de um sistema NETCONF. São descritos os modelos de RPC e de configurações. É apresentado em detalhe o funcionamento das operações base (operações de protocolo), bem como o funcionamento de filtragem de sub-árvores nas configurações NETCONF. Finalmente, é feita uma abordagem aprofundada sobre a modularidade do protocolo através do uso de aptidões.

Como actualmente existe a obrigatoriedade do uso de SSH para a camada de transporte em NETCONF, vale a pena fazer referência também ao RFC 6242[3]. Em relação à modelação de dados conseguida através do YANG, existe o RFC 6020[4], de implementação obrigatória, em que é definida a sintaxe e semântica desta linguagem, a forma como o modelo de dados definido num módulo YANG é representado em XML e como as operações NETCONF são usadas para manipular os dados. Também com implementação obrigatória, vem o RFC 6021[5], em que são descritos os tipos de dados a ser usados na linguagem YANG. Existem ainda outros RFC cuja implementação é opcional, que não são abordados neste trabalho.

O NETCONF fornece um conjunto de operações base, usadas sobre uma camada RPC (Remote Procedure Call). Estas operações permitem uma manipulação das configurações baseada em transacção de bases de dados. É uma tecnologia baseada em sessão, implementada sobre SSH (Secure Shell), tendo como alternativas de transporte SOAP (Simple Object Access Protocol), TLS (Transport Layer Security) e BEEP (Blocks Extensible Exchange Protocol). Estas operações são simplesmente transacções de conteúdo (total ou parcial) das configurações representado em XML, sendo relacionado com informação de gestão de rede. Apesar de o XML por si só já ser facilmente interpretado por um humano, surge a necessidade de definir uma linguagem de modelação mais amigável, mais interpretável com

[2]Network Configuration Protocol (NETCONF)
[3]Using the NETCONF Protocol over Secure Shell (SSH)
[4]YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)
[5]Common YANG Data Types

facilidade por um ser humano. O grupo de trabalho NETMOD desenvolveu tal linguagem de modelação, o YANG, onde é definida a semântica dos dados operacionais, dados de configuração, notificações e operações inerentes ao protocolo NETCONF, podendo ainda ser usada para definir o formato das notificações entre equipamento.[Enns et al., 2011]

Uma das características do NETCONF é a sua modularidade. Para além das operações base, é dada a possibilidade de estender as suas funcionalidades através do uso de aptidões. No entanto, estas funcionalidades adicionais podem não estar presentes em todas as implementações, surgindo a necessidade de o equipamento comunicar entre si para tomar conhecimento das funcionalidades suportadas em cada lado durante o estabelecimento de uma sessão entre cliente e servidor. Isto traz algo de vantajoso para os fabricantes, podendo desenvolver aptidões à medida das suas necessidades. Resultados detalhados sobre o estudo do NETCONF são apresentados no Anexo A.

2.6 Estudo Comparativo SNMP VS NETCONF

À medida que a tecnologia do *hardware* evolui os protocolos têm de poder acompanhar essa evolução, visto que vão sendo necessárias novas capacidades ou funcionalidades. No entanto, segundo uma *mailing list* da Cisco isto não acontece em algumas situações. Um desses casos é exemplificado com o uso de arquitecturas *dual-core* no equipamento *Catalyst 4500 series SUP7E*, onde o SNMP não devolve valores de utilização de CPU correctos. Segundo técnicos da Cisco, em Março deste ano, o consenso geral é que os *bugs* existentes do SNMP não vão ser resolvidos e que deve-se usar NETCONF como alternativa. O problema é que torna-se difícil encontrar documentação ou bibliotecas úteis na esfera NETCONF. Conforme já dito, isto deve-se muito ao facto de o NETCONF ainda não estar verdadeiramente completo, e de ainda não ter sido totalmente aceite pelos fabricantes, levando a situações em que as bibliotecas existentes ainda estão em fase de protótipo. Também existe a opinião, apoiada por uns e contestada por outros, que o NETCONF de forma alguma irá substituir o SNMP, visto que foi criado de raiz para manipular configurações, e não monitorizar um dado dispositivo.

Segundo [Ferro, 2013] no blog *etherealmind.com*, a nível conceptual, e como protocolo, o NETCONF é semelhante ao SNMP, isto é, o modelo de dados YANG é semelhante ao SMI. O autor aponta como desvantagem do SNMP, comparativamente ao NETCONF, o facto de o SNMP ser demasiado limitado para ser útil em troca de dados de configuração. Também o facto de o SNMP ser um protocolo *stateless*, torna difícil o controlo de transacções de dados em múltiplos níveis (*multi-stage data transactions*). O SNMP não executa *backups*

nem restauros de elementos, ou seja, quando é feito um *set* de uma variável não é possível desfazer essa acção, tornando operações de escrita em múltiplas variáveis arriscadas e difíceis.

Ainda segundo o autor [Ferro, 2013], é indicada uma das grandes vantagens que o NETCONF tem sobre o SNMP, as aptidões modulares para além das suas funcionalidades base, e a troca de informação do suporte destas. Isto permite que um cliente descubra a versão do servidor e respectivas aptidões suportadas, que facilita *upgrades* dinâmicos do cliente e ser retrocompatível caso um dos dispositivos não suportem a mesma versão ou aptidões. Como os dados do SNMP não são semanticamente encapsulados, o cliente deve conhecer o formato dos dados a serem lidos (*32bit INTEGER, 64-bit INTEGER,* etc). Com isto, torna-se difícil de programar a leitura, visto que é necessário definir explicitamente todos os dados. Se entretanto algo é modificado, a aplicação precisa de detectar a situação não falhar de forma abrupta, no entanto não existem bibliotecas disponíveis para o fazer.

Em relação à comparação YANG *versus* SMI, pode-se apontar as vantagens do YANG como sendo de leitura, interpretação e representação fácil, os modelos de dados de configuração estruturados de forma hierárquica, tipos de dados estruturados, extensibilidade através de mecanismos de acréscimo, suporte a definição de operações (RPCs), capacidade de validação de uma dada configuração, modularidade de dados conseguida com módulos e sub-módulos e versões bem definidas.

Com a evolução do NETCONF e o seu modelo de dados, o YANG, o protocolo tem-se manifestado cada vez mais como um novo padrão na troca de dados em gestão de redes. Segundo a opinião de alguns, o SNMP não funciona conforme exigido, e a nova geração de comunicações em cloud provavelmente vai precisar de algo novo e mais sofisticado. Nos últimos quatro anos o NETCONF amadureceu e está lentamente a ganhar terreno entre a indústria dos fabricantes. De notar que algumas ferramentas da Juniper e Cisco já usam NETCONF e YANG para troca de dados XML, tal como o OpenFlow. Para além de a linguagem XML estar firmemente estabelecida no mundo da programação, todos os RFCs relacionados com YANG são *Standards Track*. Isto indica que existe algum nível de apoio para levar a uma maior aceitação entre o público geral de gestão de redes. Assim, a nível funcional existem semelhanças com SNMP, no entanto com um formato mais adequado à super-estrutura que as linguagens de programação correntes utilizam.

Em Abril de 2012, Calle Moberg da empresa Tail-F Systems, empresa especializada em software para NETCONF, declara que a ONF (Open Networking Foundation) adoptou o NETCONF como sendo obrigatório para configuração de dispositivos que suportem OpenFlow. Comentando sobre este assunto, [Ferro, 2013] afirma que isto é muito importante, visto que irá existir um padrão estabelecido para configurar dispositivos OpenFlow e também assinala o facto de existirem bibliotecas NETCONF escritas em Python e Perl que

foram testadas e comprovadas para configuração de dispositivos. Esta última afirmação vai completamente contra outra opinião referida anteriormente, o que deixa algumas dúvidas talvez a nível de utilização práctica e demonstra que ainda existem opiniões divergentes em relação ao uso das tecnologias em comparação. Mesmo assim, com estes dois factos é possível afirmar que o mercado está no caminho para se desenvolver mais rapidamente e a criação de ferramentas de automação na *cloud* será mais fácil. Assim, com melhores ferramentas para configurar e gerir a rede, eventualmente será possível servir um dos propósitos principais da *cloud* que é automatizar tarefas operacionais. As APIs do SNMP não são ricas e são limitadas para algo mais do que monitorização de performance e disponibilidade. Greg Ferro diz ainda que num futuro próximo, será provável o aparecimento de APIs em XML, suporte para plataformas *cloud* como OpenStack, integração com NETCONF/YANG para gestão de configurações e consequentemente o reconhecimento final em que o SNMP não faz parte do futuro.

Com isto em mente, faz todo o sentido abordar a temática das SDN (Software Defined Networks), já que dispositivos OpenFlow são aplicados em redes SDN. Segundo Alvaro Retana (HP Networking) em [Ferro, 2013], as redes SDN são um passo na direcção certa, no caminho de tornar as redes mais dinâmicas e auto-adaptáveis, tendo em vista a evolução da gestão de redes. Concordando com a afirmação anterior, o autor Greg Ferro é da opinião que a gestão de redes praticada na actualidade é conseguida usando protocolos com APIs muito pobres. Tais protocolos são o SNMP, Telnet/SSH e RMON. Estes protocolos são muito limitados em termos de capacidade, sendo esta falha justificada com o facto de terem sido criados numa altura em que havia grandes restrições relativamente a CPU e memória, facto que não é verificado na actualidade. No entanto, os fabricantes têm demonstrado constantemente falta de vontade para adoptar novos protocolos como NETCONF/YANG. O valor do OpenFlow como parte das SDN é que pode ser combinado com protocolos existentes para proporcionar um impacto significante a nível de configuração e operação do plano de dados.

Segundo o estudo de [Hedstrom et al., 2011], alguns trabalhos anteriores tinham em falta uma verdadeira comparação lado-a-lado da eficiência do SNMP e do NETCONF, relativamente a funções de gestão de configurações. Um grupo de pesquisa [Sun-Mi et al., 2006] examina os melhoramentos da *performance* no protocolo NETCONF, contudo, os testes e caracterizações são baseados num *draft* do IETF nas fases iniciais do protocolo e numa implementação em fase de protótipo. Hoje em dia estão disponíveis implementações de clientes e servidores NETCONF, o que torna os estudos actuais mais credíveis. Outro grupo de pesquisa efectuou uma comparação do SNMP com *Web Services* [Pras et al., 2004] para caracterização da *performance* da gestão de rede. Apesar de ter sido comparada a performance do SNMP com uma tecnologia baseada em XML, não foi feita a comparação com o protocolo NETCONF. O estudo [Hedstrom et al., 2011] preenche essa lacuna. Ainda outro estudo, este empírico, foi efectuado por um grupo de pesquisa [Yu and Ajarmeh, 2010] onde

o SNMP e o NETCONF foram testados comparativamente em ambiente laboratorial usando uma implementação *open source* Yencap do servidor NETCONF. Apesar deste estudo ter feito uma análise comparativa de algumas centenas de *queries* de objectos de gestão com ambos os protocolos, não foi efectuada qualquer comparação de funções de configuração. Neste estudo comparativo entre SNMP e NETCONF [Hedstrom et al., 2011], foca-se a atenção na análise quantitativa da performance de ambos os protocolos em relação a funções de gestão de configurações, na qual foram analisados 100,000 objectos de gestão. Mais especificamente foram implementados um gestor SNMP e um cliente NETCONF em módulos Python para trabalhar em conjunto com as bibliotecas *open source* Net-SNMP e *ncclient*. O servidor NETCONF ConfD que contém a base de dados dos objectos de gestão está disponível comercialmente através da Tail-f Systems. Esse servidor inclui um agente SNMP embebido. Assim, é possível simular um dispositivo de rede que contém informação de configuração que pode ser actualizado por ambos os protocolos.

A aplicação cliente do NETCONF, bem como a aplicação de gestão do SNMP executaram a configuração de 100,000 objectos de gestão em sequência e guardaram o tempo de operação. Também foi incluído um *sniffer* para analisar os pacotes em circulação e efectuar medições e cálculos, tanto temporais como quantitativos. Do lado do SNMP foi utilizada a operação SET que foi optimizada para transportar mais do que um objecto por mensagem. Foi determinado que utilizando valores com *strings* de comprimento de 255 caracteres, um SET PDU (*SET Protocol Data Unit*) conseguiu transportar 36 pedidos de configuração de objectos de gestão. Uma abordagem não optimizada apenas incluiria um pedido de configuração de objecto de gestão por PDU.

Como resultados do estudo, em relação ao tempo de operação, a vantagem estava do lado do SNMP. Contudo, à medida que o número de objectos de gestão excedia os 500, a eficiência do tempo de operação do NETCONF superou ligeiramente o do SNMP, conforme é possível verificar no seguinte gráfico obtido no estudo:

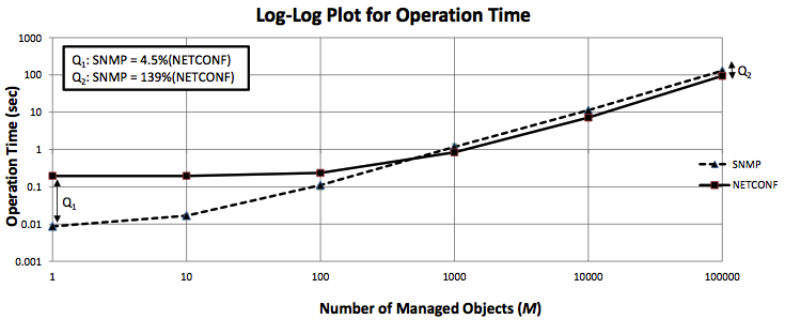


Figura 2.1: *Comparação dos protocolos: Tempo de operação. Imagem retirada de [Hedstrom et al., 2011]*

No que diz respeito a utilização de largura de banda, o protocolo NETCONF demonstrou uma maior utilização de largura de banda por número de pacotes *Ethernet* e *bytes*, devido ao *overhead* exigido por funcionalidades de segurança, orientação à conexão (baseado em sessão) e troca de informação sobre aptidões, funcionalidades estas que não existem no protocolo SNMP. No entanto, estas são necessárias para gerir redes complexas. Os resultados que permitiram chegar a estas conclusões são apresentados de seguida:

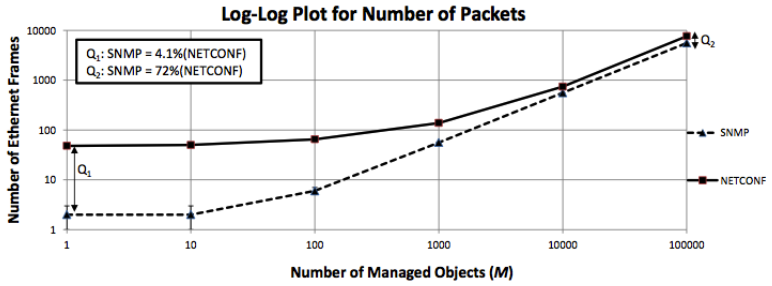


Figura 2.2: *Comparação dos protocolos: Utilização de largura de banda. Imagem retirada de [Hedstrom et al., 2011]*

Por fim, o NETCONF evidencia muito bem a sua eficiência, superior à do SNMP, em relação ao número de transacções efectuadas. O NETCONF foi capaz de configurar 100,000 objectos de gestão numa única transacção, usando os dados de configuração XML como *payload*, enquanto o melhor caso do SNMP é 2779 transacções para o mesmo número de objectos de gestão. Esta vantagem por si só, balanceia a grande utilização de largura de banda por parte do NETCONF. De seguida são apresentados os dados relativos a este aspecto:

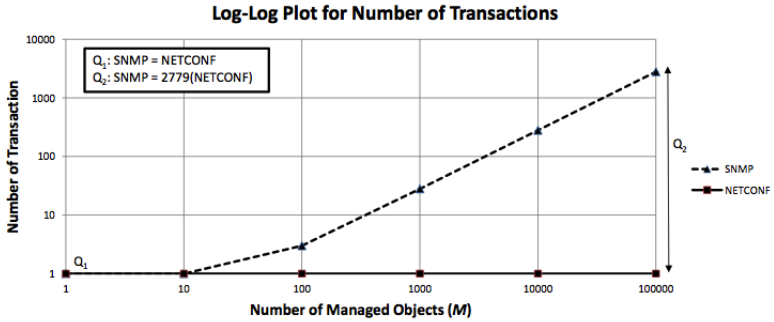


Figura 2.3: *Comparação dos protocolos: Número de transacções. Imagem retirada de [Hedstrom et al., 2011]*

O grupo termina, concluindo que o NETCONF é uma alternativa viável ao SNMP para funções de configuração, sobretudo nas redes complexas de hoje em dia, bem como nas futuras gerações de sistemas *back-office*, fornecendo assim a possibilidade de criar ferramentas mais poderosas para atender às necessidades cada vez mais avançadas das operações de configurações de rede.

Estes resultados são discutíveis, principalmente visto que as conclusões são retiradas de um ambiente laboratorial, e não numa situação real. Isto é bastante evidente relativamente ao tempo de operação. Em alguns casos apenas são efectuadas pequenas alterações nas configurações, dando assim a maior vantagem para o SNMP. No entanto é sempre de louvar trabalhos deste género que trazem mais alguma informação à comparação dos dois protocolos.

O SNMPv3 raramente é usado em operações de rede em provedores de serviço, devido à sua complexidade operacional. Isto leva a que seja criada alguma confusão na realização dos estudos já que torna-se difícil ou trabalhoso fazer estudos comparativos onde um dos lados costuma ter duas versões aceites para operar. É necessário efectuar estudos mais completos, e que abordem todo o tipo de variáveis, funcionalidades e características. Com isto quer se dizer que não basta fazer um estudo isolado sobre uma característica dos protocolos, onde as vantagens apenas se aplicam num determinado aspecto. Talvez esteja na altura de haver uma nova reunião do IAB (Internet Architecture Board), semelhante à de 2002, que gerou resultados no documento RFC3535, para actualizar as comparações dos protocolos de gestão existentes.

Apesar de não se chegar a conclusões evidentes, muito devido à subjectividade da selecção do melhor protocolo, ficam alguns pontos positivos e negativos de ambos os protocolos. Uma das razões para isto é o facto de os dois protocolos terem sido criados em alturas completamente diferentes sendo as tecnologias disponíveis mais ou menos avançadas. Outra das razões é o facto de terem objectivos ligeiramente distintos. Na próxima secção avança-se para o estudo de MIBs de pontos de acesso sem fios da norma IEEE 802.11 de forma a conhecer melhor os objectos SNMP disponibilizados e o que é possível monitorizar num ponto de acesso.

Mesmo assim, no âmbito desta dissertação o SNMP torna-se vantajoso pelo facto de ter uma base de suporte muito superior, bem como permitir monitorização de dispositivos, neste caso de pontos de acesso. Outra razão pela qual opta-se pela preferência do SNMP é a existência de MIBs para gestão de pontos de acesso sem fios. Assim, o SNMP foi o protocolo escolhido para realizar o trabalho pretendido nesta dissertação.

2.7 Estudo de MIBS de Pontos de Acesso IEEE 802.11

Nesta secção será efectuada uma introdução ao estudo das MIBs SNMP usadas em pontos de acesso sem fios com a tecnologia IEEE 802.11. Inicialmente são feitas considerações sobre o uso e propósito de cada MIB analisada e posteriormente no Anexo B será feita uma análise detalhada ao nível das varáveis utilizadas e a sua funcionalidade. Um dos objectivos

desta secção é justificar a selecção dos objectos SNMP a monitorizar, de forma a serem incluídos na aplicação de gestão e monitorização de uma rede sem fios implementada neste trabalho.

A *"IEEE802dot11-MIB"* [Pang, 2002c] é a MIB padrão criada pelo IEEE para gestão e monitorização de entidades 802.11, onde são implementados os objectos de gestão obrigatórios para todos os fabricantes de maneira a haver alguma homogeneização nos aspectos mais básicos da tecnologia. A *"CISCO-DOT11-IF-MIB"* [Pang, 2002b], que apesar de o nome poder induzir em erro e levar a pensar que é uma MIB para interfaces IEEE 802.11 (placas de rede sem fios, é na realidade para interfaces de rede com funcionalidades especiais como as de um ponto de acesso sem fios. Esta MIB, segundo a descrição da *Cisco*, fornece suporte de gestão de redes para interfaces de rádio do tipo IEEE 802.11 (pontos de acesso sem fios). Finalmente a *"CISCO-DOT11-ASSOCIATION-MIB"* [Pang, 2002a] que é uma continuação das MIBs anteriores, mas com alguns aspectos mais focados na gestão do processo de associação, trabalhando também com configurações estatísticas de reencaminhamento de pacotes de dados.

Como estas três MIBs relacionam-se entre si, logicamente, a forma como estão organizadas estruturalmente é igual em qualquer uma delas. Os objectos estão organizados hierarquicamente com quatro grupos principais de atributos, *dot11smt*, *dot11mac*, *dot11res*, *dot11phy*, que fazem correspondência das camadas da pilha OSI. O grupo *dot11smt* é dedicado a todas os atributos relacionados com gestão da estação (*Station Management*), tendo estes como função gerir processos na estação de forma a funcionar correctamente como parte de uma rede IEEE 802.11. O grupo de atributos *dot11mac* encarrega-se de fornecer suporte para o controlo de acesso, criação e verificação de *Frame Check Sequences* (FCSs) e garante a entrega adequada de dados válidos para as camadas superiores. O grupo *dot11res* (*resources*) tem como função indicar em dados legíveis, a informação que identifica a implementação dos objectos de gestão, como por exemplo o valor do identificador organizacional, o nome do fabricante, nome de modelo, entre outros. Por fim, o grupo *dot11phy* fornece o apoio necessário para informação operacional PHY, que pode variar de PHY para PHY e de STA para STA, a ser comunicada para as camadas superiores.

No Anexo B são abordadas cada uma destas MIBs num maior nível de detalhe. O procedimento consiste principalmente na análise dos agrupamentos de objectos (*object-groups*), presentes no final de cada MIB. Inicialmente, é apresentado o grupo de objectos, seguidos da sua função ou finalidade que têm em comum. Depois disto, são apresentados os objectos de maior relevância, seguidos de uma breve descrição. Estes agrupamentos são usados para descrever relações entre diferentes objectos presentes numa MIB, de forma a que todos os objectos do grupo sejam implementados em conjunto. De notar que, apesar de não ser regra geral, alguns agrupamentos de objectos não correspondem à organização da MIB em si, visto

que um objecto de uma tabela pode estar incluído em mais do que um agrupamento.

Capítulo 3

Aplicação de Gestão - Resolução do Problema de Nodos Escondidos

Este capítulo irá tratar da problemática mais específica dos nodos escondidos em redes sem fios e como a utilização de tecnologias normalizadas baseadas em SNMP podem ajudar a resolver ou atenuar este problema. Assim, será feita a apresentação e descrição do problema dos nodos escondidos e quais as suas consequências na rede, seguido da apresentação de estudos da área e propostas teóricas já existentes para o resolver. Será depois explicado em detalhe como irá ser feita a abordagem e estratégia escolhida para atacar o problema, algoritmos utilizados, bem como os objectos SNMP que irão ser utilizados.

3.1 O Problema dos Nodos Escondidos

Nas redes sem fios, existe um problema relevante que afecta o rendimento e eficiência do seu funcionamento, conhecido por *hidden node* ou nodo escondido. Este trabalho irá focar-se na resolução de problemas de nodos escondidos em redes sem fios no modo de infra-estrutura. Convém salientar este facto, já que em redes ad-hoc a abordagem é ligeiramente diferente, visto não haver um ponto de acesso centralizado.

Numa situação de nodos escondidos um determinado nodo não consegue *ver* ou saber da existência de outro nodo na rede. Isto surge, por exemplo, quando um nodo tem alcance suficiente para comunicar com o ponto de acesso, mas não consegue *ver* outros nodos que não estejam no seu alcance. Isto leva a que haja colisões e perdas na transmissão de dados. Para perceber melhor esta situação, a Figura 3.1 ilustra o problema de forma relativamente simples.

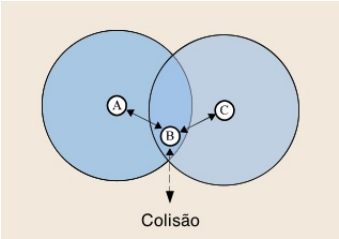


Figura 3.1: *Nodos escondidos e colisão. Imagem adaptada do artigo [Boroumand et al., 2012]*

A Figura 3.1 ilustra o cenário de nodos escondidos, representando o caso mais simples com os nodos A, B e C. O nodo B será o ponto de acesso sem fios da rede. Cada nodo pode receber pacotes com a condição de estarem dentro do alcance de transmissão. Então, nesta situação o nodo A consegue ver o nodo B mas não o nodo C visto que não estão ao alcance um do outro. Assim, o nodo B consegue receber pacotes de ambos os nodos A e C, no entanto haverá uma colisão no nodo B se ambos enviarem pacotes simultaneamente, que resultará na falha de recepção de quaisquer pacotes. Isto leva a que os nodos fonte reenviem repetidamente os pacotes. Pode ver-se que esta não é uma situação desejada, pois causa a diminuição do rendimento da rede e que por sua vez leva a outros problemas tais como, baixa eficiência, aumento de consumo de potência, atrasos na transmissão e deterioração de QoS. Logicamente, esta situação piora com o aumento do número de nodos. [Boroumand et al., 2012]

Adicionalmente, existe outra variante deste tipo de problema conhecido por nodos expostos. Nesta situação, dois nodos querem transmitir para o mesmo receptor, mas um deles conclui incorrectamente que a sua transmissão irá interferir com a transmissão do outro nodo e decide não transmitir. Este problema não será abordado neste trabalho.

Para evitar colisões de pacotes, a norma IEEE 802.11 propõe o DCF (*Distributed Coordination Function*) como método de controlo para acesso ao meio. O comportamento do DCF é baseado em dois mecanismos de percepção do meio (*carrier sense*). O primeiro é a implementação de percepção de meio físico, o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). O segundo é a implementação de percepção de meio virtual, oatravés do mecanismo *Request To Send / Clear To Send* (RTS/CTS). Quando o problema de nodos escondidos ocorre mais frequentemente, a probabilidade de o mecanismo CSMA/CA falhar aumenta, o que acaba por levar à degradação de eficiência significante. É nestas situações de falha do CSMA/CA que o mecanismo RTS/CTS deverá ser utilizado para atenuar o problema de nodos escondidos.

O mecanismo RTS/CTS funciona da seguinte forma: antes de o transmissor enviar os dados, é enviado o pacote de controlo RTS para pedir autorização para enviar os dados; como resposta ao RTS, o receptor envia para o transmissor o pacote de controlo CTS, indicando que está disponível para receber os dados, se este for o caso. Estes pacotes de controlo incluem

informação sobre quanto tempo o canal irá estar ocupado para efectuar a troca de dados. Assim, nodos da mesma rede podem adiar a sua transmissão, ficando à escuta desses pacotes de controlo. Para ilustrar melhor este mecanismo é apresentada a Figura 3.2.

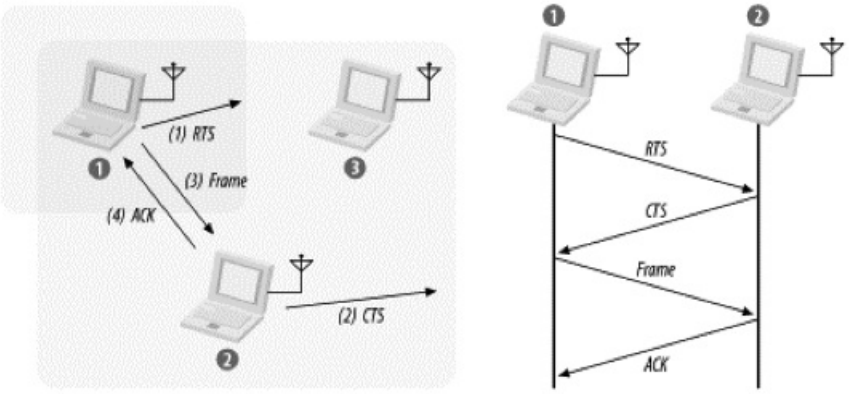


Figura 3.2: *Mecanismo RTS/CTS. Imagem retirada de [Gast, 2002]*

Na figura 3.2, o nodo 1 quer enviar dados e inicia o processo enviando um pacote RTS. Para além deste pacote requisitar a reserva do meio para transmissão, *silencia* outras estações que consigam escutar o pacote. Se o nodo destino receber o RTS irá responder com o pacote CTS. Os pacotes CTS também *silenciam* os nodos vizinhos. Uma vez terminada a troca de pacotes RTS/CTS, o nodo 1 pode transmitir dados sem se ter de preocupar com interferâncias com quaisquer nodos escondidos. [Gast, 2002]

No entanto, o mecanismo RTS/CTS adiciona *overhead* nas transmissões, que pode levar à degradação da eficiência e rendimento da rede quando não existem nodos escondidos. De forma evitar esta degradação desnecessária deve reduzir-se as trocas RTS/CTS e a sua prévia activação. No entanto, a norma IEEE 802.11 define que o uso de RTS/CTS é decidido tendo em conta o tamanho do pacote a transmitir, nomeadamente, quando este é maior do que um valor limite definido como *RTS Threshold*. Assim, pode haver a situação de não existirem nodos escondidos mas ao ser transmitido um pacote maior do que o valor *RTS Threshold*, o RTS/CTS ser activado sem necessidade, introduzindo *overhead* na transmissão. Com isto em mente, decidiu-se desenvolver uma ferramenta de detecção automática de nodos escondidos que possa ser utilizado de forma a poder activar ou desactivar o mecanismo RTS/CTS correctamente, ou seja, na presença ou suspeita de presença de nodos escondidos. Mais ainda, a ferramenta a desenvolver seria uma aplicação de gestão com tecnologia SNMP.

3.2 Estudos da Área e Soluções Propostas

Tal como referido no final da secção anterior, para resolver o problema dos nodos escondidos de forma mais eficaz e porventura elegante, é necessário detectar a existência desse tipo de nodos na rede, activando o RTS/CTS apenas quando estritamente necessário. Para isso foi efectuada uma pesquisa de mecanismos de detecção de nodos escondidos e formas de tornar a utilização do mecanismo RTS/CTS mais eficiente.

Uma das soluções propostas por [Choi, 2008] é um algoritmo de aglomeração de terminais (nodos). Esse algoritmo dividiria os terminais de um BSS (*Basic Service Set*) em agrupamentos, de forma a que os próprios terminais possam detectar transmissões dentro do mesmo agrupamento, fazendo assim com que não exista o problema de nodos escondidos dentro de um dado agrupamento. Quando um ou mais agrupamentos são derivados pelo algoritmo proposto, um período de contenção CP (*Contention Period*) é dividido em n sub-períodos SP (*Sub-Periods*), sendo n o número de agrupamentos existente. Cada um destes sub-períodos é atribuído a um agrupamento de maneira a não ficarem sobrepostos, e assim os terminais dentro de cada agrupamento podem competir pelo acesso ao canal, com a redução da probabilidade de ocorrência de nodos escondidos dentro desse agrupamento. É também proposto um algoritmo de *fairness* para dividir um CP nos mais pequenos SPs.

Outra solução proposta para a redução de trocas RTS/CTS desnecessárias é um método de RTS/CTS adaptativo *On/Off* presente no trabalho de [Shigeyasu et al., 2011]. Este mecanismo decide o uso do RTS/CTS mediante o número de nodos escondidos em redor do receptor da transmissão. O mecanismo funciona da seguinte forma: se o número de nodos escondidos ultrapassa um dado limite, é activado o RTS/CTS antes da transmissão; se esta situação não se verificar, a transmissão é efectuada normalmente sem o RTS/CTS reduzindo assim o *overhead* da mesma. Para conseguir isto, cada nodo tem dois tipos de listas, uma lista de nodos vizinhos e uma lista de nodos escondidos. Cada terminal preenche ambas as listas escutando por pacotes RTS/CTS ou DATA. Depois disto, cada nodo na rede encaminha a sua lista de nodos escondidos para outros nodos de maneira a esses poderem saber da existência de nodos escondidos na sua vizinhança.

Algo mais radical é proposto por [Yang and Ma, 2008], isto é, um novo protocolo de controlo de acesso ao meio baseado em contenção, esse protocolo é o CR-MAC (*Channel Reservation MAC*). Este protocolo tira partido da característica de escuta de um canal sem fios partilhado de forma a trocar informação de reserva do canal com pouca introdução de *overhead*. Cada nodo pode reservar o canal para o próximo pacote em espera na fila de transmissão. Teoricamente o protocolo CR-MAC consegue atingir um melhor rendimento de transmissão que o RTS/CTS em tráfego saturado. Este protocolo também reduz o número de colisões de pacotes, poupando assim energia para retransmissões.

A proposta dos autores [Kim and Choi, 2012] aborda um facto muito importante que é saber diferenciar as causas de perdas de pacotes. Apesar deste trabalho apenas contemplar as redes 802.11n propõe uma solução interessante baseada na utilização de agregação de tramas, ACK em blocos, e adaptação de ligação rápida FLA (*Fast Link Adaptation*). Usa assim um novo mecanismo de detecção de nodos escondidos que toma em atenção as três causas de perda de pacotes que são colisões, nodos escondidos e problemas no canal. O mecanismo detecta nodos escondidos apoiando-se em estatísticas da camada MAC e blocos ACK recebidos, determinando se deverá utilizar RTS/CTS ou não. A detecção de nodos escondidos depende da capacidade do transmissor diferenciar as suas perdas de pacotes de acordo com a causa. Se o transmissão conseguir identificar a causa dessas perdas, pode então seleccionar medidas apropriadas para a próxima transmissão. As principais causas são as seguintes. Colisões (interferência síncrona) que são sinais de interferência causados por transmissões de outros nodos dentro do alcance de percepção do meio do transmissor. Nodos escondidos (interferência assíncrona) que são sinais de interferência causados por nodos fora do alcance de percepção do meio do transmissor. Finalmente problemas do canal tais como ruído, desvanecimento, e distorção que resultam em erros do canal.

Já a proposta de [Mjidi et al., 2008] aproxima-se bastante da estratégia escolhida para esta dissertação, defende que a utilização de forma eficiente da activação ou desactivação do mecanismo RTS/CTS é muito importante para maximizar o rendimento de uma rede sem fios, sendo sugerido um esquema dinâmico de utilização do RTS/CTS. Os algoritmo que os autores propõem ajusta dinamicamente o parâmetro *RTS Threshold* de acordo com a taxa de entrega de pacotes que é um bom indicador de tráfego na rede. Se a taxa de entrega é inferior a um determinado limite, é activado o mecanismo RTS/CTS de forma a evitar colisões. Caso contrário, os pacotes são enviados utilizando o esquema normal, o CSMA/CA. É defendido que as principais vantagens desta abordagem são a simplicidade e alta taxa de exactidão. Para além disso, apenas depende da taxa de sucesso de entrega de pacotes, independentemente do tamanho da rede. O ajustamento dinâmico do parâmetro *RTS Threshold* garante o equilíbrio entre elevado número de colisões e melhor utilização do canal.

Como se pode ver, existem inúmeras abordagens, no entanto algumas das apresentadas implicam mudanças profundas ao nível MAC, ou apresentam soluções com implementação impossível de realizar com tecnologia SNMP, que é o objectivo principal desta dissertação. Por fim, é apresentada outra solução [Chen and Vukovic, 2007], que é a que mais se enquadra dentro dos objectivos tecnológicos deste trabalho. A solução a seguir neste trabalho propõe um mecanismo de detecção e mitigação de nodos escondidos denominada *RTS on Demand*. É um mecanismo que utiliza dinamicamente o RTS/CTS tendo em conta a situação de interferência, monitorizando vários parâmetros tais como trama de resposta a um nodo desconhecido, tentativas de retransmissão e taxa de colisões. Esses parâmetros serão

monitorizados através de SNMP, em objectos presentes na MIB do ponto de acesso. Na secção seguinte este mecanismo é explorado em maior detalhe, sendo também apresentado o algoritmo a implementar.

3.3 Algoritmo

Tal como referido anteriormente, procurou-se uma solução para resolver ou atenuar o problema dos nodos escondidos e em que a tecnologia base suportada fosse o SNMP. Para isso seria necessário poder detectar a existência de nodos escondidos, bem como actuar de forma mais eficiente do que simplesmente activar o RTS/CTS quando o tamanho do pacote transmitido passar o limite definido no objecto *dot11RTSThreshold* da MIB *IEEE802dot11-MIB*, evitando assim a introdução de overhead desnecessariamente.

Nesse sentido, será apresentado de seguida um resumo da estratégia a seguir, estratégia essa proposta no artigo "*An RTS-on-demand Mechanism to Overcome Self-interference in an 802.11 System*"[Chen and Vukovic, 2007] que sugere um mecanismo dinâmico da utilização do RTS/CTS. Isto é conseguido através da monitorização de alguns parâmetros tais como taxa de colisão causada por falha da transmissão, por exemplo quando ocorre uma falha de descodificação de uma trama, tentativas de retransmissão e, por fim, a recepção de tramas ACK ou CTS sem que outro dispositivo tenha detectado tramas DATA ou RTS. Esta solução inclui três sub-algoritmos todos eles focados na optimização da utilização de RTS/CTS baseado na detecção de colisões. De notar que ao longo desta secção surgem variáveis tais como $N_{RoD-R-ac}$, por exemplo, em que $_{ac}$ significa activar e $_{de}$ significa desactivar. De salientar ainda que ao longo da apresentação dos algoritmos, a referência a nodo será sempre o ponto de acesso, já que vai ser com as MIBs presentes no dispositivo que será recolhida a informação que define os critérios de activação do RTS/CTS.

O primeiro algoritmo é denominado *RTS on Demand using Acknowledgement* (RoD-A) e baseia-se na recepção de qualquer trama de resposta tal como CTS ou ACK. Se essas tramas de resposta estão endereçadas para um nodo desconhecido enquanto o nodo está no estado de *backoff*, então o nodo deverá começar a usar RTS/CTS. Neste caso o nodo será o ponto de acesso sem fios. Usando o cenário tradicional (figura 3.1) de três nodos A, B e C em que B está ao alcance de A e C, mas A e C não estão ao alcance um do outro (escondidos), pode-se descrever o algoritmo da seguinte forma. Se o nodo C recebe uma trama CTS ou ACK endereçada do nodo A para o nodo B sem escutar tramas RTS ou DATA enviadas pelo nodo A, o nodo C pode assumir que o nodo A está escondido para si e deverá usar RTS/CTS para reduzir as probabilidades de estar envolvido numa colisão com o nodo escondido, o nodo A.

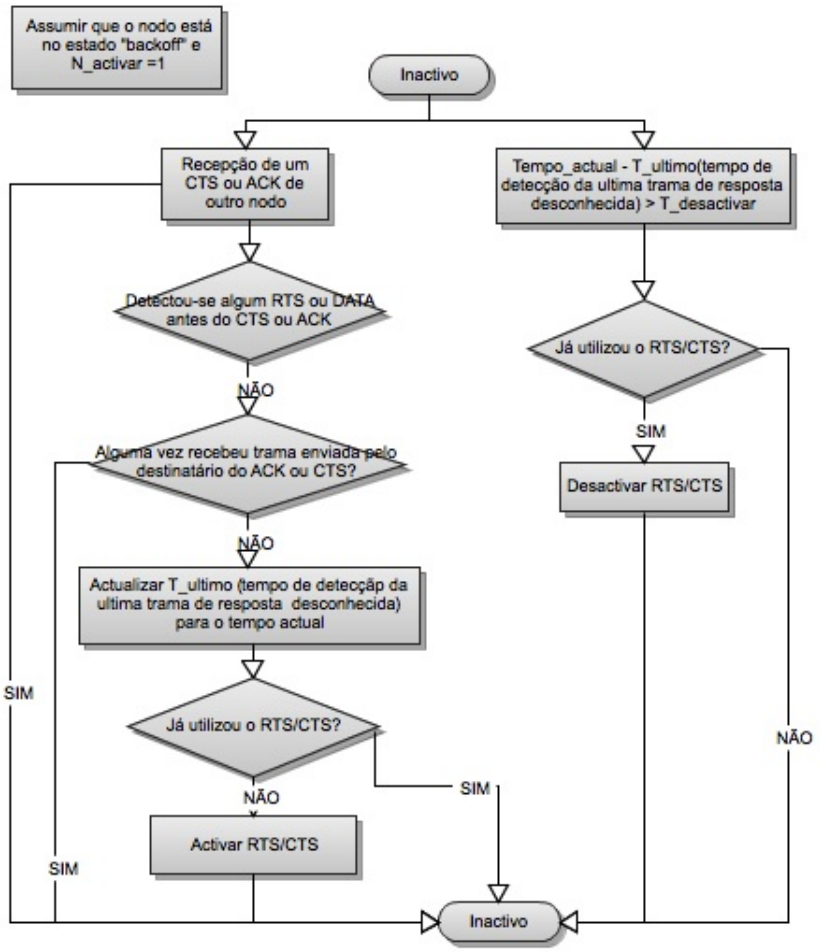


Figura 3.3: *Diagrama de fluxo do RoD-A. Imagem adaptada do artigo [Chen and Vukovic, 2007]*

É importante notar que um dado nodo apenas deverá contar eventos de recepção de tramas CTS ou ACK com destino desconhecido com a condição de estar no estado *backoff*. Isto porque quando um nodo não tem dados para transmitir, não existe necessidade de monitorizar a recepção de tramas ACK e CTS, já que a situação de nodo escondido pode mudar rapidamente devido a mobilidade dos nodos ou destes serem ligados ou desligados. Com o algoritmo RoD-A um nodo (o ponto de acesso) define um limite $N_{activar}$ para activar o RTS/CTS depois de detectar $N_{activar}$ tramas CTS ou ACK desconhecidas consecutivamente. O nodo deverá então definir também um temporizador $T_{desactivar}$ de forma a desactivar a utilização do RTS/CTS caso não sejam detectadas mais tramas CTS ou ACK desconhecidas dentro do intervalo de tempo $T_{desactivar}$.

Aplicando o algoritmo RoD-A com limites apropriados, um nodo pode detectar eficientemente um nodo escondido e activar dinamicamente o RTS/CTS de forma a reduzir o impacto causado pelo nodo escondido quando há uma transmissão em curso. O algoritmo RoD-A é um mecanismo pró-activo, na medida em que um nodo tenta pró-activamente detectar um nodo escondido mesmo quando esse nodo escondido não tenha sido a causa da diminuição de rendimento na rede.

O segundo algoritmo de nome *RTS on Demand using Retransmission* (RoD-R), baseia-se na informação recolhida através da monitorização das tentativas de retransmissão de um nodo (também neste caso o ponto de acesso). Neste algoritmo, o RTS/CTS é activado quando um nodo (ponto de acesso) verifica que a transmissão de cada pacote sofreu mais de $N_{RoD-R-ac}$ tentativas de retransmissão durante os últimos $N_{tentativa}$ pacotes consecutivos, ou durante o último período $T_{activar}$. De forma a melhor verificar a situação de nodos escondidos, um nodo poderá considerar também a carga do sistema, como por exemplo carga de CPU ou a taxa de ocupação do canal. Por exemplo, o nodo apenas irá utilizar o RoD-R quando a carga do sistema estiver abaixo de um determinado nível $L_{activar}$. Isto faz todo o sentido visto que se a actividade do sistema for relativamente baixa, o número excessivo de retransmissões será muito provavelmente causado por nodos escondidos em vez de competição excessiva pelo meio e consequente contenção.

Um nodo que esteja a utilizar o RoD-R pode também desactivar o RTS/CTS se se verificarem as condições seguintes:

- Se a carga do sistema estiver acima do nível limite $L_{desactivar}$.
- Se o número de tentativas de retransmissão de cada pacote forem menores que N_{RoD-R} durante os últimos $N_{tentativa-de}$ pacotes consecutivos ou durante o último período $T_{desactivar}$.

O algoritmo RoD-R é um mecanismo reactivo, visto que é activado quando os problemas causados pelos nodos escondidos já ocorreram. No artigo é ainda mencionado que este algoritmo deveria ser executado mediante informação recolhida através da monitorização das tentativas de retransmissão nos nodos. Como nodos interpreta-se os pontos de acesso e os clientes. Como o RTS/CTS é um mecanismo opcional na norma IEEE 802.11, e o valor de *RTS Threshold* não é especificado na norma, esse valor tem de ser gerido separadamente em cada nodo.[Mjidi et al., 2005] Contudo, não é possível fazer monitorização SNMP nos clientes. Para isso seria necessária a implementação de um agente SNMP com suporte da MIB *IEEE802dot11* e provavelmente algumas alterações a controladores das placas de rede dos clientes. Por isto, o programa a implementar irá trabalhar apenas com pontos de acesso. Na secção de implementação este problema será novamente abordado, com algumas explicações mais detalhadas. De seguida é apresentado o diagrama de fluxo deste algoritmo.

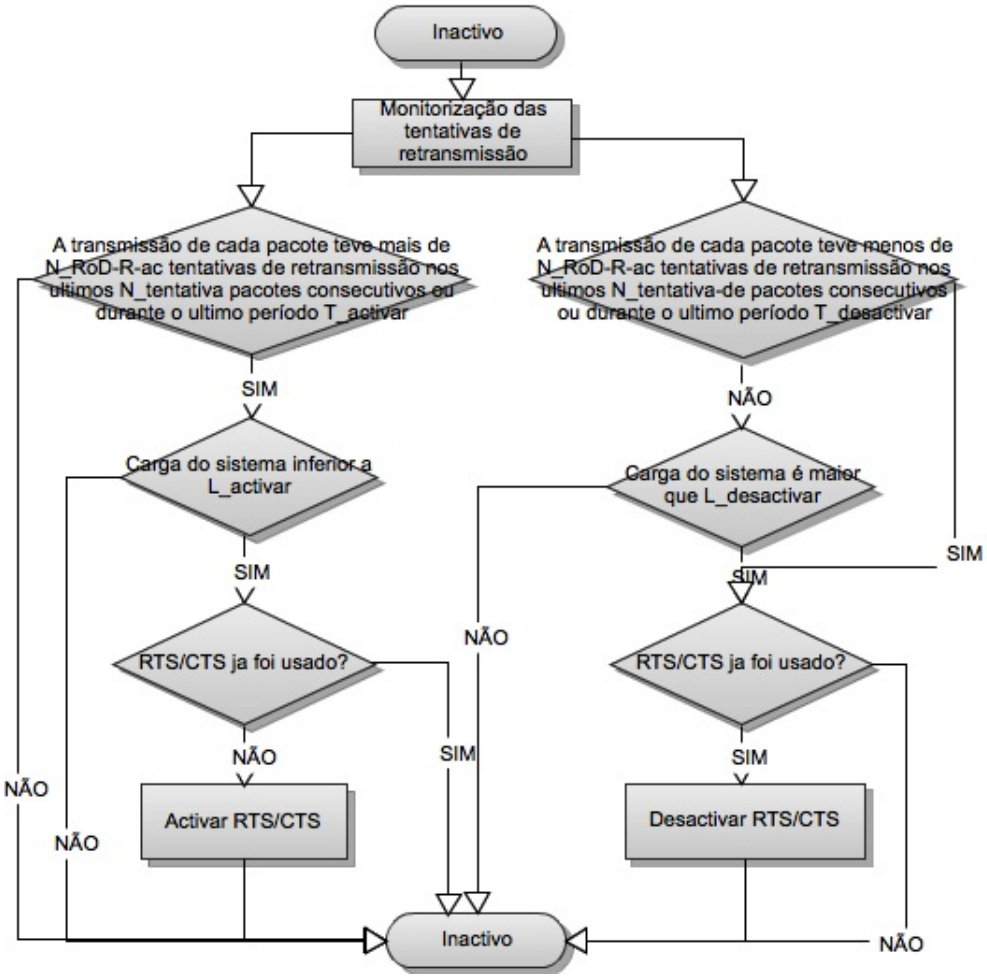


Figura 3.4: *Diagrama de fluxo do RoD-R. Imagem adaptada do artigo [Chen and Vukovic, 2007]*

O terceiro algoritmo proposto no artigo é o *RTS on demand using collision rate* (RoD-C). Ao contrário dos dois algoritmos anteriores, onde um dispositivo 802.11 toma a decisão de usar o RTS/CTS baseado na informação recolhida por si próprio (ponto de acesso), o RoD-C depende da informação dada pelo receptor pretendido numa dada transmissão e trocas de tais informações na rede. Tal como sugerido pelo nome, a informação sobre colisões recolhidas em cada nodo é essencial para o algoritmo RoD-C. Cada nodo no sistema irá monitorizar o estado de colisões e partilhar essa informação com os seus vizinhos através de difusão de uma forma periódica. Quando um nodo começa a transmissão para outro nodo, irá determinar a utilização do RTS/CTS dependendo da informação de colisões recebida por esse nodo. Por exemplo, quando a taxa de colisão monitorizada no receptor pretendido exceder um certo limite C_{RoD-ac} de 30%, o nodo deverá activar o RTS/CTS para esse nodo.

Tal como os anteriores algoritmos, um nodo que utilize RoD-C, poderá desactivar o RTS/CTS na transmissão para outro nodo quando a taxa de colisão verificada for inferior a um certo limite C_{RoD-de}. A taxa de colisões é definida pela percentagem de colisões ocorridas na recepção de pacotes durante um dado período. Se não houver actividade no canal durante o período de monitorização, a taxa de colisão será definida como zero. Um nodo fará a actuali-

zação da sua taxa de colisão a cada período de monitorização. De notar que é relativamente difícil detectar colisões com precisão através de monitorização de tramas não verificadas, porque poderão ter sido causadas por erros no canal ou colisão.

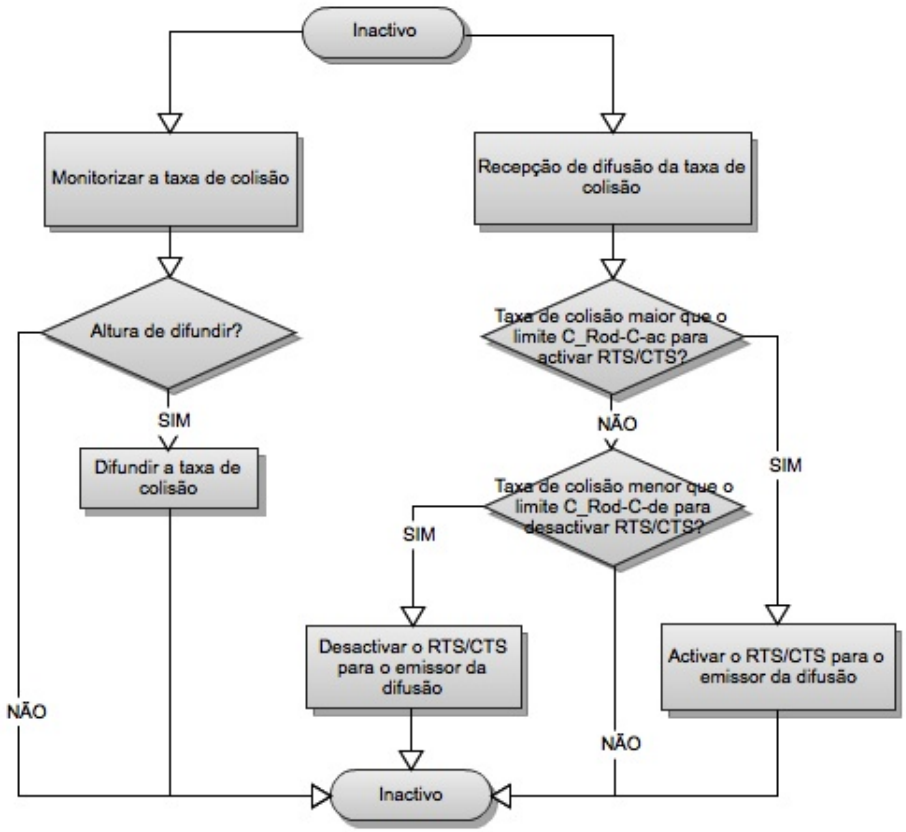


Figura 3.5: *Diagrama de fluxo do RoD-C. Imagem adaptada do artigo [Chen and Vukovic, 2007]*

Para além de um mecanismo para detecção de colisões, de forma a estimar as taxas de colisões, será também necessário um mecanismo de difusão de forma a um nodo propagar a informação de colisões recolhida no detector de colisões. Um dado nodo irá difundir periodicamente (por exemplo $T_{difusao}$) a informação de colisão utilizando uma trama de controlo ou trama de gestão. Essa trama poderá ser uma nova trama ou uma trama reutilizada para transportar a informação da taxa de colisão.

Conforme sugerido pelos autores do artigo, um nodo poderá optar por utilizar um dos três algoritmos apresentados ou até uma combinação dos mesmos, dependendo da disponibilidade da informação. Por exemplo, se o detector de colisões não está disponível, e a maior parte dos nodos não suportam o RoD-C, então o nodo pode optar por utilizar o RoD-A e o RoD-R. Como estes dois algoritmos são complementares, convém definir muito bem os critérios para os limites requeridos. Uma solução simples passaria por usar soma lógica simples onde um conjunto de limites inclui os critérios para cada algoritmo.

Agora que estão apresentados os três algoritmos no artigo, convém fazer algumas considerações sobre a sua implementação, que serão abordadas a seguir.

3.4 Implementação da Aplicação

Nesta secção, serão apresentados os objectos que são monitorizados de forma a poder tomar as decisões presentes nos algoritmos, bem como algumas considerações sobre problemas de implementação e soluções encontradas ou sugeridas para trabalho futuro.

Na implementação do algoritmo RoD-A surgiram alguns problemas nomeadamente a falta de objectos que permitiriam o seu correcto funcionamento. Nas MIBs disponíveis não existem objectos que permitam saber se houve a recepção de tramas CTS ou ACK, sem que antes tenham sido detectados pacotes RTS ou DATA. Para além disso não é possível saber se essas tramas estão endereçadas a um nodo desconhecido ao ponto de acesso. A MIB *IEEE802dot11-MIB* apenas disponibiliza os objectos *dot11RTSSuccessCount* e *dot11RTSFailureCount*, em que o primeiro simplesmente incrementa o seu valor quando um CTS é recebido em resposta a um RTS, e o segundo incrementa o seu valor quando não é recebido um CTS em resposta a um RTS, ou seja, apenas indicam o sucesso ou falha do mecanismo RTS/CTS. Em relação às tramas ACK, apenas é disponibilizado o objecto *dot11ACKFailureCount* que é incrementado quando uma confirmação (ACK) não é recebida, mas era esperada.

Para conseguir isto, seria necessário fazer análise em tempo real pacote a pacote do tráfego na rede, e essa situação, para além de não viável no âmbito deste trabalho, introduziria uma carga bastante pesada no funcionamento do ponto de acesso. Mesmo assim, caso tal solução existisse implementada, a MIB do ponto de acesso teria de disponibilizar pelo menos um objecto que reportasse tal situação, por exemplo com o incremento de um contador. Com esse objecto seria relativamente fácil implementar a solução pretendida pelos autores do artigo referido anteriormente. Assim, o algoritmo RoD-A teve de ser posto de parte pelas razões acima mencionadas.

Com o algoritmo RoD-R a situação já é diferente. Nas MIBs analisadas existem objectos que permitem saber se o número de retransmissões começa a ser demasiado recorrente num dado intervalo de tempo. Esses objectos são *dot11RetryCount* e *dot11MultipleRetryCount*. Estes dois contadores incrementam o seu valor em situações semelhantes, com a diferença de o primeiro incrementar com tramas recebidas após ser pedida a sua retransmissão e o segundo incrementar quando foi necessário mais que uma retransmissão. No entanto, estes objectos provavelmente não funcionariam para o efeito pretendido, visto ser feita apenas a

contagem de retransmissões que acabaram por ter sucesso. Poderia também ser utilizado o contador *cDot11ClientMsduFails*, que é incrementado quando um MSDU (*MAC Service Data Unit*) não é transmitido com sucesso devido ao número de tentativas de transmissão ter excedido o seu limite.

Um objecto que tem o mesmo fim é o *dot11FailedCount*, e por estar presente na MIB *IEEE802dot11-MIB* fica garantida a sua implementação independentemente da marca do fabricante. Este objecto do tipo contador incrementa o seu valor quando um MSDU não é transmitido com sucesso, devido ao número de tentativas de transmissão ter excedido o valor de um dos objectos *dot11ShortRetryLimit* ou *dot11LongRetryLimit*, com 7 e 4 tentativas respectivamente. Com este objecto a ser monitorizado pode-se detectar a situação de nodos escondidos e activar o RTS/CTS mediante o aumento de retransmissões num dado intervalo de tempo, tal como sugerido no algoritmo RoD-R.

Como complemento, e para de certa forma garantir que as retransmissões não são causadas apenas por aumento significativo da carga no sistema, será utilizado em conjunto um objecto que permita saber a carga do sistema. Neste caso optou-se pela carga de CPU do ponto de acesso. Esse objecto não está presente nas MIBs analisadas. Contudo, após uma pesquisa descobriu-se que tal objecto está disponível em MIBs *Cisco*, a *CISCO-PROCESS-MIB* e a sua versão anterior para equipamento mais antigo *OLD-CISCO-CPU-MIB*. Na mais recente são disponibilizados objectos que permitem saber a um dado momento a percentagem de ocupação (carga) do CPU. Esses objectos são *cpmCPUTotal5sec*, *cpmCPUTotal1min* e *cpmCPUTotal5minRev*, que dão a informação de carga do sistema, respectivamente durante os últimos 5 segundos, 1 minuto ou 5 minutos. Na mais antiga, existem objectos semelhantes mas com outra nomenclatura (OID), *busyPer*, *avgBusy1* e *avgBusy5*. Conforme o equipamento disponível serão utilizados os objectos mais convenientes. Mesmo no caso de o ponto de acesso para testes ser de outro fabricante, costumam haver objectos idênticos que forneçam o mesmo tipo de dados.

No caso do algoritmo RoD-C, as perspectivas de implementação também não seriam as melhores. Apesar disso foi encontrada uma alternativa que será abordada mais adiante. Tal como sugerido no artigo, nos algoritmos RoD-A e RoD-R o nodo (ponto de acesso) toma a decisão de utilizar o RTS/CTS baseado na informação recolhida por si próprio. Tal não acontece no algoritmo RoD-C, que depende da informação fornecida pelo receptor pretendido numa dada transmissão e troca dessa mesma informação na rede. Neste cenário cada nodo calcula a sua taxa de colisão através de um detector de colisões baseado em níveis de energia e transmite essa informação aos nodos vizinhos. Como neste trabalho o foco está sobre o ponto de acesso utilizando o paradigma da arquitectura centralizada SNMP, este algoritmo ficou também excluído.

No entanto, há uma alternativa ao algoritmo RoD-C que se sugere como *RTS on demand using errors* (RoD-E), em que é monitorizada a ocorrência de erros na transmissão, nomeadamente erros de descodificação. Apesar de não ser um indicativo tão forte como o aumento do número de retransmissões, optou-se por esta alternativa por existirem objectos normalizados em MIBs .O algoritmo será muito semelhante ao RoD-R com a diferença que em vez de se monitorizar o número de retransmissões num intervalo de tempo, irá ser monitorizado o número de erros de descodificação. Uma maneira de evitar a activação do RTS/CTS quando não há nodos escondidos é a seguinte. Ao activar o RTS/CTS, continua-se a monitorizar a frequência dos erros de descodificação. Se esta diminuir, pode-se concluir á partida que eram nodos escondidos que estavam a causar os erros. Caso contrário, os erros são causados por outro problema de rede e com isto o RTS/CTS é desactivado.

Uma falha de descodificação pode ocorrer quando o cálculo de verificação da trama (FCS) falha, que por sua vez reflecte o estado de integridade da rede, e poderá indicar a presença de nodos escondidos. Com este fim em vista, será utilizado o objecto *dot11FCSErrorCount* que faz a contagem do número de erros de FCS detectados num MPDU recebido. Poderia ter sido utilizado o objecto *cd11IfRecFrameMacCrcErrors* que incrementa sempre que é recebida uma trama com um erro CRC a nível MAC, mas como o FCS é um algoritmo de CRC, não há vantagens em utilizar um objecto de uma MIB proprietária já que *dot11FCSErrorCount* está presente numa MIB normalizada, não proprietária, a MIB *IEEE802dot11-MIB*. Ainda como alternativa poderia ter sido utilizado o objecto *cDot11ClientMicErrors* que indica o número de erros MIC (*Message Integrity Check*), mas como o algoritmo que implementa o MIC é o algoritmo *Michael* obrigaria ao uso dessa técnica de encriptação 802.11, que não foi considerada nesta dissertação.

Para activar o mecanismo RTS/CTS irá ser utilizado objecto *dot11RTSThreshold* da *IEEE80-2dot11-MIB*. Conforme a sua definição por omissão, o RTS/CTS é activado com pacotes maiores que 2347 octetos. Se o seu valor for alterado para zero, a negociação RTS/CTS é activada em qualquer transmissão, independentemente do tamanho do pacote. Então, para activar o RTS/CTS nos algoritmos mencionados basta fazer uma operação SNMP *set*, colocando o valor a zero. Para desactivar, repete-se a operação, repondo o valor por omissão, para activar o RTS/CTS apenas quando houver pacotes grandes em circulação na rede.

Na secção que define o algoritmo a usar, foi mencionado o facto de o programa não poder monitorizar os clientes de uma rede sem fios. Isto devido ao facto de a maioria dos computadores ou sistemas terminais não terem um agente SNMP que suporte a MIB *IEEE802dot11*. Assim, decidiu-se implementar a aplicação apenas para pontos de acesso. Contudo, e caso no futuro as drivers das placas de rede dos sistemas terminais implementem os objectos desta MIB, seria apenas necessária uma de duas soluções, que utilizariam o mesmo programa implementado:

- A primeira passaria por instalar e configurar correctamente o agente SNMP no computador e instalar o programa localmente. Desta forma, o programa teria acesso local para poder activar o mecanismo RTS/CTS dinamicamente, sem haver a necessidade de um sistema centralizado de monitorização que introduziria tráfego SNMP na rede. Em termos de segurança esta seria a opção mais segura já que o agente SNMP seria configurado para apenas responder a pedidos com endereço local, ou seja, da própria máquina.

- A segunda solução seria a utilização do programa como sistema centralizado de monitorização, sem haver a necessidade de instalação do programa nos computadores. Tal como o programa a implementar recebe uma lista de pontos de acesso a monitorizar, também seria possível aplicar a mesma estratégia para os computadores. A diferença seria que para os pontos de acesso os endereços destes teriam de ser introduzidos manualmente pelo administrador de rede, enquanto no caso dos computadores (clientes) teria de ser feito de forma automatizada para se poder adicionar à medida que novos clientes se associam a um ponto de acesso. Isto poderia ser conseguido através da utilização da *CISCO-DOT11-ASSOCIATION-MIB*. Adicionando ao programa a funcionalidade de percorrer a *cDot11ClientConfigInfoTable* que é uma tabela que contém por sua vez entradas *cDot11ClientConfigInfoEntry*. Seria criada uma entrada por cada cliente associado a um ponto de acesso. Essas entradas para além de outros objectos, conteriam o objecto *cDot11ClientIpAddress* que tem o endereço IP do cliente associado. Assim seria possível recolher todos os endereços *IP* associados no ponto de acesso e iniciar a monitorização e activação dinâmica do mecanismo RTS/CTS.

3.5 Detalhes da Implementação

Para a implementação foi utilizada a biblioteca Java SNMP4J. Esta biblioteca *open-source* tem uma vasta API permitindo trabalhar com SNMP em ambiente Java. Tem disponíveis entre outras as operações *SNMP get* e *SNMP set* que serão necessárias para interagir com o agente SNMP presente no ponto de acesso de forma a conseguir implementar os algoritmos propostos. O programa terá uma interface gráfica, de forma poder ser feita a visualização dos valores representados num gráfico. Para criar esses gráficos foi utilizada a biblioteca JFreeChart, também *open-source*.

Para conseguir ler os valores SNMP de um agente, a API SNMP4J precisa que sejam fornecidos os OIDs (*Object Identifiers*), bem como o tipo de variável SNMP, para poder utilizar os métodos Java apropriados. Essas informações são apresentadas de seguida:

- ***dot11FailedCount***
  OID: 1.2.840.10036.2.2.1.3

Tipo: *Counter32*
Permissão: Leitura

- ***cpmCPUTotal5sec***
  OID: 1.3.6.1.4.1.9.9.109.1.1.1.1.3
  Tipo: *Gauge32*
  Permissão: Leitura

- ***busyPer***
  OID: 1.3.6.1.4.1.9.2.1.56
  Tipo: *INTEGER*
  Permissão: Leitura

- ***dot11FCSErrorCount***
  OID: 1.2.840.10036.2.2.1.12
  Tipo: *Counter32*
  Permissão: Leitura

- ***dot11RTSThreshold***
  OID: 1.2.840.10036.2.1.1.2
  Tipo: *INTEGER*
  Permissão: Leitura e Escrita

De uma forma resumida o programa recolhe valores de um ou mais pontos de acesso através da operação *get* do SNMP (em *Java* através da biblioteca SNMP4J), efectuando assim a monitorização de cada ponto de acesso, recorrendo a tecnologia *multi-thread*. Depois guarda-os numa estrutura de dados apropriada, onde serão efectuados cálculos e aplicado o algoritmo sugerido, de forma a saber se, num dado momento, o resultado do algoritmo sugere que seja activado o RTS/CTS ou não. Para além disso, é possível alternar entre dois modos de monitorização, um que usa o algoritmo RoD-R e outro que usa o algoritmo RoD-E. Caso seja sugerido, o RTS/CTS é activado através do uso de *snmpSet* para a variável na MIB do ponto de acesso referenciada acima (*dot11RTSThreshold*).

De seguida é apresentado o diagrama do funcionamento geral do programa. Os detalhes sobre o que fazem cada uma das *threads* mencionadas no diagrama serão explorados mais à frente.

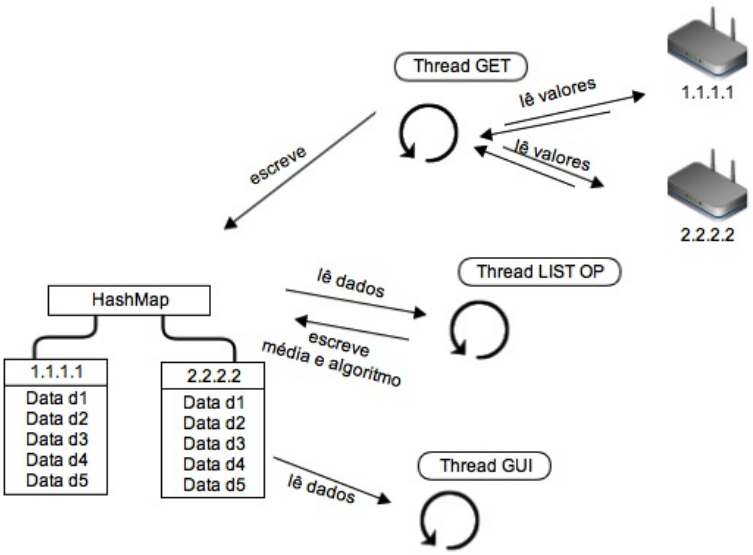


Figura 3.6: *Diagrama geral do funcionamento do programa.*

A estrutura de dados usada pela aplicação de gestão é relativamente simples e consiste num *HashMap* em que a chave é o endereço *IP* do ponto de acesso, e o valor é a lista de objectos Data monitorizados por SNMP na forma de *ArrayList*. Assim será usado um *HashMap<String, ArrayList<Data>>*. Esta estrutura de dados está presente na Figura 3.6.

Optou-se pelo *HashMap* tendo em conta que as mensagens SNMP podem não chegar por ordem devido ao SNMP ser assíncrono e usar UDP para o seu transporte. Assim, à medida que as threads executam, os valores da operação *GET* são extraidos para o objecto *Data*, e este é inserido no *ArrayList* correspondente, visto que a chave é o endereço *IP* do ponto de acesso consultado. Após cada inserção, o *ArrayList* correspondente é ordenado por tempo, através do campo *time* de *Data* que contém o *uptime* do ponto de acesso em questão.

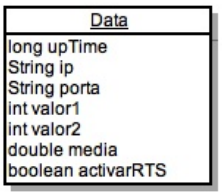


Figura 3.7: *Variáveis do objecto Data.*

Este *HashMap* foi encapsulado numa classe denominada *FakeHashMap*, de forma a poder fazer o controlo de concorrência das operações padrão *Java* da classe *HashMap*. Esse controlo é feito através do uso de *ReentrantLock* e das suas operações *lock* e *unlock*, protegendo as secções críticas de execução. Assim é garantido que o meio partilhado é acedido por uma *thread* de cada vez.

O programa principal está dividido em três *threads* (módulos) independentes que executam três tarefas distintas paralelamente, partilhando em comum a estrutura de dados mencionada anteriormente, fazendo escrita ou leitura na mesma.

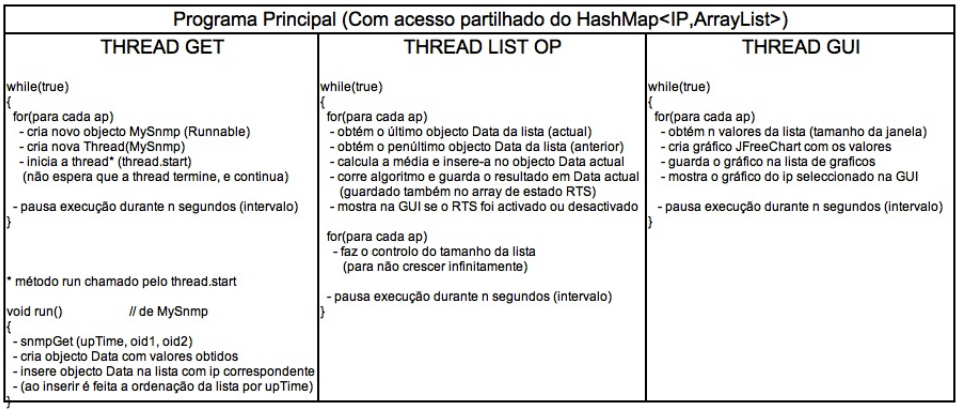


Figura 3.8: *Esquema do programa com pormenores das threads.*

A *thread get* escreve na estrutura de dados os valores obtidos através do *get* SNMP. Nesta *thread* são iniciadas para execução *sub-threads MySnmp* para cada ponto de acesso existente. Cada objecto *MySnmp (sub-thread)* contém um apontador para a estrutura de dados para poder escrever durante a sua execução. Contém ainda o endereço *IP*, porta de comunicação e os OIDs (*object identifiers*) para indicar os objectos SNMP a monitorizar. Esta *thread* consiste num ciclo contínuo em que são iniciadas as *sub-threads* para execução e no fim de cada iteração é pausada a execução durante um intervalo de tempo especificado, intervalo esse que será o intervalo de monitorização.

Em cada iteração do ciclo principal existe um ciclo interior que mediante o número de pontos de acesso existentes lança *n threads* distintas, cada uma com o endereço *IP* de cada um dos pontos de acesso. Cada uma dessas *threads* são conseguidas através da implementação da interface *Runnable* na classe *MySnmp*, que obriga a implementação do método *run()*, método esse que é executado quando é feito *start()* à thread. No método *run()*, o método que contém as instruções a executar por cada *thread* criada, é executado o método *get* da mesma classe de forma a obter os valores pretendidos através de SNMP. Depois, esses valores são inseridos na estrutura de dados, na lista com o *IP* correspondente, sendo essa lista ordenada após a inserção.

Assim, no caso de uma *thread* demorar muito tempo a responder e a adicionar os valores à lista, as outras *threads* não ficam à espera. Cada uma das *threads* é iniciada e o programa continua a sua execução para a iteração seguinte após o intervalo de monitorização. Com isto consegue-se que o programa não bloqueie por causa de uma *thread* que não consiga obter resposta SNMP.

A *thread* simulador, que não aparece no diagrama da figura 13 por ser usada simplesmente para testes, simula a obtenção de valores de vários pontos de acesso e insere-os na estrutura de dados. Esses valores foram introduzidos manualmente de forma a conseguir simular as várias situações que causam as diferentes respostas previstas no algoritmo. Esta *thread* é vista apenas como código para testes, de forma a conseguir confirmar se o algoritmo responde conforme suposto, bem como o correcto funcionamento do *multi-threading* do programa com vários pontos de acesso artificiais a gerar valores.

A *thread list op* (operações sobre as listas) faz a leitura da estrutura de dados dos valores escritos pela *thread get*, fazendo operações com estes valores, tais como cálculo da média e aplicação do algoritmo. Também nesta *thread* existe um ciclo contínuo em que no fim é feita uma pausa na execução para criar intervalos de monitorização. Em cada iteração desse ciclo existem vários sub-ciclos que correm n vezes mediante o número de pontos de acesso a monitorizar.

O primeiro desses sub-ciclos que é executado tantas vezes quantos ponto de acesso existem, vai buscar o objecto Data actual e o anterior (o último e penúltimo objectos da lista de cada ponto de acesso). Com estes dois objectos é possível obter os valores de forma a poder calcular a média do número de retransmissões ou erros num dado intervalo de tempo. Depois de a média ser calculada o seu valor é inserido no campo média do objecto Data actual, de forma a poder ser lido pela thread da interface gráfica, apresentada de seguida.

A média é calculada através da seguinte fórmula:

$$media = \frac{\Delta retrans}{\Delta tempo} = \frac{retransmissoes_{actual} - retransmissoes_{anterior}}{tempo_{actual} - tempo_{anterior}}$$

Com a média calculada já é possível correr o algoritmo e saber se se recomenda a activação ou desactivação do mecanismo RTS/CTS. O algoritmo é executado imediatamente a seguir ao cálculo da média, visto necessitar dela como parâmetro. Mais uma vez, um por cada ponto de acesso. O método de cálculo do algoritmo precisa de argumentos, que são a média, a carga de CPU e o estado actual do mecanismo RTS/CTS (se está activado ou desactivado). Os dois primeiros são fornecidos pela média e carga de CPU do objecto *Data* actual, o terceiro é fornecido através de um *array* de booleanos que contém o estado actual do mecanismo RTS/CTS para cada um dos pontos de acesso em monitorização.

O método do algoritmo retorna um valor booleano, de recomendação de activação ou desactivação do mecanismo RTS/CTS, e esse valor é guardado no array de estado mencionado anteriormente. Assim, é possível saber sempre o estado do mecanismo RTS/CTS. De notar que após o retorno desse valor é tomada a decisão de activar ou desactivar o RTS/CTS, executando as instruções necessárias para tal.

Existe outro sub-ciclo que executa uma vez para cada ponto de acessoe que garante que o *array* de cada ponto de acesso não cresce infinitamente. Isto é muito importante, visto que o programa poderá ficar a executar horas, dias ou até semanas continuamente em dispositivos de memória limitada, como um *Raspberry Pi*. Assim, para cada array (ponto de acesso) individualmente, se o tamanho da lista de valores exceder o tamanho especificado no componente da interface gráfica *jComboBoxJanela*, é removido o valor mais antigo. Desta forma é mantido um array fixo com n valores, que serão lidos pela *thread* da interface gráfica, de forma a gerar o gráfico pretendido. Com isto também é conseguido o efeito de gráfico rolante. Como com cada novo valor a entrar é eliminado o mais antigo, no gráfico aparece sempre o novo valor, e o antigo desaparece.

No ciclo principal é ainda usado um mecanismo de detecção de *reboots* e estado de cada ponto de acesso. Isto foi conseguido pelo facto de que quando um ponto de acesso faz um *reboot*, os valores de *upTime* são forçados a aparecer a zero. Quando é detectado o valor zero nesse campo, a lista desse ponto de acesso é apagada na totalidade. Quando isto acontece o tamanho da lista fica a zero e assim é possível sinalizar o *reboot* ou o estado desligado do ponto de acesso na interface gráfica.

Na *thread GUI* são lidos os dados presentes na estrutura de dados, de forma a poder construir o gráfico que representa visualmente o estado das variáveis monitorizadas pelo programa. Isso é conseguido através da biblioteca *JFreeChart* para *Java*. Esta secção é relativamente simples e consiste num ciclo contínuo como os anteriores, com uma pausa na execução. Em cada iteração, e novamente uma vez para cada ponto de acesso presente, são lidos os valores a mostrar no gráfico e guardados em *arrays* temporários, para serem recebidos como argumento no método de construção do gráfico. Depois de o gráfico ser construído, é guardado num *array* de *JFreeChart* para poder ter sempre presentes os gráficos de todos os pontos de acesso. Isto com o intuito de na interface gráfica ser possível seleccionar o *IP* do ponto de acesso desejado, mostrando o gráfico correspondente.

Na interface gráfica estão presentes várias funcionalidades para além da visualização do gráfico com os valores de monitorização do ponto de acesso (Ver Figura 3.9):

1. O botão Iniciar inicia a execução do programa.
2. A caixa de selecção de IP permite a selecção do ponto de acesso cujo gráfico é visualizado.
3. A caixa de selecção permite o aumento ou diminuição do número de elementos da lista visualizados, que corresponde ao tamanho da janela de visualização do gráfico.

4. O botão de *ON/OFF* do algoritmo permite ao utilizador seleccionar se quer que o algoritmo seja aplicado ou apenas recomende a aplicação do algoritmo, mas não aplique.

5. Os campos de texto *Recomendado* e *Aplicado*. O primeiro mostra se o algoritmo deveria ser aplicado e o segundo mostra se está realmente a ser aplicado ou não. Se o botão 4 estiver *ON*, aparece simultaneamente ACTIVADO no recomendado e no aplicado (aplicou o algoritmo). Se estiver *OFF*, mostra apenas ACTIVADO no recomendado (sugeriu o algoritmo).

6. A área de texto permite saber simultaneamente o estado RTS/CTS de todos os pontos de acesso a serem monitorizados.

7. A etiqueta *Tamanho Lista* permite saber o tamanho actual da lista de cada ponto de acesso.

8. A etiqueta *Refresh* permite saber o intervalo de tempo usado nos ciclos de monitorização

9. A etiqueta *Estado* permite saber o estado actual do ponto de acesso escolhido na caixa de selecção de IP.

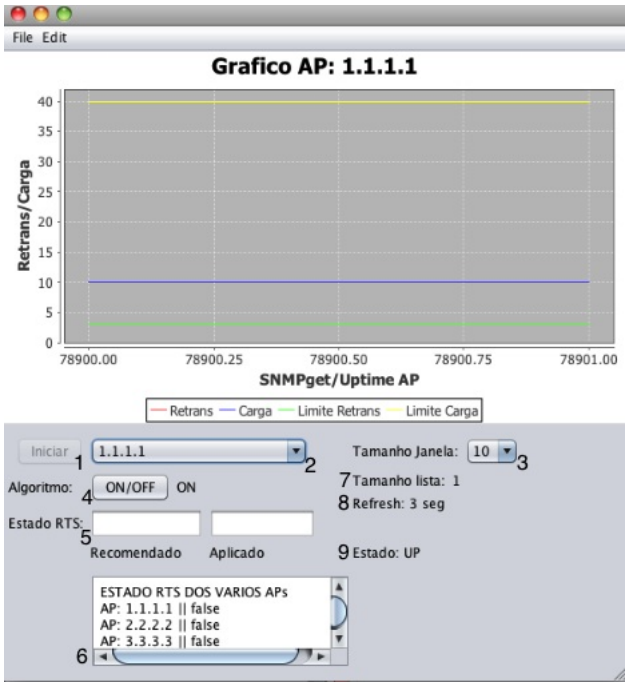


Figura 3.9: *Interface gráfica da aplicação de gestão.*

Nas próximas figuras são representadas situações específicas do funcionamento do programa.

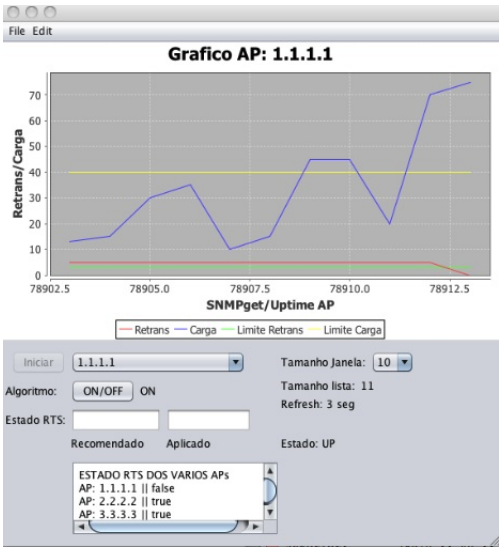


Figura 3.10: *Execução do programa sem activação do algoritmo.*

Na figura 3.10 pode ver-se que, para o ponto de acesso com o endereço 1.1.1.1, os campos de texto *Recomendado* e *Aplicado* estão vazios, o que significa que o algoritmo não recomenda a activação do mecanismo RTS/CTS. Isto porque apesar da média de retransmissões estar acima do limite, a carga de CPU também está. Logo, segundo os critérios do algoritmo, não é necessária a activação.

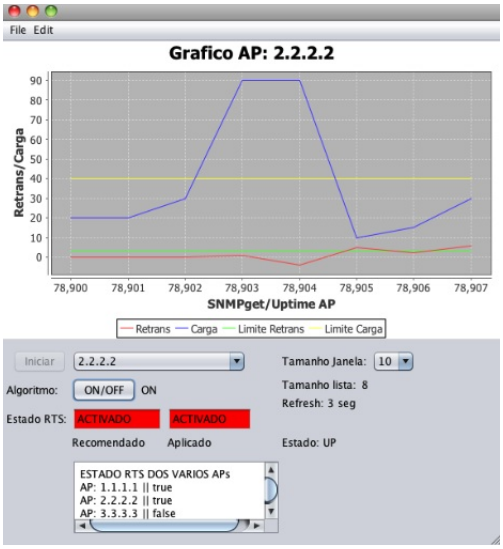


Figura 3.11: *Execução do programa com activação do algoritmo.*

A figura 3.11 retrata a situação em que para o ponto de acesso 2.2.2.2 os campos de texto *Recomendado* e *Aplicado* estão a vermelho e com o texto *ACTIVADO*, o que significa que o algoritmo recomenda e a aplicação de gestão activa o mecanismo RTS/CTS. Neste caso, a média de retransmissões está acima do limite e a carga de CPU abaixo do limite, o que se segundo os critérios do algoritmo escolhido, deve proceder-se à activação do mecanismo

RTS/CTS.

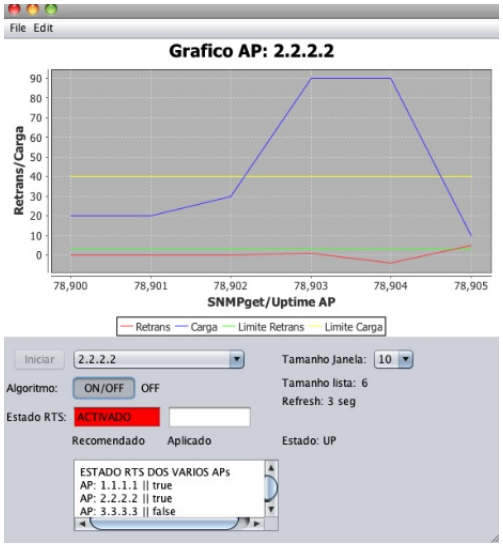


Figura 3.12: *Execução do programa com recomendação de activação do algoritmo, mas não o aplica.*

Na situação da figura 3.12, para o ponto de acesso 2.2.2.2 o campo *Recomendado* está a vermelho e apresenta o texto *ACTIVADO*, mas o campo *Aplicado* está vazio. Isto acontece porque o botão de algoritmo está no estado *OFF* (algoritmo desligado), o que significa que o algoritmo apesar de ser recomendado, não foi aplicado.

Existe um ficheiro de configuração que permite ao gestor da rede especificar vários parâmetros de funcionamento do programa tais como, modo operacional, tempo do intervalo de monitorização, limites do algoritmo, e a lista de pontos de acesso a monitorizar.

```
// configuracao do modo operacional (0 = Rod-R | 1 = RoD-E )
// e tempo de sleep em segundos
conf 0 3
// configuracao de limites
// (limite retransmissao, limite carga, limite erros)
limites 3 40 10
//lista de pontos de acesso (ip  porta)
//ap 192.168.10.254 161
ap 1.1.1.1 100
ap 2.2.2.2 100
```

Para além disto, foi ainda adicionada a capacidade de o programa criar ficheiros de *log* tanto no formato *txt* como *csv*, de forma a possibilitar análise posterior dos valores obtidos numa sessão do programa.

Ainda como funcionalidade extra foi adicionado um botão que activa ou desactiva permanentemente o mecanismo RTS/CTS transformando o funcionamento da aplicação de gestão numa simulação. Foi adicionado à aplicação mais uma funcionalidade. Uma terceira linha no gráfico que representa a média ponderada dos valores monitorizados (retransmissões ou erros).

Capítulo 4

Testes e Resultados

Este capítulo é dedicado a todo o procedimento da fase de testes da aplicação implementada. Inicialmente será explicada a configuração do ponto de acesso sem fios usado para os testes. Seguidamente serão descritos todos os testes realizados, com os respectivos pormenores tais como equipamento usado, localização do equipamento e o que foi testado concretamente. Finalmente serão apresentados os resultados dos testes e conclusões tiradas a partir desses resultados.

Para poder realizar os testes do programa implementado neste trabalho, foi fornecido pela Universidade do Minho um ponto de acesso sem fios *Cisco Aironet 1100*. Uma das principais condicionantes para a escolha de tal equipamento seria o suporte das MIBs referidas anteriormente na secção de Implementação. Assim é garantido que podem ser feitas consultas SNMP aos objectos pretendidos, de forma a obter os seus valores para monitorizar o estado do ponto de acesso e tomar a decisão de activar o mecanismo RTS/CTS consoante esses mesmos valores.

Por omissão, o ponto de acesso tem o agente SNMP desactivado, e consequentemente não tem configuradas *community strings* para acesso SNMP. Para activar o agente SNMP no equipamento é necessária a configuração de uma ou mais *community strings*, que definem uma espécie de palavra-passe para a aplicação de gestão poder comunicar com o agente presente no dispositivo. Para além disso é preciso definir uma *MIB view*, na qual é especificado o sub-conjunto de objectos acessíveis pela *community string*.

Por omissão, ou seja, sem a criação de uma *MIB view*, fica apenas disponível para consulta a sub-árvore de objectos da *MIB-II*. Como outra das MIBs a ser utilizada é a *IEEE802dot11-MIB*, e esta pertence a outro ramo na árvore de objectos, é necessária a criação de uma *MIB view* mais abrangente que inclua tanto a *IEEE802dot11-MIB*, bem como outras MIBs *Cisco*. Foram introduzidos os seguintes comandos na configuração do ponto de acesso:

```
snmp-server view isoview iso included
snmp-server community public view isoview RO
snmp-server community private view isoview RW
```

A primeira linha define a *MIB view* desejada. A segunda linha define uma *community string* apenas com permissões de leitura usando a *view* definida, enquanto que a terceira linha faz o mesmo mas com permissões de leitura e escrita. Após tentar aceder aos objectos pretendidos para monitorização no software desenvolvido, verificou-se que estas configurações estão correctas e pode-se então prosseguir com os testes.

Para determinar se o algoritmo sugerido resolve ou atenua o problema dos nodos escondidos, nos testes a realizar foi necessário simular esse problema e monitorizar os valores de retransmissões e erros. Para isso, foram usados dois computadores colocados no limite do alcance do sinal do ponto de acesso. Desta forma, consegue-se que ambos os computadores consigam estar ao alcance do ponto de acesso, mas não ao alcance um do outro. Para poder confirmar se a situação de nodos escondidos foi criada com sucesso, foi necessário introduzir tráfego na rede, pondo os dois computadores a descarregar um ficheiro de 700MB através um servidor FTP, instalado no mesmo computador que corre o programa de monitorização, esperando-se o aumento do número de retransmissões ou erros.

Nos primeiros testes verificou-se que a instância do objecto do tipo contador *dot11FailedCount* da MIB *IEEE802dot11* incrementa quando um MSDU não é transmitido com sucesso e consequentemente, é necessária a retransmissão. Concluiu-se que o contador não é útil para o efeito desejado devido ao facto de apenas considerar as transmissões (do ponto de acesso para os clientes) e não as recepções (dos clientes para o ponto de acesso). O que realmente interessa são as retransmissões dos clientes, já que são os clientes que são os nodos escondidos e é neles que o número de retransmissões aumentaria. Verificou-se de facto, que os valores no gráfico da aplicação de monitorização não aumentaram muito significativamente quando era introduzida carga na rede. Provavelmente se fosse possível monitorizar por SNMP os clientes, seria possível observar um aumento significativo do número de retransmissões.

Assim, para conseguir notar diferenças significativas tentou-se a monitorização do objecto *dot11FCSErrorCount* que contabiliza o número de erros FCS detectados. A ideia geral seria que o aumento do número de erros de processamento dos pacotes, traduzisse também o aumento de colisões devido à existência de nodos escondidos. Com isto abandonou-se o algoritmo RoD-R que monitoriza as retransmissões e trabalhou-se apenas com o algoritmo RoD-E que monitoriza os erros.

As primeiras simulações de nodos escondidos foram efectuadas da seguinte forma: inicialmente, um dos computadores (PC1) foi colocado no limite do alcance do ponto de acesso,

mas de forma a ainda ter sinal (30%) para poder descarregar o ficheiro do servidor FTP. O outro computador (PC2) foi colocado muito perto do ponto de acesso. Para melhor se perceber os gráficos apresentados, convém explicar o o processo sequencial:

1. Depois da preparação dos computadores, foi iniciado o programa de monitorização com um tamanho de janela suficientemente grande para poder capturar todo o procedimento.

2. Com o programa a correr, foi iniciada a transferência FTP no PC1.

3. Alguns segundos depois, foi iniciada a transferência FTP no PC2.

4. Alguns segundos depois, o PC2 foi levado para longe do ponto de acesso, quase até ao ponto de perder o sinal (30%).

5. Depois de dois a três minutos, o PC2 foi levado para junto do ponto de acesso novamente.

6. Aguardou-se alguns segundos e terminou-se o programa.

Este procedimento foi repetido várias vezes.

Durante os testes, o intervalo entre consultas SNMP foi de cinco segundos. Em cada um dos gráficos estão presentes três linhas. A linha vermelha representa a média de erros FCS, a linha azul representa a carga de CPU e a linha verde representa a média ponderada dos erros FCS, de forma a obter-se uma linha mais estável, visto os valores variarem bastante no eixo vertical. No eixo horizontal estão representados os valores de *up time* do ponto de acesso em *timeticks* (centésimos de segundo). A a biblioteca usada para representar os gráficos adapta-se automaticamente ao valor mais alto, os valores da carga de utilização do CPU foram divididos por 10. Estes testes foram efectuados com o mecanismo RTS/CTS desligado.

Conforme é possível observar na figura 4.1, onde estão representados os gráficos correspondentes aos primeiros quatro testes efectuados, os valores demonstram um comportamento idêntico ou padrão. Quando o PC2 é afastado do ponto de acesso, até ficar quase fora do seu alcance e consequentemente fora do alcance do PC1 (do ponto 4 ao ponto 5), a média de erros aumenta significativamente. É provável que o incremento e decremento relevante dos erros sejam devidos não só à distância para o ponto de acesso, como também ao fenómeno de nós escondidos.

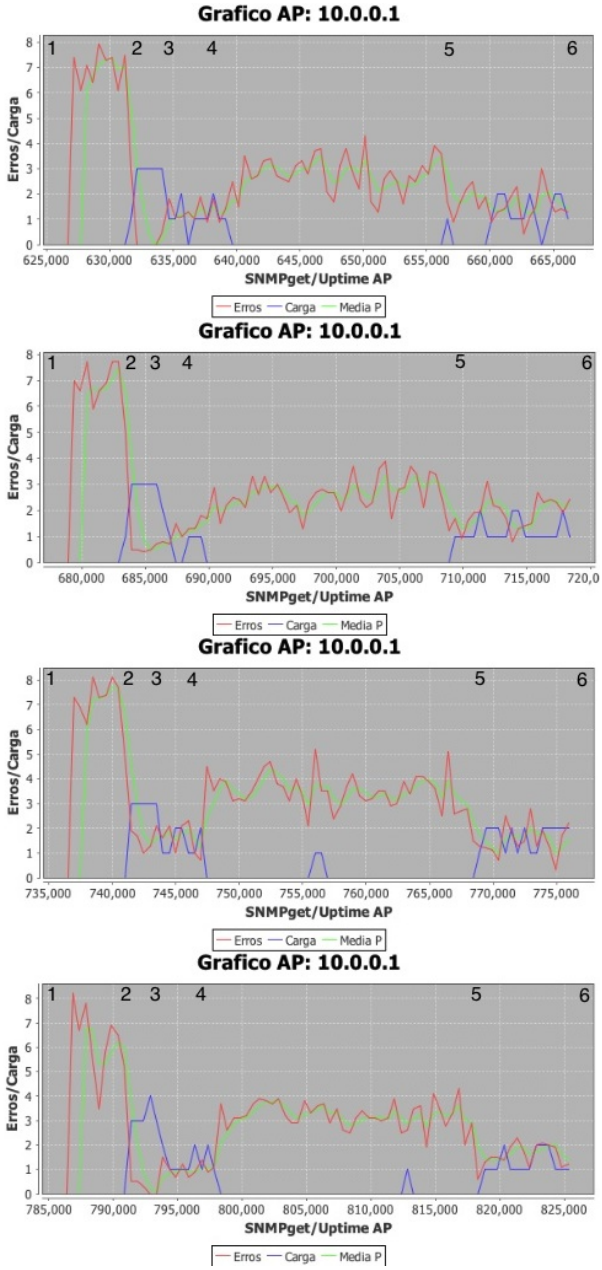


Figura 4.1: *Gráficos de alguns dos testes efectuados.*

Entretanto foram repetidos os mesmos testes, mas desta vez com o mecanismo RTS/CTS ligado. Para o conseguir, foi desligada a activação automática do mecanismo no aplicação implementada e activado manualmente. Conforme já referido anteriormente, alterando o valor do objecto *dot11RTSThreshold* para zero, todos os pacotes são transmitidos usando a negociação RTS/CTS. Podia esperar-se que ao activar o mecanismo RTS/CTS no ponto de acesso, que a média de erros baixasse, mas não foi isso que aconteceu. De certa forma isto já era previsível, já que seria necessário activar o mecanismo RTS/CTS nos clientes. Os gráficos destes testes não diferem quase nada em relação aos gráficos anteriores.

Será que a média de erros baixou, mas de forma quase irrisória e não foi possível observar no gráfico? Será que o mecanismo RTS/CTS não fez qualquer diferença? Depois de

verificar e confirmar que os valores obtidos pelo programa estavam correctos, procedeu-se à investigação da causa deste problema.

Ao investigar a primeira dúvida surgiu a hipótese de que dois computadores poderiam não ser suficientes para fazer com que a média de erros aumentasse consideravelmente. Pensou-se nisto devido ao facto de o ponto de acesso pertencer à gama empresarial, que tem produtos com melhores características, como por exemplo um melhor rendimento das comunicações, mesmo com bastante tráfego em circulação.

Depois de alguma pesquisa foi encontrado um estudo realizado por [Parsons, 2013], no qual foram efectuados testes, para tentar determinar o ponto de ruptura de vários pontos de acesso de diferentes marcas e modelos. Por ponto de ruptura entenda-se a altura em que o ponto de acesso já está tão sobrecarregado que deixa simplesmente de responder aos pedidos e de funcionar correctamente. Para além de testes de rendimento, foram feitos testes sobre o número de erros de cada ponto de acesso.

Neste estudo, para cada ponto de acesso, um de cada vez, estavam ligados vários *iPads*, cada um dos quais a fazer *streaming* de vídeo enquanto num *MacBook Pro* corria uma transferência FTP (*upload e download*) de um ficheiro de 600MB do servidor FTP que corria noutro computador. No teste inicial fizeram parte cinco *iPads*, e em cada nova iteração do teste eram adicionados outros cinco, até chegar ao ponto de ruptura do ponto de acesso, que foi considerado quando mais de 50% dos *iPads* não conseguiam receber em condições o vídeo por streaming. De seguida são apresentados alguns dos resultados desses testes.

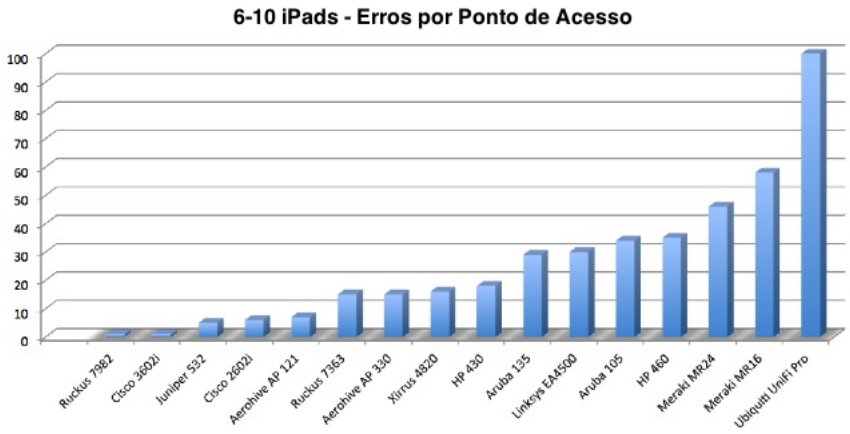


Figura 4.2: *Testes com 6 a 10* iPads. *(Imagem adaptada do estudo "Wi-Fi Stress Test") [Parsons, 2013]*

Mesmo após adicionar mais cinco *iPads* aos 5 iniciais, a maioria dos pontos de acesso ainda conseguem controlar relativamente bem a carga introduzida. Alguns tiveram mais erros, mas apenas um não conseguiu terminar os testes com 10 *iPads*.

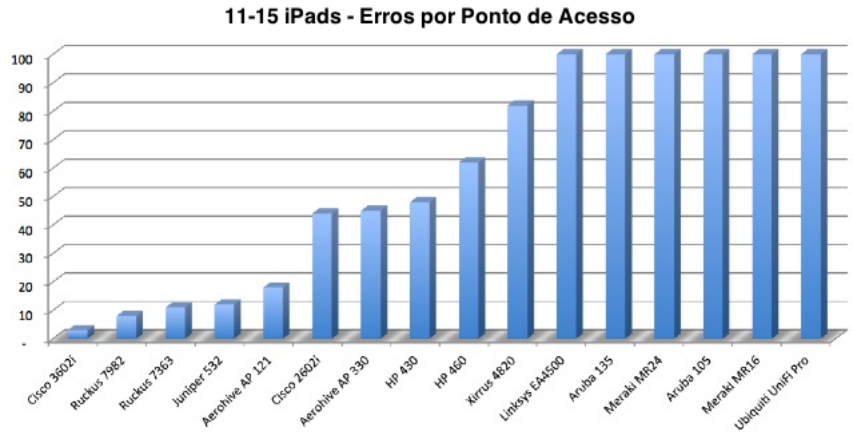


Figura 4.3: *Testes com 11 a 15* iPads. *(Imagem adaptada do estudo "Wi-Fi Stress Test") [Parsons, 2013]*

Com 11 a 15 *iPads*, mais de um terço dos ponto de acesso não conseguiram acompanhar a tarefa pedida. Alguns dos pontos de acesso ainda conseguiram manter o seu rendimento aceitável durante o *download* FTP, o mesmo já não aconteceu com o *upload* FTP, onde se notou maior degradação do rendimento.

Figura 4.4: *Testes com 16 a 20* iPads. *(Imagem adaptada do estudo "Wi-Fi Stress Test") [Parsons, 2013]*

Apenas quando se alcançou os 20 *iPads*, a maioria dos pontos de acesso falhou. Logo, pode concluir-se que à partida, apenas com dois terminais, conforme os testes efectuados, a média de erros não aumentará significativamente. Isto partindo do princípio que o ponto de acesso utilizado neste trabalho tenha mais ou menos a mesma performance do dispositivo da mesma marca nos testes apresentados, o *Cisco 3602i*.

Para investigar a segunda dúvida, se o mecanismo RTS/CTS fez alguma diferença ou não, foi colocada uma nova hipótese: pela razão indicada na secção de implementação, o programa não activa dinamicamente o mecanismo RTS/CTS nos clientes, apenas nos pontos de acesso. Mas se fosse possível activar o mecanismo RTS/CTS manualmente nos clientes,

dever-se-iam notar melhorias significativas no combate aos efeitos dos nodos escondidos. Em algumas versões do sistema operativo *Windows 7* é possível escolher nas definições avançadas da placa de rede a activação do RTS/CTS, noutras versões, é possível alterar o valor do *RTS Threshold*.

Seguiram-se novos testes de forma a poder verificar as diferenças tendo activado o mecanismo RTS/CTS, desta vez nos clientes. Foram feitas duas rondas de testes, sem e com nodos escondidos. Na primeira, foram postos a fazer download de um ficheiro de 160MB três computadores e um telemóvel *Android*, todos com clientes FTP instalados, todos dentro da mesma sala, sem a situação de nodos escondidos e foram medidos os tempos de transferência. Esta primeira ronda serviu de controlo para a segunda ronda de testes, com o intuito de poder relacionar tempos de transferência com melhores condições de comunicação, criadas pela activação do mecanismo RTS/CTS quando exitem nodos escondidos. Na Tabela 4.1 são apresentados os resultados dessa primeira ronda de testes.

	Com RTS/CTS				Sem RTS/CTS			
	PC1	PC2	PC3	Android	PC1	PC2	PC3	Android
exec.1	4.31	4.43	4.38	4.36	4.05	4.16	4.29	4.42
exec.2	4.28	4.34	4.05	4.24	3.48	4.19	3.21	4.50
exec.3	4.15	4.42	3.56	4.37	4.18	4.31	4.24	4.44
exec.4	4.20	4.26	4.04	4.02	4.04	4.10	3.56	4.55
exec.5	4.23	4.32	3.55	4.26	3.38	3.50	3.09	4.35
exec.6	4.27	4.36	4.23	4.32	4.27	4.40	4.1 0	4.15
exec.7	4.26	4.31	3.56	4.16	4.27	4.39	4.12	4.15
média	4.22	4.33	4.19	4.23	4.03	4.14	4.22	4.33
variância	0	0	0.1	0.02	0.14	0.14	0.24	0.03
desvio padrão	0.05	0.06	0.32	0.14	0.37	0.38	0.49	0.18

Tabela 4.1: *Tempos de transferência em minutos, com e sem RTS/CTS. Sem nodos escondidos.*

Com estes resultados constatou-se que sem nodos escondidos, os tempos médios de transferência do PC1 e PC2 são mais altos com RTS/CTS do que sem RTS/CTS mas curiosamente no caso do PC3 e *Android* foram mais baixos. No caso dos primeiros dois, isto explica-se pela introdução de overhead desnecessário, já que não há situação de nodos escondidos. No caso do PC3 e *Android* a explicação é tentada no próximo parágrafo. Em relação à variância e desvio padrão, nota-se que com RTS/CTS os valores são bastante mais baixos do que sem RTS/CTS, com a excepção do dispositivo *Android*.

Estas diferenças de valores nos dispositivos explicam-se pelo facto de os computadores serem todos diferentes, e consequentemente usarem diferentes tecnologias e potências de antena. A explicação mais plausível é a que o PC1 e PC2 são dois computadores portáteis normais, que

apesar de serem de marcas diferentes devem ser aproximadamente da mesma gama, daí os seus valores serem relativamente próximos. O PC3 é um *notebook eeepc*, que tem características como potência da antena bastante diferentes dos dois anteriores, e o telemóvel *Android* pela mesma razão, daí os seus valores serem diferentes dos outros dois computadores.

Esta primeira ronda de testes foi efectuada numa hora em que se encontravam várias pessoas no Departamento de Informática a utilizar várias redes sem fios com muitos canais sobrepostos, daí poder haver alguma interferência nas comunicações. Apesar disto, alterou-se o canal do ponto de acesso para um que tivesse menos sobreposições com outras redes.

A ideia geral dos testes da segunda ronda foi ver se é possível de alguma forma notar melhorias nas taxas de transmissão quando o RTS/CTS é activado nos clientes que estão em situação de nodos escondidos, razão pela qual o mecanismo foi criado. Para isso foram colocados novamente os dois computadores afastados um do outro, garantindo que o seu raio de alcance não se intercepta, mas de forma a conseguirem comunicar com o ponto de acesso. Nestes testes apenas foram usados dois computadores para conseguir controlar melhor a situação de nodos escondidos e pela facilidade de activação e desactivação do RTS/CTS em ambiente *Windows*.

Para poder ter alguma noção de melhorias nas taxas de transmissão foram medidos os tempos de transmissão por FTP de um ficheiro de 28MB. Depois de várias repetições dos testes com o mecanismo RTS/CTS activado e desactivado nos clientes, surgiram os seguintes resultados, apresentados na Tabela 4.2:

	Com RTS/CTS		Sem RTS/CTS	
	PC1	PC2	PC1	PC2
exec.1	1.42	1.27	1.43	1.51
exec.2	1.37	1.23	1.48	1.58
exec.3	1.32	1.26	1.45	1.44
exec.4	1.09	1.26	1.10	1.20
exec.5	1.22	1.17	1.18	1.29
média	1.28	1.24	1.33	1.40
variância	0.02	0	0.03	0.02
desvio padrão	0.13	0.04	0.17	0.16

Tabela 4.2: *Tempos de transferência em minutos, com e sem RTS/CTS. Com nodos escondidos.*

Com estes dados pode concluir-se que realmente as melhorias são notórias quando o mecanismo RTS/CTS é activado. O tempo médio de transferência diminuiu com a activação do RTS/CTS, sendo mais notório no caso do PC2. Apesar de essa diminuição parecer não muito significativa, caso tivesse sido utilizado um ficheiro maior, pensa-se que as diferenças seriam maiores e mais significativas. Pode também verificar-se melhorias observando a variância e desvio padrão, sendo que com o RTS/CTS activado os tempos de transferência não estão tão dispersos, principalmente no caso do PC2, que é o computador que está mais afastado do ponto de acesso.

Capítulo 5

Conclusões

Ao longo deste trabalho foram abordados vários aspectos relacionados com a gestão e monitorização de redes sem fios. Com o estudo das tecnologias SNMP e NETCONF foi possível ter uma visão mais aprofundada de como os protocolos funcionam e como contribuem para a gestão e monitorização de redes sem fios. No estudo comparativo entre as duas tecnologias, foi determinado que o protocolo SNMPv3 não é muito utilizado por parte dos provedores de serviço na configuração de equipamento, devido à sua complexidade operacional. Os estudos disponíveis não são muito conclusivos, focando-se apenas nalgumas funcionalidades ou características. Não há um estudo completo e exaustivo que ajude a determinar qual dos dois protocolos é melhor. Provavelmente nenhum deles, já que depende muito de outras tecnologias implementadas na rede bem como o tipo de monitorização e gestão da rede. O SNMP foi o protocolo escolhido para desenvolver a aplicação deste trabalho porque é suportado muito mais amplamente por equipamentos de rede sem fios, nomeadamente as MIBs *IEEE802dot11-MIB*, *CISCO-DOT11-IF-MIB* e *CISCO-DOT11-ASSOCIATION-MIB*, e permite monitorização efectiva, ao contrário do NETCONF que só permite configuração.

No estudo das MIBs SNMP implementadas em pontos de acesso conseguiu-se obter um maior nível de conhecimento sobre o que se passa nas camadas inferiores e todos os seus processos e técnicas. Este estudo possibilitou também a escolha dos objectos mais indicados para fazer monitorização e configuração dos dispositivos, focando-se em objectos que indiquem a presença de nodos escondidos tais como número de retransmissões e número de erros de processamento.

Depois de analisar as várias propostas para atenuar o problema de nodos escondidos, optou-se pelo uso dinâmico do mecanismo RTS/CTS. Assim, evita-se o uso desnecessário do mecanismo quando não existem nodos escondidos na rede, bem como a introdução de *overhead* de forma permanente na rede. Contudo, em vez das duas opções disponíveis, RTS/CTS activado para todos os pacotes ou RTS/CTS com o tamanho de pacotes por omissão, teria

sido interessante poder controlar o valor do tamanho do pacote (*RTS Threshold*) de forma mais granular.

A aplicação desenvolvida mostrou-se capaz de desempenhar as tarefas pretendidas, possibilitando ainda a monitorização de outros objectos que sejam necessários em trabalhos futuros que possam vir a seguir esta linha de investigação.

Nos testes iniciais foi possível detectar o aumento da média de erros no ponto de acesso, possível indicativo da presença de nodos escondidos. Com isto é garantido que definindo um valor limite, a aplicação responderia com a activação do mecanismo RTS/CTS. Contudo essa activação apenas foi possível no ponto de acesso e não nos terminais ou clientes. Estes, que seriam os nodos escondidos é que teriam cada um deles de activar o mecanismo RTS/CTS. Mas devido ao facto de não ser possível monitorizar e configurar os objectos das MIBs pretendidas nos terminais, nem alterar automaticamente o valor de *RTS Threshold* para activação do mecanismo RTS/CTS, esses testes tiveram de ser efectuados manualmente.

Foi assim possível verificar que numa rede em que estejam presentes nodos escondidos, a activação do mecanismo RTS/CTS traz melhorias ao seu rendimento, consequência do menor número de colisões de pacotes.

A principal conclusão do trabalho agora apresentado foi que é possível construir ferramentas para gestão automática de redes sem fios através de tecnologias normalizadas como o SNMP. Sendo assim, seria muito útil que os clientes ou terminais de redes sem fios possam ser monitorizados e reconfigurados através dessas mesmas tecnologias, quer para auto-reconfiguração como para gestão remota.

5.1 Trabalho Futuro

O ambiente de testes não foi o ideal havendo outros canais em sobreposição. Caso fosse possível realizar testes em condições de pouca ou nenhuma interferência seria esperado *throughput* mais constante e uma distribuição mais homogénea nos tempos de transferência. Outro aspecto que poderia trazer resultados mais claros e conclusivos seria a utilização de computadores iguais, com as mesmas potências de antena e características rádio. Os testes poderiam ter sido mais exaustivos, com mais computadores ligados ao ponto de acesso, de forma a simular um ambiente mais realista. Assim haveria mais tráfego em circulação, adicionando carga no ponto de acesso e talvez assim, os sinais de presença de nodos escondidos seriam mais evidentes.

Para finalizar, o programa desenvolvido neste trabalho está pronto a monitorizar qualquer

agente SNMP, seja ele num ponto de acesso ou idealmente num computador. Para isso bastaria correr em *background* o programa individualmente em cada um dos computadores para poder activar o mecanismo RTS/CTS dinamicamente. Outra hipótese seria o uso do programa como sistema centralizado de monitorização. Desde que o programa saiba os endereços *IP* dos clientes, a monitorização e activação adequada do mecanismo RTS/CTS seria garantida. Para tal, apenas seria necessário recolher informação dos clientes associados, nomeadamente o endereço *IP* através da tabela presente na MIB CISCO-DOT11-ASSOCIATION-MIB. A lista de endereços seria populada a partir desta tabela, e seria criada uma *thread* de monitorização/configuração para cada um dos clientes.

Referências Bibliográficas

Bierman, A. and Bjorklund, M. 2012. Network configuration protocol (netconf) access control model. *IETF RFC 6536*. 82

Biri, A. and Afifi, H. 2008. A novel protocol for securing wireless internet service provider's hotspots. 6

Boroumand, L., Khokhar, R. H., Bakhtiar, L. A., and Pourvahab, M. 2012. A review of techniques to resolve the hidden node problem in wireless networks. xiii, 28

Calhoun, P., Cisco Systems, I., Montemurro, M., Motion, R. I., Stanley, D., and Networks, A. 2009a. Control and provisioning of wireless access points (capwap) protocol binding for ieee 802.11. *RFC 5416*. 8

Calhoun, P., Cisco Systems, I., Montemurro, M., Motion, R. I., Stanley, D., and Networks, A. 2009b. Control and provisioning of wireless access points (capwap) protocol specification. *RFC 5415*. 8

Calhoun, P., Suri, R., Cam-Winget, N., Cisco Systems, I., Williams, M., Sciences, G. A. ., Hares, S., Hara, B. O., and S.Kelly. 2010. Lwapp: Lightweight access point protocol. *RFC 5412*. 8

Callon, R., Corker, K., Nodine, M., Ong, J., Stillman, M., and Westcott, J. 2009. Disciplines for effective network management. 15

Chadha, R. 2004. Applications of policy-based network management. 6

Chen, Y. and Vukovic, I. 2007. An rts-on-demand mechanism to overcome self-interference in an 802.11 system. xiii, 31, 32, 33, 34, 35, 36, 37

Choi, W.-Y. 2008. Clustering algorithm for hidden node problem in infrastructure mode ieee802.11 wireless lans. 30

Clemm, A. 2006. Network management fundamentals. *Cisco Press*. 13

Enns, R., Bjorklund, M., Schoenwaelder, J., and Bierman, A. 2011. Network configuration protocol (netconf). *IETF RFC 6241*. 18

Ferro, G. 2013. DisponÃvel em marÃ§o de 2013: http://etherealmind.com/. 18, 19, 20

Gast, M. S. 2002. 802.11 wireless networks - the definitive guide. xiii, 29, 88, 91

Hedstrom, B., Watwe, A., and Sakthidharan, S. 2011. Protocol efficiencies of netconf versus snmp for configuration management functions. xiii, 20, 21, 22

Iijima, T., Kimura, H., Atarashi, Y., and Higuchi, H. 2012. Netconf over websocket. *IETF*. 84

Kim, M. and Choi, C.-H. 2012. Hidden node detection in ieee 802.11n wireless lans. 31

Li, H. and Chen, G. 2006. Wireless lan network management system. 7

Mauro, D. and Schmidt, K. 2005. Essential snmp. 11

Mjidi, M., Chakraborty, D., Nakamura, N., Koide, K., Takeda, A., and Shiratori, N. 2008. A new dynamic scheme for efficient rts threshold handling in wireless networks. 31

Mjidi, M., Chakraborty, D., Nakamura, N., and Shiratori, N. 2005. The impact of dynamic rts threshold adjustment for ieee 802.11 mac protocol. 35

Pang, F. 2002a. Cisco-dot11-association-mib : Cisco dot11 association mib file. *Disponivel [Julho 2013] em ftp://ftp.cisco.com/pub/mibs/v2/CISCO-DOT11-ASSOCIATION-MIB.my*. 24

Pang, F. 2002b. Cisco-dot11-if-mib : Cisco ieee 802.11 interface mib file. *Disponivel [Julho 2013] em ftp://ftp.cisco.com/pub/mibs/v2/CISCO-DOT11-IF-MIB.my*. 24

Pang, F. 2002c. Ieee802dot11-mib : Ieee 802.11 management information base file. *Disponivel [Julho 2013] em ftp://ftp.cisco.com/pub/mibs/v2/IEEE802dot11-MIB.my*. 24

Parsons, K. 2013. Wi-fi stress test - a vendor-independent access point analysis. *Disponivel [Julho 2013] em http://www.wlanpros.com/wi-fi-stress-test-a-vendor-independent-access-point-analysis/*. xiii, 55, 56

Pras, A., Drevers, T., van de Meent, R., and Quartel, D. A. C. 2004. Comparing the performance of snmp and web services-based management. 20

Raghavendra, R., Acharya, P., Belding, E., and Almeroth, K. 2008. Antler: A multi-tiered approach to automated wireless network management. 5

Schoenwaelder, J., Perelman, V., Ersue, M., and Watsen, K. 2012. Network configuration protocol light (netconf light). *IETF*. 86

Scott, M. and Bjorklund, M. 2010. Yang module for netconf monitoring. *IETF RFC 6022*. 80

Shigeyasu, T., Akimoto, M., and Matsuno, H. 2011. Throughput improvement of ieee802.11 dcf with adaptive rts/cts control on the basis of existence of hidden terminals. 30

Sridhar, T. 2006. Wireless lan switches: Functions and deployment. *Cisco: The Internet Protocol Journal 9,* 3. 9, 10

Sun-Mi, Y., Ju, H. T., and Hong, J. W. 2006. Performance improvement methods for netconf-based configuration management. 20

Weinstein, K., Wang, W., Peters, K., Gelman, D., and Dimarogonas, J. 2011. A domain-level data model for automating network configuration. 6

Yaacob, A., Tan, I., Chien, S. F., and Tan, H. K. 2010. Arima based network anomaly detection. 6

Yan, B. and Chen, G. 2009. Model-based fault diagnosis for ieee 802.11 wireless lans. 6

Yang, Y. and Ma, M. 2008. A novel contention-based mac protocol with channel reservation for wireless lans. 30

Yen, L.-H. and Yeh, T.-T. 2006. Snmp-based approach to load distribution in ieee 802.11 networks. 7

Yu, J. and Ajarmeh, I. A. 2010. An empirical study of the netconf protocol. 20

Zhao, Z.-G. 2008. A hierarchy management framework for automated network fault identification. 5

Zhu, H., Lin, X., Ho, P.-H., Shen, X., and Shi, M. 2007. Ttp based privacy preserving inter-wisp roaming architecture for wireless metropolitan area networks. 7

Apêndice A

Network Configuration Protocol

Neste apêndice são apresentados os resultados detalhados sobre o estudo do protocolo NETCONF.

O protocolo pode ser dividido conceptualmente em quatro camadas, conforme a figura seguinte:

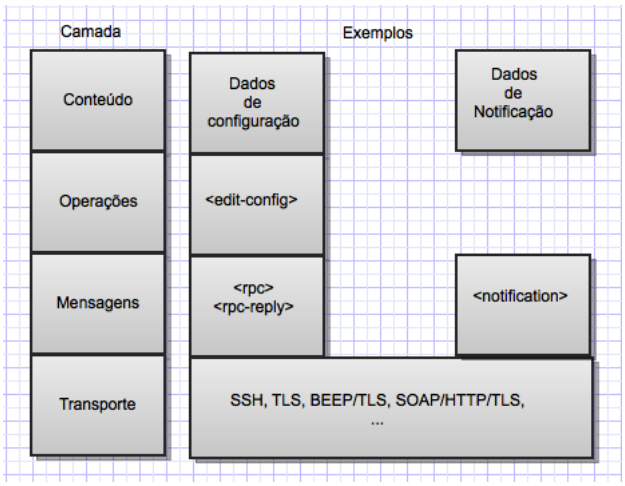


Figura A.1: *Camadas conceptuais do protocolo NETCONF*

A camada de transporte cria uma ligação segura entre o servidor e o cliente. O NETCONF pode correr sobre qualquer protocolo de transporte, desde que respeite um conjunto básico de requisitos. A camada de mensagens tem presente mecanismos para codificação de RPCs e notificações. Na camada seguinte, operações, é definido um conjunto de operações invocadas como métodos RPC com parâmetros codificados em XML. A camada de conteúdo é onde são representados os modelos de dados, implementados em YANG.

Numa mensagem NETCONF, codificada em XML, o elemento <rpc> delimita um pedido enviado do cliente para o servidor. Esse elemento tem um campo obrigatório denominado

message-id, de modo a ser possível manter uma sequência de mensagens e também para poder fazer a correspondência com os <rpc-reply> provenientes de uma resposta do receptor do pedido. De seguida é mostrado um exemplo muito simples de uma mensagem RPC em que o campo *message-id* é 101, e é chamado um método *my-own-method* com parâmetros <my-first-parameter> e <another-parameter> com valores respectivos de 14 e *fred*.

```
<rpc message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <my-own-method xmlns="http://example.net/me/my-own/1.0">
    <my-first-parameter>14</my-first-parameter>
    <another-parameter>fred</another-parameter>
  </my-own-method>
</rpc>
```

Neste momento, se a mensagem RPC chegou devidamente ao servidor, este irá responder com outra mensagem RPC mas agora com o elemento <rpc-reply>, obviamente com o mesmo *message-id* para que seja possível manter uma conversação e responder aos pedidos apropriados. Agora é mostrada uma sequência de mensagens RPC <rpc> e <rpc-reply>. Neste exemplo é feito um pedido <get> que inclui informação adicional sobre um identificador de utilizador. Na resposta esta informação é enviada em conjunto com o conteúdo pedido.

```
<rpc message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:ex="http://example.net/content/1.0"
     ex:user-id="fred">
  <get/>
</rpc>

<rpc-reply message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:ex="http://example.net/content/1.0"
     ex:user-id="fred">
  <data>
    <!-- contents here... -->
  </data>
</rpc-reply>
```

Presentes nas mensagens RPC codificadas em XML estão ainda os elementos <rpc-error>. Este elemento é incluído na resposta <rpc-reply> se ocorreu um erro durante o processamento de um pedido RPC, no qual é explicado o tipo de erro conforme a camada

onde ocorreu, uma etiqueta de erro onde é identificada a condição de erro, e a gravidade do erro, conforme seja um erro, ou apenas um aviso. Está incluído ainda uma descrição breve do erro e os conteúdos do erro específicos em relação ao protocolo ou modelo de dados. Por fim, é também usado o elemento <ok> que é usado para informar que o pedido foi processado sem erros nem avisos e nenhuma informação foi devolvida como resultado da operação. De seguida é ilustrado o exemplo da ocorrência de um erro provocado pela falta do atributo *message-id* no elemento <rpc>.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
  </get-config>
</rpc>

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rpc-error>
    <error-type>rpc</error-type>
    <error-tag>missing-attribute</error-tag>
    <error-severity>error</error-severity>
    <error-info>
      <bad-attribute>message-id</bad-attribute>
      <bad-element>rpc</bad-element>
    </error-info>
  </rpc-error>
</rpc-reply>
```

A.1 Operações

Como já referido anteriormente, o protocolo NETCONF disponibiliza um conjunto de operações-base para gerir configurações de dispositivos e recolher informação de estado do dispositivo. De seguida são apresentadas e descritas tais operações.

Operação <get-config>

Faz uma recolha parcial ou total de uma configuração. Recebe como parâmetros a configuração-fonte, onde a informação será recolhida, e um elemento <filter> para filtrar informação no caso de recolhas parciais. Como resposta positiva, o servidor envia uma reposta <rpc-reply> contendo um elemento <data> com os resultados do pedido. Caso contrário, um elemento

<rpc-error> é incluído na resposta. De seguida é mostrado um exemplo em que é pedida toda a informação presente em <users>.

```
<rpc message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://example.com/schema/1.2/config">
        <users/>
      </top>
    </filter>
  </get-config>
</rpc>

<rpc-reply message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <top xmlns="http://example.com/schema/1.2/config">
      <users>
        <user>
          <name>root</name>
          <type>superuser</type>
          <full-name>Charlie Root</full-name>
          <company-info>
            <dept>1</dept>
            <id>1</id>
          </company-info>
        </user>
        <!-- additional <user> elements appear here... -->
      </users>
    </top>
  </data>
</rpc-reply>
```

Esta operação carrega uma configuração ou partes desta, numa *datastore* específica. Permite que uma nova configuração seja representada de diferentes formas, sejam elas em ficheiro local ou ficheiro remoto. Se a *datastore* não existir, é criada com esta operação. O dispositivo analisa as configurações fonte e destino e executa as mudanças desejadas. A

configuração destino não é necessariamente substituída como em <copy-config>, mas sim alterada de acordo com a configuração fonte onde foi especificado e pedido.

Os elementos da configuração podem conter o atributo *operation*, que identifica o ponto da configuração onde a operação <edit-config> deve ser executada. Caso este atributo não seja especificado, a configuração fonte é simplesmente fundida com a configuração destino. O atributo *operation* pode tomar diferentes valores como por exemplo, *merge*, *replace*, *create*, *delete* e *remove*.

Recebe como parâmetros a configuração destino a ser editada, como por exemplo, <running/> ou <candidate/>, e o parâmetro opcional <default-operation> que pode tomar os valores de *merge*, *replace* e *none*. Existe ainda o elemento <test-option> que só pode ser usado se o dispositivo suportar a aptidão *:validate*, e pode tomar um dos seguintes valores apresentados de seguida. *test-then-set* executa um teste de validação da configuração antes de fazer o *set* no dispositivo, e caso ocorra algum erro, a operação <edit-config> não é executada. É também possível executar com o valor *set*, em que não é feita qualquer validação. Em último caso, é ainda possível usar o valor *test-only* em que, como o nome indica, apenas é executado o teste sem a tentativa de executar o *set* da configuração.

Finalmente, e também como elemento opcional relativo a erros de configuração, existe o *error-option* que por sua vez pode tomar os valores seguintes. Se for escolhido *stop-on-error*, a operação <edit-config> é abortada no momento em que é encontrado o primeiro erro. Com *continue-on-error* se forem encontrados erros, o processamento da configuração não é interrompido, sendo o erro gravado e gerada uma resposta negativa no final. Uma opção interessante é a *rollback-on-error* onde, em caso de erro, o servidor interrompe a operação <edit-config> e restaura a configuração anterior, ou seja, antes do início da operação. Contudo, esta opção necessita da aptidão *:rollback-on-error*. Como resposta positiva, se o dispositivo foi capaz de satisfazer o pedido, é enviada uma <rpc-reply> contendo o elemento <ok>. Caso contrário, é enviada uma resposta <rpc-error>.

Para exemplificar esta operação é editado o valor de MTU para 1500 numa interface com o nome *Ethernet0/0* na configuração corrente.

```
<rpc message-id="101"
      xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <edit-config>
     <target>
       <running/>
     </target>
     <config>
```

```
        <top xmlns="http://example.com/schema/1.2/config">
          <interface>
            <name>Ethernet0/0</name>
            <mtu>1500</mtu>
          </interface>
        </top>
      </config>
    </edit-config>
  </rpc>

  <rpc-reply message-id="101"
       xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

Outro exemplo é o de remoção da interface *Ethernet0/0* da configuração corrente:

```
<rpc message-id="101"
       xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <edit-config>
      <target>
        <running/>
      </target>
      <default-operation>none</default-operation>
      <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
        <top xmlns="http://example.com/schema/1.2/config">
          <interface xc:operation="delete">
            <name>Ethernet0/0</name>
          </interface>
        </top>
      </config>
    </edit-config>
  </rpc>

  <rpc-reply message-id="101"
       xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
  </rpc-reply>
```

Operação <copy-config>

Esta operação cria ou substitui uma configuração na sua totalidade, com o conteúdo de outra configuração existente. Se a configuração destino existe é substituída, caso contrário, é criada uma nova configuração. Se a aptidão *:url* é suportada, é possível usar o elemento <url> como parâmetro fonte ou destino. No entanto, mesmo que a aptidão *:writable-running* seja suportada, o dispositivo pode optar por não usar a configuração corrente como destino da operação <copy-config>.

O dispositivo pode ainda optar por não suportar operações de cópia remota, onde as configurações fonte e destino são elementos <url>, isto é justificado no caso de os parâmetros fonte e destino conterem o mesmo URL ou configuração. Nesse caso, um erro deve ser devolvido contendo a mensagem de valor inválido. Recebe como parâmetros a configuração destino, e a configuração fonte. Tal como as operações anteriores, a resposta positiva é dada através da inclusão do elemento <ok> na resposta <rpc-reply>. De seguida é exemplificada uma operação deste tipo, com a respectiva resposta.

```
<rpc message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <copy-config>
    <target>
      <running/>
    </target>
    <source>
      <url>https://user:password@example.com/cfg/new.txt</url>
    </source>
  </copy-config>
</rpc>

<rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

Operação <delete-config>

Esta operação elimina na totalidade uma configuração, à excepção da configuração corrente (<running/>) que não pode ser eliminada. Tal como em <copy-config>, também pode ser usado como parâmetro um elemento <url> caso a aptidão apropriada seja suportada, a aptidão *:url*. Logicamente, numa operação deste tipo, o único parâmetro é a configuração a

eliminar. Como resposta positiva ao processamento da operação, é enviada uma <rpc-reply> com o elemento <ok>. Já como resposta negativa é incluído um elemento <rpc-error> em <rpc-reply>.

Operação <lock>

A operação <lock> permite ao cliente (gestor) bloquear uma configuração de um dispositivo na totalidade. Esta operação normalmente não é efectiva durante muito tempo, sendo apenas usada para situações de concorrência no acesso à configuração pretendida, em que outro cliente tente também fazer alterações na configuração. A tentativa de bloqueio de uma configuração deve falhar quando outra sessão ou entidade já executou o bloqueio.

Recebe como parâmetro obrigatório a configuração a bloquear. As respostas positivas e negativas são as mesmas das outras operações. Um bloqueio não deve ser activado se outra sessão ou actividade já o tiver adquirido, se a configuração a bloquear é a <candidate>, tendo sido modificada e essas modificações ainda não foram enviadas, ou no último caso, se a configuração a bloquear é a <running> e outra sessão encontra-se a efectuar o envio.

Operação <unlock>

Logicamente, uma operação de desbloqueio será necessária, sendo usada para a configuração poder voltar a ser usada por outras sessões ou entidades. Esta operação não é executada com sucesso caso o bloqueio especificado não se encontra activo ou se a sessão que executa a operação <unlock> não é a mesma que obteve o bloqueio.

Operação <get>

Operação usada para recolher a configuração corrente ou informação de estado de um dispositivo. Recebe como parâmetro um filtro, o elemento <filter>, para poder especificar mais detalhadamente a informação a recolher. Se o filtro não for especificado, é devolvida toda a configuração e informação de estado.

Operação <close-session>

Esta operação simplesmente faz o pedido de terminação de uma sessão NETCONF. No momento em que esta operação é processada pelo servidor, a sessão é terminada de forma elegante ao contrário da operação <kill-session> que força a terminação da mesma. No decorrer da operação, o servidor liberta quaisquer bloqueios que estejam activos, bem como recursos associados com a sessão.

Operação <kill-session>

É forçada a terminação de uma sessão NETCONF. Uma entidade NETCONF que receba este pedido, irá interromper imediatamente todas as operações em curso, libertar bloqueios e recursos associados com a sessão, e fecha todas as ligações associadas.

A.2 Aptidões (Capabilities)

Conforme já dito anteriormente, o protocolo NETCONF tem a possibilidade de estender as suas funcionalidades básicas, através do uso de aptidões ou em inglês *capabilities*. Nesta secção do documento é feita uma análise mais detalhada dessas aptidões. Cada um dos elementos de rede NETCONF tem de saber quais as aptidões que os outros podem usar. Para isto, durante o estabelecimento de uma ligação, cada dispositivo (cliente e servidor) deve comunicar as aptidões que tem disponíveis. Isto é feito através de uma mensagem com o elemento <hello> que contém uma lista com as aptidões suportadas.

Logicamente, deve suportar pelo menos as funcionalidades básicas, e no caso de suporte a várias versões, deverá indicá-las todas. Uma outra condição que deve ocorrer para permitir o funcionamento correcto do protocolo é que os dispositivos que vão enviar a mensagem hello, não devem esperar por outros dispositivos, mas sim, enviar logo que a ligação seja estabelecida. De notar que a implementação de algumas aptidões introduz novas operações, ou altera o funcionamento de operações já existentes. Agora são apresentadas algumas aptidões presentes no RFC 6241.

Writable-Running Capability

Esta aptidão permite efectuar alterações directas à configuração corrente do dispositivo, ou seja, é permitido executar as operações <edit-config> e <copy-config> à configuração <running>. As operações <edit-config> e <copy-config> são alteradas, sendo agora permitido usar a configuração <running> como <target> das operações.

Candidate Configuration Capability

Com esta funcionalidade adicionada ao dispositivo é possível utilizar uma terceira configuração para além da <running> e da <startup>, a <candidate> que é usada para manter dados de configuração que podem ser alterados sem interferir com a configuração corrente do dispositivo. Para além disso, é ainda possível com a operação <commit> enviar esta configuração para funcionar como configuração corrente.

O cliente pode rejeitar quaisquer mudanças que não tenham sido enviadas, através da operação <discard-changes>, que substitui o conteúdo de <candidate> com o conteúdo de <running>. Ambas as operações <commit> e <discard-changes> são adicionadas com esta aptidão, bem como alterado o funcionamento das operações <get-config>, <edit-config>, <copy-config>, <validate>, <lock> e <unlock> que passam a poder usar como parâmetro a configuração <candidate>

Confirmed Commit Capability

Esta aptidão permite ao servidor suportar a operação <cancel-commit> e adição dos parâmetros <confirmed>, <confirm-timeout>, <persist>, e <persist-id> na operação <commit>. No caso de uma operação de <commit> confirmada, se a confirmação não é recebida antes do período de tempo limite, a operação deve ser revertida. Por omissão, o valor de tempo limite é de 600 segundos, mas pode ser ajustado com o parâmetro <confirm-timeout>. Se ocorrer um novo <commit> e o anterior ainda não foi confirmado, o tempo limite é reiniciado para o seu valor definido.

Já no caso de ocorrer um envio confirmado mas o elemento <persist> não estar presente, qualquer envio posterior deve ser feito na mesma sessão que o primeiro, até ser confirmado. Se o elemento <persist> estiver presente, os envios posteriores podem ser efectuados em qualquer sessão, mas devem incluir o elemento <persist-id> com um valor igual ao do elemento <persist>, de forma a identificar a quem enviar a confirmação. Para cancelar um envio confirmado, e reverter as alterações sem ter de esperar pelo tempo limite, o cliente pode restaurar a configuração para o seu estado anterior através do uso da operação <cancel-commit>.

Rollback-on-Error Capability

Com esta aptidão, o servidor passa a suportar o valor **rollback-on-error** no parâmetro <error-option> da operação <edit-config>. Tal como o nome indica, permite que durante a alteração de uma configuração se ocorrer algum erro, esta é restaurada para o ponto imediatamente antes da tentativa de alteração. Em configurações partilhadas esta funcionalidade pode criar situações indesejadas, como por exemplo, o restauro causado por um erro ser efectuado ao mesmo tempo que outro elemento da rede esteja a alterar a configuração. Neste caso, as alterações do segundo são revertidas. Para evitar este tipo de situações é recomendado o uso de bloqueio da configuração enquanto operações são efectuadas.

Validate Capability

O processo de validação consiste em verificar se uma configuração contém erros sintácticos ou semânticos, antes de aplicá-la a um dado dispositivo. Com a adição desta aptidão é garantido o suporte da operação <validate> que verifica pelo menos erros de sintaxe, bem como o suporte do parâmetro <test-option> na operação <edit-config>. Este parâmetro pode conter o valor *test-only*, que valida uma configuração, mesmo que não tenha sido enviada.

Distinct Startup Capability

Com esta adição, o dispositivo suporta *datastores* separadas para a configuração <startup> e <running>. A configuração <startup> é carregada pelo dispositivo na inicialização, e operações que afectariam a configuração <running> não são automaticamente copiadas para a configuração <startup>. Para conseguir isto, é necessário executar a operação <copy-config> de forma a actualizar o conteúdo de <startup> com o conteúdo de <running>.

URL Capability

Com esta aptidão, é dada a possibilidade a um dispositivo NETCONF de aceitar o elemento <url> nos parâmetros <source> e <target>, na operação <copy-config> por exemplo. Outro exemplo é a operação <edit-config>, que é alterada para que o elemento <url> seja usado em alternativa ao elemento <config>. Com isto, é possível utilizar uma configuração remota, sendo acessível por HTTP, FTP, ou ficheiro.

XPath Capability

Quando esta aptidão é anunciada, indica que o dispositivo NETCONF suporta o uso de expressões Xpath no elemento <filter>, tipicamente usado em operações do tipo <get>. A expressão Xpath deve devolver um conjunto de nodos. Se isto não ocorrer, a operação falha com o erro *invalid-value*. A mensagem de resposta contém sub-árvores seleccionadas pela expressão de filtragem. Com isto, as operações <get> e <get-config> são modificadas de forma a aceitar o valor <xpath> no atributo *type* do elemento <filter>.

A.3 Monitorização e outras considerações

Tendo em vista o estudo comparativo a ser realizado, convém analisar dois documentos, um relativo à monitorização (RFC 6022) e outro relativo ao modelo de controlo de acesso (RFC 6536), na medida em que estes trazem funcionalidades que podem ser tão boas ou

melhores do que as já existentes em SNMP.

No primeiro, intitulado *YANG Module for NETCONF Monitoring*, é definido um modelo de dados com a finalidade de monitorizar o protocolo NETCONF. Esse modelo de dados inclui informação sobre armazenamento de dados, sessões, *locks* e estatísticas, dados esses, indispensáveis à gestão de um servidor NETCONF. São ainda definidos métodos para descoberta do modelo de dados suportado por um servidor NETCONF e definida a operação <get-schema> para recolher os dados.[Scott and Bjorklund, 2010]

Para efectuar a monitorização existe um conjunto de dados de estado do dispositivo. Esses dados são representados numa árvore *netconf-state*, sendo essa a raiz da árvore. Como sub-nodos existem cinco grupos principais de dados, conforme representado na figura seguinte:

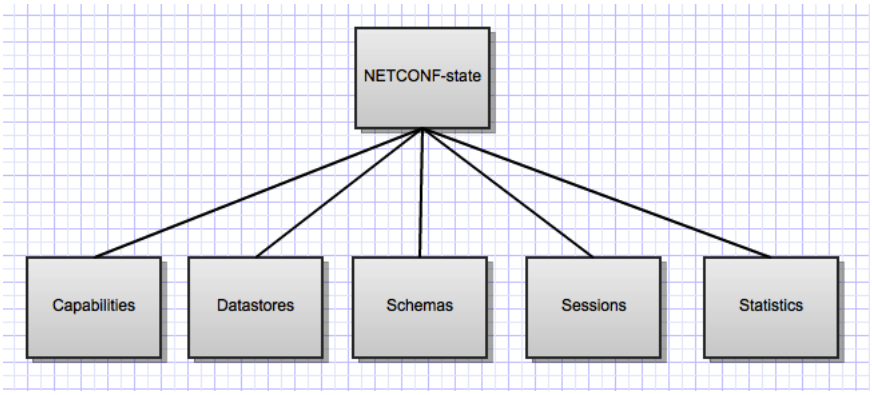


Figura A.2: *Diagrama da árvore monitorização NETCONF*

A sub-árvore *Capabilities* contém uma lista das aptidões suportadas pelo servidor (dispositivo). Na sub-árvore *Datastores*, está presente uma lista de configurações, como por exemplo <running>, <startup> e <candidate>, suportadas pelo dispositivo, bem como informação relativa a estas mesmas configurações. Em *Schemas* tal como nas sub-árvores anteriores, é incluida informação necessária para identificar os *schemas* e permitir a sua recolha. Em *Sessions*, são listadas todas as sessões NETCONF activas no dispositivo, incluindo contadores para cada uma delas. No sub-nodo *Statistics* são armazenados contadores globais do dispositivo NETCONF.

Procede-se agora a uma análise mais minuciosa desta estrutura de dados, percorrendo cada uma das sub-árvores, de forma a conhecer melhor o conteúdo de cada uma delas.

- ***Capabilities***
  Conforme dito no parágrafo anterior, esta sub-árvore contém as aptidões suportadas

pelo servidor NETCONF. Esta lista inclui todas a informação de aptidões trocadas durante o estabelecimento da sessão.

- ***Datastores***
  Nesta secção não há muito mais a acrescentar a não ser o facto de para cada configuração e respectivo nome, é também armazenada informação sobre se está bloqueada ou não (*locks*), na forma de uma lista.

- ***Schemas***
  Nesta sub-árvore é guardada a informação relativa aos *schemas*. Os dados principais são o identificador, a versão, formato, *namespace* XML e localização. O identificador para além da sua função, é usado na operação <get-schema>. O formato contém um valor que representa a linguagem de modelação de dados usada, como por exemplo, *xsd*, *yang*, *yin*, *mg* e *rnc*.

- ***Sessions***
  Aqui é armazenada informação relativa a sessões NETCONF. Existe um identificador de sessão *session-id* e o tipo de transporte da sessão, no campo *transport*. Para identificação de clientes autenticados pelo protocolo de transporte é guardada a string *username* bem como o endereço IP associado ao cliente. É ainda guardada informação relativa à data e hora de *login*, número de mensagens RPC correctamente recebidas, números de erros em mensagens RPC de entrada e saída e número de notificações enviadas.

- ***Statistics***
  Dados estatísticos e contadores do servidor NETCONF populam esta sub-árvore. Quando o sistema é iniciado, é guardada a data e hora em *netconf-start-time*. É contado o número de sessões iniciadas, bem como o número de sessões canceladas devido a mensagens *hello* inválidas, e sessões interrompidas devido a *timeout* ou fecho de transporte. São também armazenados todos os contadores relativos à sessão, já descritos no parágrafo *Sessions*.

Com este documento torna-se possível que o NETCONF caminhe no sentido de colmatar a falta de mecanismos de monitorização de rede e de igualar ou até superar na tarefa em que o SNMP é tão utilizado e conhecido por isso. Com a liberdade de desenvolvimento dada pelo uso do XML em vez de MIBs que de certa forma são mais restritivas, o NETCONF tem grandes hipóteses de se tornar por excelência no protocolo de gestão de redes.

O segundo documento para análise é o RFC 6536, com o título *Network Configuration Protocol (NETCONF) Access Control Model*. O propósito deste documento é a criação de mecanismos de controlo de acesso, de modo a criar restrições de diferentes níveis a quem efectua operações sobre as configurações e o seu conteúdo, criando assim um ambiente seguro para

as tarefas de configuração. Estes mecanismos devem ser bem estruturados e usados de forma segura e simples, permitindo ainda interoperabilidade entre diferentes fabricantes. A arquitectura dos servidores NETCONF deve ser protegida em três pontos conceituais. Operações de protocolo, onde são necessárias permissões para invocar operações específicas. Nas configurações, onde deve haver permissões para ler ou alterar dados. Por fim, nas notificações em que é preciso autorização para receber tipos de eventos específicos.[Bierman and Bjorklund, 2012]

Um aspecto importante deste mecanismo é a sua facilidade de utilização. Se for demasiado complicado de usar e configurar, provavelmente não será muito bem aceite, e por consequência disso não será utilizado amplamente. Então, é sugerido que as tarefas simples sejam fáceis de executar, não sendo necessário quase nenhum conhecimento na área. Em relação às tarefas mais complexas, apesar de simplificadas, já é necessário algum conhecimento adicional e até usar outros mecanismos. Logicamente também deve ser possível que esse controlo de acesso permita configurar regras de acesso tanto para um utilizador, como para um grupo de utilizadores, suportando o conceito de grupos administrativos e distinguindo claramente uma conta de administrador e outros tipos de utilizadores menos privilegiados.

Para tudo isto funcionar, é necessária a correspondência entre utilizadores e grupos, que pode ser feito num servidor RADIUS. O mecanismo deve ser também capaz de desactivar parcialmente ou totalmente os procedimentos de controlo de acesso, sem eliminar quaisquer regras já existentes. Isto será usado por exemplo em situações de manutenção. No que diz respeito ás configurações, deve estar presente o controlo de acesso no momento da execução de, por exemplo, uma operação <edit-config>. Devem também existir regras para restringir o acesso a determinadas sub-árvores das configurações existentes.

No âmbito do controlo de acesso nas configurações (*datastores*), foi criado um modelo de direitos de acesso, o CRUDX, e contempla todas as operações do protocolo NETCONF, Create, Read, Update, Delete e eXec. A título de exemplo são analisadas algumas das operações NETCONF. Nas operações <get> e <get-config>, os nodos a que o cliente não tem acesso são omitidos na mensagem de resposta <rpc-reply>. Isto é feito para permitir que os filtros funcionem como é suposto, em vez de causarem um erro de acesso não permitido, que de outra forma, daria acesso não autorizado de leitura a alguns nodos. No que diz respeito à filtragem em NETCONF, o critério de selecção é aplicado a um subconjunto de nodos que o utilizador está autorizado a ler, e não a configuração inteira.

Em operações <edit-config>, os direitos de acesso aplicam-se a todas as operações que resultariam num acesso em particular a configuração destino. Se o acesso efectivo da operação é *none*, (operação por omissão, *none*), então nenhum controlo de acesso será aplicado a esse nodo. Isto é necessário para permitir acesso a uma sub-árvore dentro de uma grande

estrutura. Por exemplo, um utilizador pode ter autorização para criar uma nova entrada em */interfaces/interface* mas não ter autorização para criar ou eliminar o seu nodo pai, */interfaces*. Se */interfaces* já existir na configuração destino, então, a operação efectiva será *none* para */interfaces* no caso de uma entrada em */interfaces/interface* ser editada.

Se o resultado efectivo da operação resultar na criação, remoção ou actualização de um nodo, e o utilizador não tiver as permissões adequadas para esse nodo, a operação é rejeitada com um erro de acesso não permitido. Em operações de fusão e substituição (*merge* e *replace*), pode incluir nodos que não alterem partes da configuração existente. Estes nodos são ignorados pelo servidor e não necessitam de direitos de acesso por parte do cliente. Finalmente, os conteúdos de nodos específicos não devem ser expostos em nenhum elemento <rpc-error> na resposta.

Em operações do tipo <copy-config>, é necessário ter em atenção o facto de como resultado desta, o administrador pode substituir uma configuração inteira. Se a configuração origem da operação for a <running> e a configuração destino for a <startup>, o cliente apenas necessita de permissão para executar a operação <copy-config>. Contudo existem outros casos em que a situação não é tão simples. Se o cliente não tem acesso de leitura a certos nodos, estes são omitidos. Também, se a operação resultar na criação, remoção ou actualização de nodos, e o cliente não tiver permissões de acesso, a operação é rejeitada com um erro de acesso não permitido.

No caso de remoção de uma configuração com a operação <delete-config>, o acesso não é permitido por omissão. As regras de controlo de acesso devem ser explicitamente configuradas para permitir a sua invocação. Numa operação <commit> o servidor deve ser capaz de determinar os nodos que diferem da configuração corrente, e verificar as permissões de criação, actualização e remoção apenas para esses nodos. Por exemplo, se uma sessão tem permissões para ler toda a configuração mas apenas permissão para editar um nodo, essa sessão deve ser capaz de editar e submeter apenas esse nodo.

Por fim, uma operação em que é importante ter controlo de acesso é a operação <kill-session>. Esta operação, não altera directamente a configuração, no entanto é possível que uma sessão interrompa uma segunda sessão que esteja a editar uma configuração, o que não é desejável. O acesso a esta operação não é permitido por omissão, tal como a operação <delete-config>, também as regras de controlo de acesso devem ser configuradas explicitamente.

A.4 Propostas em estado experimental

Também algumas propostas interessantes surgiram muito recentemente no grupo de trabalho do NETCONF, ainda em estado experimental. Como já referido anteriormente, de momento é obrigatório implementar NETCONF sobre SSH. Apesar disto, continua a haver novas sugestões para concretizar a camada de transporte do protocolo. Um desses exemplos é o documento *NETCONF over WebSocket* de T. Iijima, onde é obviamente sugerido que seja usado o protocolo WebSocket. Tendo em conta o aumento do uso de computação em *cloud*, têm aparecido sistemas de gestão de rede que suportam interfaces baseadas em navegadores web.

O problema é que a maior parte dos protocolos de gestão de rede não suportam o transporte via HTTP. Apesar da tentativa do RFC 4743 (NETCONF sobre SOAP), em que foi definido o transporte via SOAP com HTTPS, esta não foi vista como uma boa solução porque o HTTP não preenchia os requisitos das especificações do NETCONF para notificações, que exigem bi-direccionalidade. Para colmatar esta falha, o uso de WebSocket faz com que o protocolo possa ser implementado para funcionar na *cloud*, através de um navegador web.

O protocolo WebSocket foi desenhado para ser implementado em navegadores e servidores web, mas pode ser usado por qualquer aplicação do paradigma cliente-servidor. A comunicação entre estes elementos é feita de forma a permitir situações em que o servidor queira enviar dados para o cliente, sem que o cliente os tenha pedido, útil no caso das notificações NETCONF e permitir circulação de mensagens em ambos os sentidos, mantendo a conexão aberta. Assim, é possível uma conversação bi-direccional entre clientes e servidores, característica básica de um sistema de gestão de redes.

Não existe qualquer intenção de substituir o SSH como mecanismo de trasporte para NETCONF. Mesmo assim, no RFC 6241 (SSH) é especificado que o NETCONF pode ser transportado por qualquer protocolo que implemente um conjunto de funcionalidades. Essas funcionalidades são operação orientada à conexão e autenticidade, integridade e confidencialidade. Ora, o protocolo WebSocket implementa este conjunto de funcionalidades, sendo orientado à conexão e a sua autenticação é assegurada por mecanismos disponíveis para um servidor HTTP genérico, tais como *cookies*, *HTTP Authentication* ou TLS. A integridade e confidencialidade são garantidas como o uso de WebSocket sobre TLS. [Iijima et al., 2012]

De momento, já existem implementações de servidores WebSocket, como por exemplo Jetty e Kaazing. O mesmo pode ser dito dos respectivos clientes, com implementação nos navegadores Chrome e Firefox. Existem ainda bibliotecas para clientes WebSocket, importantes para desenvolver aplicações sobre WebSocket. Como o protocolo disponibiliza uma API JavaScript para navegadores, é facilitado o desenvolvimento de clientes NETCONF para executar

troca de mensagens. Deste modo, um ficheiro HTML escrito com a API para receber e enviar mensagens NETCONF funciona como um cliente NETCONF no navegador.

Assim, este tipo de sistema de gestão não precisa de instalação num computador para o administrador efectuar as suas tarefas. Uma das vantagens disso é o facto de se poder gerir os dispositivos de rede em qualquer computador ou dispositivo com acesso a um navegador, como por exemplo um *tablet*. Todos estes factos são vantajosos tanto para os administradores de rede de uma operadora, como para os fabricantes.

Outra proposta que, apesar de ter expirado recentemente, é relativamente interessante no âmbito deste trabalho é o documento *Network Configuration Protocol Light (NETCONF Light)* com autoria de J. Schoenwaelder. É sugerida uma modularização do protocolo NETCONF que permite aos dispositivos anunciarem que apenas suportam um subconjunto das funcionalidades e/ou operações NETCONF. Isto é vantajoso em situações em que os dispositivos não possuem muitos recursos (processamento, memória, interfaces) para suportar o protocolo NETCONF na totalidade, como seria o caso em pontos de acesso *thin*. Outro ponto interessante desta modularização é o facto de ser possível fazer melhoramentos graduais dos dispositivos para suportarem NETCONF.

No contexto de um provedor de internet sem fios, com inúmeros pontos de acesso para gerir, faz todo o sentido optar por uma solução que use pontos de acesso *thin*, ou seja, o mais leves possíveis em termos de recursos locais, e a maior parte do processamento é efectuado em pontos centralizados. Com isto, a utilização de NETCONF Light torna-se muito atractiva. É também interessante do ponto de vista dos fabricantes porque é possível ir adicionando funcionalidades NETCONF gradualmente como se pode ver de seguida. Se um determinado dispositivo tem como objectivo implementar uma aptidão específica ao fabricante, apenas vai necessitar da camada de troca de mensagens do NETCONF, RPC, RPC-reply e notificações. Se um dispositivo apenas precisar de ler uma configuração, então só precisa de implementar a operação <get-config>. Claro que à medida que seja preciso, pode-se adicionar qualquer outro módulo em qualquer ponto no futuro.

Existem casos em que o fabricante ainda não desenvolveu certos módulos por falta de tempo, ou falta de interesse numa funcionalidade. Um desses casos seria um dispositivo em que é desejada a implementação completa, mas o fabricante ainda não terminou todas as operações do tipo <edit-config>. Neste caso, o dispositivo pode apenas implementar a operação <copy-config> para poder pelo menos copiar uma configuração. Outro caso seria a não implementação das operações <lock> e <unlock>, devido á sua plataforma ainda não possuir nenhum mecanismo para fornecer essa funcionalidade. Assim, os dispositivos podem anunciar um subconjunto específico das operações que suporta.

É possível ainda limitar o número de sessões em concorrência tendo em vista a contenção de recursos. Caso já tenha sido atingido o número máximo de sessões, a implementação NETCONF Light rejeita o estabelecimento destas, não passando sequer pelo processo de troca de mensagens <hello>. Numa implementação NETCONF Light, pode-se optar por não suportar determinadas operações. Assim, quando essas operações são invocadas, mas não são suportadas, deve ser devolvida uma mensagem de erro com um elemento <rpc-error> com um valor <error-tag> contendo *operação não suportada.*[Schoenwaelder et al., 2012]

Apêndice B

Estudo de MIBs de Pontos de Acesso IEEE 802.11

B.1 IEEE802dot11-MIB

O primeiro agrupamento desta MIB é o *dot11SMTbase*, em que fazem parte objectos relacionados com o funcionamento básico de uma estação (STA), permitindo o funcionamento cooperativo como elemento de uma rede 802.11. Aqui estão presentes os seguintes objectos:

```
dot11StationID
dot11MediumOccupancyLimit
dot11CFPollable
dot11CFPPeriod
dot11CFPMaxDuration
dot11AuthenticationResponseTimeOut
dot11PrivacyOptionImplemented
dot11PowerManagementMode
dot11DesiredSSID
dot11DesiredBSSType
dot11OperationalRateSet
dot11BeaconPeriod
dot11DTIMPeriod
dot11AssociationResponseTimeOut
```

- ***dot11StationID***
  Obviamente, este objecto identifica uma estação, através do seu endereço MAC.

- ***dot11MediumOccupancyLimit***
  Este objecto só é utilizado em estações implementem acesso sem contenção, com a técnica PCF (*Point Coordination Function*). Indica o período de tempo máximo em unidades de tempo (TU - 1024 microsegundos) que um ponto de coordenação (*point*

coordinator) pode controlar o uso do meio sem fios sem ceder o controlo, de forma a permitir pelo menos uma instância de acesso DCF (*Distributed Coordination Function*) ao meio. Por outras palavras, este objecto dita número de unidades de tempo em que há acesso sem contenção ao meio. Aumentando este valor aumenta-se a capacidade alocada ao serviço sem contenção. Diminuindo o valor, reduz-se o tempo disponível para o serviço sem contenção. [Gast, 2002]

- ***dot11CFPollable, dot11CFPPeriod e dot11CFPMaxDuration***
  Como estes três objectos são relacionados com o CFP (*Contention Free Period*) são analisadas em conjunto. O objecto *dot11CFPollable* do tipo booleano, tomando o valor de verdadeiro, prepara a estação para responder a *CF-Poll* com uma trama dentro de um intervalo SIFS (*Short Interframe Space*). Já o objecto *dot11CFPPeriod* representa o número de intervalos DTIM (*Delivery Traffic Indication Message*) entre períodos sem contenção, visto que esses períodos começam sempre com uma mensagem DTIM. Estas mensagens fazem parte da trama de sinalização (*Beacon Frame*), portanto o tempo entre o começo de períodos sem contenção pode ser calculado através da multiplicação do valor do período CFP pelo intervalo DTIM. Por fim, o objecto *dot11CFPMaxDuration* dita a duração máxima do período sem contenção em unidades de tempo, no momento da criação de um BSS (*Basic Service Set*). Este valor pode ser gerado pelo ponto de coordenação PCF. [Gast, 2002]

- ***dot11BeaconPeriod e dot11DTIMPeriod***
  Estes dois objectos também estão relacionados com os três anteriores, tendo uma relação mais próxima com *dot11CFPPeriod*. O objecto *dot11BeaconPeriod* especifica a duração do intervalo de sinalização (*Beacon Interval*) em unidades de tempo. O objecto *dot11DTIMPeriod* simplesmente indica o número de intervalos de sinalização entre transmissões DTIM. Ambos os valores mencionados são transmitidos nas tramas de sinalização e de resposta a sonda (*Beacon Frame e Probe Response*). [Gast, 2002]

- ***dot11AuthenticationResponseTimeOut e dot11AssociationResponseTimeOut***
  A finalidade de estes dois objectos é fundamentalmente a mesma, mas para processos diferentes. O objecto *dot11AuthenticationResponseTimeOut* indica o intervalo de tempo em que cada passo da autenticação de uma estação pode demorar, antes de ser reconhecida uma falha na autenticação. Analogamente, o objecto *dot11AssociationResponseTimeOut* tem a mesma função mas para os passos de associação de uma estação.

- ***dot11PrivacyOptionImplemented*** Com este objecto do tipo booleano é possível verificar se a opção WEP está implementada no dispositivo, e o valor pré-definido é falso. De sublinhar que não indica se a opção WEP está em uso ou não, mas apenas a sua implementação.

- ***dot11PowerManagementMode*** Este atributo especifica que modo de gestão de energia está em uso, para o valor (1) está activo, para o valor (2) está em modo de poupança de energia. Quando consultado por software de gestão, irá responder sempre com o valor de activo visto que o dispositivo necessita de estar ligado para enviar a trama de resposta. Um outro uso para a consulta deste objecto pode ser para determinar com que regularidade a estação está activa.

- ***dot11DesiredSSID*** Durante o processo de sonda as estações podem ser configuradas para procurar por uma rede em específico, a qual é identificada pela seu SSID (Service Set ID). Este valor pode ser configurado de forma a uma estação associar-se preferencialmente com tal rede.

- ***dot11DesiredBSSType*** Tal como o objecto anterior, este objecto também é usado para configurar o processo de sonda. Conforme desejado, a estação pode ser "forçada"a associar-se a uma rede em infra-estrutura com o valor 1, uma BSS (Basic Service Set) independente com o valor 2, ou qualquer BSS com o valor 3. Isto é normalmente utilizado para filtragem de tramas de sinalização e de resposta a sonda (*Beacon Frame e Probe Response*).

- ***dot11OperationalRateSet*** Este atributo indica o conjunto de taxas de dados que a estação deve transmitir dados, de forma a verificar as taxas de dados suportadas no processo de associação. Ao contrário do que acontece mais recentemente, estes valores eram incrementos de 500 kbps, entre 1 Mbps to 63.5 Mbps, com 127 valores no máximo. Com padrões actuais, estes valores são simplemente usados como etiquetas porque o intervalo de 1 a 127 apenas permite uma taxa máxima de 63.5 Mbps.

Por razões explicadas de seguida, é feita uma quebra na ordem dos grupos. O grupo anteriormente analisado, o *dot11SMTbase* contém objectos que também fazem parte de outros grupos, nomeadamente *dot11SMTbase2* e *dot11SMTbase3*. Assim, para evitar a repetição de análise individual de objectos, são apresentados os objectos de cada um dos dois novos grupos sendo feita apenas análise individual aos objectos que não estão presentes no grupo *dot11SMTbase*. Procede-se então à análise do grupo *dot11SMTbase2*.

```
dot11MediumOccupancyLimit
dot11CFPollable
dot11CFPPeriod
dot11CFPMaxDuration
dot11AuthenticationResponseTimeOut
dot11PrivacyOptionImplemented
dot11PowerManagementMode
dot11DesiredSSID
dot11DesiredBSSType
```

```
dot11OperationalRateSet
dot11BeaconPeriod
dot11DTIMPeriod
dot11AssociationResponseTimeOut

dot11DisassociateReason
dot11DisassociateStation
dot11DeauthenticateReason
dot11DeauthenticateStation
dot11AuthenticateFailStatus
dot11AuthenticateFailStation
```

Como é possível verificar, apenas os últimos seis objectos, diferem do grupo anterior. De referir que estes seis objectos fazem parte do grupo de notificações da MIB *dot11Notification-Group*. Então novamente para evitar repetições no documento, é aqui analisado em conjunto o grupo de notificações.

- ***dot11DisassociateReason e dot11DisassociateStation***
  Estes objectos pertencem ao sub-grupo de notificações *dot11Disassociate*. O primeiro contém o código de razão (*Reason Code*) transmitido mais recentemente numa trama de dissociação. Estes códigos servem para indicar a razão pela qual a operação, neste caso a dissociação, foi efectuada com sucesso ou insucesso. O segundo objecto contém o endereço MAC da estação que transmitiu mais recentemente uma trama de dissociação. Usando estes dois objectos em conjunto, um administrador de rede pode verificar que estação foi expulsa da rede com a justificação correspondente.

- ***dot11DeauthenticateReason e dot11DeauthenticateStation***
  Presentes no sub-grupo de notificações *dot11Deauthenticate*, estes dois objectos têm funções similares às dos dois objectos anteriores, com a diferença de serem aplicados à operação de desautenticação.

- ***dot11AuthenticateFailStatus e dot11AuthenticateFailStation***
  Finalmente, estes objectos fornecem informação sobre falhas de autenticação e fazem parte do sub-grupo de notificações *dot11AuthenticateFail*. O primeiro contém o código de estado (*Status Code*) da trama de falha de autenticação mais recente. O segundo objecto contém o endereço MAC da estação que transmitiu uma trama de autenticação mais recentemente.

Por fim, o grupo *dot11SMTbase3* é usado quando a estação tem capacidades de operações em multi-domínio. Da mesma forma que o grupo anterior, adicionalmente com este grupo surgem mais três objectos, que serão analisados de seguida.

- ***dot11MultiDomainCapabilityImplemented***
  Este objecto do tipo booleano, quando assume o valor de verdadeiro, indica que a implementação da estação consegue suportar múltiplos domínios. Obviamente, quando o valor é falso indica que não tem essa capacidade.

- ***dot11MultiDomainCapabilityEnabled***
  Objecto muito semelhante ao anterior, apenas com a diferença de indicar se a capacidade de suporte a múltiplos domínios está activada ou desactivada.

- ***dot11CountryString***
  Com este objecto é possível identificar o país a partir do qual a estação está a operar. Os primeiros dois octetos desta *string* correspondem ao código de dois caracteres descrito no documento ISO/IEC 3166-1. O terceiro octeto poderá tomar um dos seguintes valores. Um espaço ASCII, caso os regulamentos do país onde opera envolvem todos os ambientes, exteriores ou interiores. Um caracter ASCII 'O', caso os regulamentos sejam apenas para ambientes exteriores (*Outdoor*). Um caracter ASCII 'I', caso os regulamentos sejam apenas para ambientes interiores (*Indoor*).

Outro grupo essencial ao funcionamento de um ponto de acesso sem fios é o *dot11SMTprivacy* que logicamente cobre os aspectos relacionados com privacidade e segurança das comunicações. De uma forma mais específica, este grupo contém um conjunto de objectos essenciais na implementação de WEP na estação. Mas antes de continuar convém esclarecer um aspecto sobre este grupo. Na MIB existem duas tabelas de armazenamento de chaves WEP, a *dot11WEPDefaultKeysTable* e a *dot11WEPKeyMappings*. A primeira é relativamente simples. Cada interface pode ter até um máximo de quatro chaves por omissão associadas, devido ao facto de a especificação WEP permitir quatro chaves por omissão em cada rede. Algo mais complexa, a segunda tabela tira partido do facto de a WEP suportar o uso de uma chave diferente por cada endereço MAC existente. As chaves podem então ser mapeadas num par único de endereço de origem e endereço de destino. Para cada interface no sistema, um certo número de pares endereço-chave podem ser associados com essa interface. Cada interface usa um índice auxiliar de forma a identificar todos os endereços MAC associados com chaves e adicionalmente, informação relativa a cada endereço.[Gast, 2002] Com isto em conta, são apresentados agora os objectos pertencentes a este grupo e seguidamente é feita uma breve descrição dos mesmos.

```
dot11PrivacyInvoked
dot11ExcludeUnencrypted
dot11WEPICVErrorCount
dot11WEPExcludedCount
dot11WEPDefaultKeyID
dot11WEPDefaultKeyValue
dot11WEPKeyMappingWEPOn
```

```
dot11WEPKeyMappingLength
dot11WEPKeyMappingValue
dot11WEPKeyMappingAddress
dot11WEPKeyMappingStatus
```

- ***dot11PrivacyInvoked***
  Este atributo indica através de um valor booleano se o 802.11 WEP é o mecanismo usado para transmissão de tramas do tipo *Data*.

- ***dot11ExcludeUnencrypted***
  Este objecto do tipo booleano, quando toma o valor de verdadeiro a estação não deve informar a interface de serviço MAC sobre a chegada de MDSUs (*MAC Service Data Unit*) que têm o sub-campo WEP *Frame Control* com valor igual a zero. De forma inversa, quando o valor deste objecto é falso, a estação pode aceitar MSDUs que tenham o sub-campo WEP *Frame Control* com valor igual a zero. Com isto, é possível excluir as tramas que não estejam encriptadas ou, por outro lado, aceitá-las.

- ***dot11WEPICVErrorCount***
  Este objecto do tipo contador, deverá incrementar o seu valor quando uma trama é recebida com o sub-campo WEP *Frame Control* contém o valor 1 e o valor do ICV (*Integrity Check Value*) recebido não é consistente com o valor ICV calculado através dos conteúdos da trama. De uma forma muito básica, este objecto indica o número de insucessos no processo de desencriptação de uma trama.

- ***dot11WEPExcludedCount***
  Este contador incrementa o seu valor quando é recebida uma trama com o sub-campo WEP *Frame Control* com valor igual a zero e o valor de *dot11ExcludeUnencrypted* faz com que essa trama seja descartada.

- ***dot11WEPDefaultKeyID e dot11WEPDefaultKeyValue***
  Com estes dois objectos é possível fazer o mapeamento numa tabela de chaves WEP, fazendo-os corresponder directamente como identificador e chave. Essa tabela é denominada na MIB por *dot11WEPDefaultKeysTable*

- ***dot11WEPKeyMappingWEPOn e dot11WEPKeyMappingLength***
  O primeiro objecto indica se o mecanismo WEP está a ser utilizado quando existe comunicação com a estação referida em *dot11WEPKeyMappingAddress*. O segundo indica simplesmente o número máximo de tuplos que podem estar presentes na tabela *dot11WEPKeyMappings*.

- ***dot11WEPKeyMappingAddress e dot11WEPKeyMappingValue***
  O primeiro objecto contém o endereço MAC da estação para a qual os valores do mapeamento serão usados, enquanto que o segundo contém simplesmente o valor de uma chave WEP.

- ***dot11WEPKeyMappingStatus***
  Este objecto está representado na tabela *dot11WEPKeyMappings* por forma de uma coluna que é usada para criar, modificar e eliminar instâncias de objectos.

O próximo grupo a ser analisado é o *dot11MACbase*. Neste grupo estão presentes objectos para assegurar o devido suporte da camada MAC a controlo de acesso, geração e verificação de FCSs (*Frame Check Sequence*), bem como a entrega apropriada de dados válidos às camadas superiores. De seguida são apresentados e analisados em maior detalhe esses objectos.

```
dot11MACAddress
dot11RTSThreshold
dot11ShortRetryLimit
dot11LongRetryLimit
dot11FragmentationThreshold
dot11MaxTransmitMSDULifetime
dot11MaxReceiveLifetime
dot11ManufacturerID
dot11ProductID
dot11Address
dot11GroupAddressesStatus
```

- ***dot11MACAddress***
  Este objecto identifica uma estação pelo seu endereço MAC. Por omissão, é um endereço globalmente único, atribuído pelo fabricante. No entanto, este pode ser alvo de uma mudança forçada por um administrador de rede, caso seja necessário.

- ***dot11RTSThreshold***
  Quaisquer tramas de dados ou gestão maiores que o limite RTS (*Request To Send*) devem ser transmitidas usando a negociação RTS/CTS (*Request To Send / Clear To Send*). Por omissão, o valor deste limite é de 2347 octetos, tendo o efeito de desactivar a desobstrução RTS/CTS antes da transmissão. Se o valor for zero, é activada a negociação RTS/CTS antes de qualquer transmissão de qualquer tamanho de pacote.

- ***dot11ShortRetryLimit e dot11LongRetryLimit***
  O primeiro objecto indica o número máximo de tentativas de transmissão de uma trama, caso essa trama seja menor do que o limite RTS. Tramas curtas podem ser retransmitidas até esse limite antes de ser detectada uma condição de falha e serem identificadas como pertencentes a protocolos de camadas superiores. O segundo objecto tem a mesma funcionalidade, mas para tramas maiores do que o limite RTS.

- ***dot11FragmentationThreshold***
  Este objecto especifica o tamanho máximo em octetos de um MPDU (*MAC Protocol*

Data Unit). Um MSDU deverá ser dividido em fragmentos se o seu tamanho exceder o valor deste objecto, incluindo cabeçalhos e *trailers* MAC. Um MSDU ou MMPDU (MAC Management Protocol Data Unit) deverá ser fragmentado quando a trama resultante contém um endereço individual no campo *Address1* e o comprimento desta trama é maior do que o especificado neste objecto.

- ***dot11MaxTransmitMSDULifetime***
  Este objecto representa em unidades de tempo, o intervalo de tempo em que uma estação pode tentar transmitir uma trama. Depois deste intervalo, todas as tentativas de transmissão deverão ser canceladas.

- ***dot11MaxReceiveLifetime***
  Este objecto indica o intervalo de tempo depois da recepção de um fragmento inicial, em que são aceites outros fragmentos. Depois deste intervalo de tempo quaisquer tentativas de reconstrução de tramas a partir de fragmentos deverão ser canceladas.

- ***dot11ManufacturerID e dot11ProductID***
  Estes dois objectos são utilizados para identificar o ponto de acesso em questão. O primeiro identifica o nome do fabricante e poderá conter informação adicional conforme o desejo do fabricante. O segundo contém o nome do produto que é único para cada fabricante, podendo também incluir informação adicional.

- ***dot11Address e dot11GroupAddressesStatus***
  Estes dois objectos fazem parte da tabela *dot11GroupAddressesTable* que contém um conjunto de endereços MAC que identificam os endereços *multicast* a partir dos quais essa estação deverá receber tramas. O primeiro objecto contém simplesmente um endereço MAC que identifica endereços *multicast*. O segundo objecto é utilizado para criar, modificar e eliminar instâncias de objectos nessa tabela.

Agora serão abordados dois grupos em conjunto, visto terem fins idênticos. O grupo *dot11CountersGroup*, são contadores de implementação obrigatória que não estão presentes no segundo grupo. O grupo *dot11MACStatistics* como o próprio nome indica, fornece informação estatística sobre as operações na camada MAC. De salientar que este grupo é opcional, visto que não está directamente relacionado com o correcto funcionamento de um ponto de acesso sem fios, sendo no entanto, de enorme valor para um administrador de sistemas. De seguida são apresentados os objectos presentes nestes dois grupos.

```
dot11TransmittedFragmentCount
dot11MulticastTransmittedFrameCount
dot11FailedCount
dot11ReceivedFragmentCount
dot11MulticastReceivedFrameCount
```

```
dot11FCSErrorCount
dot11TransmittedFrameCount
dot11WEPUndecryptableCount

dot11RetryCount
dot11MultipleRetryCount
dot11FrameDuplicateCount
dot11RTSSuccessCount
dot11RTSFailureCount
dot11ACKFailureCount
```

- ***dot11TransmittedFragmentCount***
  Este objecto é incrementado por cada fragmento *unicast* ou *multicast* (dados ou gestão) confirmados. Tramas que não tenham sido fragmentadas, mas podem ser transmitidas sem fragmentação também podem fazer com que este contador incremente o seu valor.

- ***dot11MulticastTransmittedFrameCount***
  De forma similar ao objecto anterior, este objecto é incrementado de cada vez que uma trama *multicast* é enviada. Contudo, ao contrário do anterior, não é necessária a confirmação da trama para incrementar o contador.

- ***dot11FailedCount***
  Este contador deverá incrementar quando um MSDU não é transmitido com sucesso, devido ao número de tentativas de transmissão ter excedido o valor de um dos objectos *dot11ShortRetryLimit* ou *dot11LongRetryLimit*. Uma situação em que este contador poderá incrementar bastante será numa rede com carga de tráfego elevada.

- ***ReceivedFragmentCount e dot11MulticastReceivedFrameCount***
  O primeiro objecto incrementa o seu valor por cada MPDU do tipo dados ou gestão recebido com sucesso. Da mesma forma, o segundo objecto é incrementado quando é recebido com sucesso uma trama multicast.

- ***dot11FCSErrorCount***
  A função deste objecto é contabilizar o número de erros FCS detectados num MPDU recebido. Isto acontece de cada vez que o cálculo de verificação da trama (FCS) falha, e este indicador pode ser monitorizado para verificar o estado de integridade de uma rede.

- ***dot11TransmittedFrameCount***
  Este objecto deverá ser incrementado por cada MSDU transmitido com sucesso. A definição do sucesso de uma trama enviada é a recepção de confirmação.

- ***dot11WEPUndecryptableCount***
  Este objecto é incrementado de cada vez que uma trama recebida indica que está encriptada com WEP, mas não é possível desencriptar. Normalmente, em estações que não implementem WEP numa rede em que tramas WEP são usadas, este contador incrementa muito rapidamente. Isto pode também acontecer quando o mapeamento de chaves é inválido ou a chave por omissão é incorrecta.

- ***dot11RetryCount e dot11MultipleRetryCount***
  Estes dois contadores incrementam o seu valor em situações semelhantes, com a diferença de o primeiro incrementar com tramas recebidas após ser pedida a sua retransmissão e o segundo incrementar quando foi necessário mais que uma retransmissão.

- ***dot11FrameDuplicateCount***
  Este contador incrementa o seu valor por cada trama recebida com o campo *Sequence Control* a indicar que é uma duplicação. A situação de tramas duplicadas surge quando as confirmações não são recebidas. De forma a poder fazer uma estimativa do número de confirmações perdidas. Então, este contador incrementa quando é recebida uma trama duplicada.

- ***dot11RTSSuccessCount e dot11RTSFailureCount***
  O primeiro objecto é incrementado quando um CTS é recebido em resposta a um RTS, indicando que o meio está livre para transmitir. Em contraste, o segundo objecto é incrementado quando não é recebido um CTS em resposta a um RTS, indicando que o meio está ocupado para transmissão.

- ***dot11ACKFailureCount***
  Por fim, este contador é incrementado quando uma confirmação (ACK) não é recebida, mas era esperada.

Os objectos do grupo *dot11ResourceTypeID* são usados para simplesmente identificar uma estação através do seu fabricante, nome de produto e versão. Conforme o formato seguido ao longo deste capítulo, são apresentados os objectos e seguidamente analisados em detalhe.

```
dot11ResourceTypeIDName
dot11manufacturerOUI
dot11manufacturerName
dot11manufacturerProductName
dot11manufacturerProductVersion
```

- ***dot11ResourceTypeIDName***
  Este objecto contém o nome do identificador do tipo de recurso. Neste valor, apenas pode ser efectuada a operação de leitura, e contém sempre o valor RTID (*Resource*

Type ID). O valor deste objecto não deverá ser usado como identificador para qualquer outro objecto.

- ***dot11manufacturerOUI e dot11manufacturerName***
  O primeiro objecto contém o OUI (*Organizationally Unique Identifier*). Este identificador é único para cada fabricante. O segundo objecto é uma string utilizada para identificar o nome do fabricante do equipamento em questão.

- ***dot11manufacturerProductName e dot11manufacturerProductVersion***
  Estes dois objectos, como obviamente se pode concluir pelo nome identificam o equipamento pelo seu nome de produto, bem como a sua versão.

Outro grupo presente nesta MIB é o *dot11SmtAuthenticationAlgorithms*, do qual fazem parte dois objectos que constroem uma tabela de algoritmos de autenticação. Essa tabela é muito simples contendo apenas um índice por cada par de objectos, o *dot11AuthenticationAlgorithm* e o *dot11AuthenticationAlgorithmsEnable*. O primeiro contém os valores por omissão *"Open System"* e *"Shared Key"*, sendo possível adicionar outros algoritmos à tabela. O segundo, do tipo booleano, tomando o valor de verdadeiro activa a aceitação do algoritmo descrito na entrada correspondente na tabela, em tramas de autenticação recebidas pela estação. Obviamente caso o valor seja falso, o algoritmo usado é *"Open System"*, não tendo qualquer tipo de autenticação.

De seguida são reunidos na mesma seccção de análise três grupos da MIB relacionados com a gestão da camada PHY. O *dot11PhyOperationComplianceGroup* contém objectos relacionados com o funcionamento da camada PHY. Já o grupo *dot11PhyAntennaComplianceGroup* trata dos aspectos relacionados com antenas e respectivas taxas de dados para a norma IEEE 802.11. Por fim, no grupo *dot11PhyTxPowerComplianceGroup*, estão presentes objectos para controlo e gestão da potência de transmissão do equipamento. De seguida são listados os objectos presentes nestes três grupos, separados por uma linha em branco, e posteriormente analisados em maior nível de detalhe.

```
dot11PHYType
dot11CurrentRegDomain
dot11TempType

dot11CurrentTxAntenna
dot11CurrentRxAntenna
dot11DiversitySupport

dot11NumberSupportedPowerLevels
dot11TxPowerLevel1 -- dot11TxPowerLevel8
dot11CurrentTxPowerLevel
```

- ***dot11PHYType***
  Este objecto do tipo inteiro identifica o tipo PHY suportado pelo PLCP (*PHY Layer Convergence Procedure*) e PMD (Physical Medium Dependent). O valor de 8 bits pode corresponder aos tipos seguintes:
  FHSS (*Frequency Hopping Spread Spectrum*) 2.4 GHz = 01
  DSSS (*Direct Sequence Spread Spectrum*) 2.4 GHz = 02
  IR (Infra-Vermelhos) = 03
  OFDM (*Orthogonal Frequency Division Multiplexing*) 5GHz = 04
  HRDSSS (High-Rate Direct Sequence Spread Spectrum) = 05.

- ***dot11CurrentRegDomain***
  Este objecto indica o domínio regulamentar corrente da instância do PMD suportado. Corresponde a um dos *RegDomains* presentes em dot11RegDomainsSupported, que será um dos próximos grupos a analisar.

- ***dot11TempType***
  Com este objecto é possível descrever o intervalo da temperatura operacional do equipamento, visto que existem diferentes requisitos de temperaturas de operação conforme as condições ambientais de funcionamento do equipamento. Os valores e intervalos de temperatura possíveis são os seguintes. Tipo 1, com valor X'01', intervalo comercial dos 0 (zero) aos 40 graus centígrados. Tipo 2, com valor X'02', intervalo industrial dos -30 aos 70 graus centígrados.

- ***dot11CurrentTxAntenna e dot11CurrentRxAntenna***
  Estes dois objectos indicam a antena usada para transmitir e a antena usada para receber dados, respectivamente.

- ***dot11DiversitySupport***
  Este objecto fornece suporte para diversidade de antenas no equipamento. Isto quer dizer que é usada mais do que uma antena no equipamento, sendo possível melhorar a qualidade e fiabilidade do sinal através de várias técnicas, tais como diversidade espacial, diversidade de padrões, diversidade de polarização e diversidade de envio/recepção. Este objecto pode tomar os seguintes valores:
  X'01' - diversidade está disponível e é efectuada através de antenas presentes na lista *dot11DiversitySelectionRx*.
  X'02'- diversidade não é suportada.
  X'03'- diversidade é suportada bem como o seu controlo, e nesse caso o objecto *dot11DiversitySelectionRx* pode ser dinâmicamente alterado pelo LME (Layer Management Entity).

- ***dot11NumberSupportedPowerLevels***
  Este atributo representa o número de níveis de potência suportados pelo PMD e pode

ter um valor entre 1 e 8.

- ***dot11TxPowerLevel1 – dot11TxPowerLevel8***
  Estes objectos de níveis do 1 ao 8, indicam a potência de transmissão associada a cada nível de potência. Assim é possível mudar para um determinado nível de potência já criado.

- ***dot11CurrentTxPowerLevel***
  Com este objecto é possível verificar o nível de potência actual usado para transmitir dados. Em alguns casos, este valor é usado para determinar a sensibilidade do receptor a requisitos CCA (*Clear Channel Acessment*).

Novamente são reunidos mais três grupos da MIB, muito relacionados com os grupos anteriores. Esses grupos são os seguintes. O *dot11PhyRegDomainsSupportGroup* que contém objectos que especificam os domínios de regulamentos suportados. O *dot11PhyAntennasListGroup*, relacionado com as antenas suportadas pelo equipamento. Finalmente o *dot11PhyRateGroup* que contém objectos para suporte das taxas de dados. Tal como na secção anterior, os objectos são listados e separados por uma linha em branco, e se seguida analisados.

```
dot11RegDomainsSupportedValue

dot11SupportedTxAntenna
dot11SupportedRxAntenna
dot11DiversitySelectionRx

dot11SupportedDataRatesTxValue
dot11SupportedDataRatesRxValue
```

- ***dot11RegDomainsSupportedValue***
  Este objecto é utilizado em conjunto com um índice na tabela *dot11RegDomainsSupportedTable*. Como existem diferentes requisitos operacionais dependendo do domínio regulamentar, surge a necessidade de haver suporte para esses mesmos diferentes requisitos. Então, com este objecto os valores possíveis são:
  FCC (EUA) = X'10'
  DOC (Canada) = X'20'
  ETSI (maior parte da Europa) = X'30'
  Espanha = X'31'
  França = X'32'
  MKK (Japão) = X'40'

- ***dot11SupportedTxAntenna e dot11SupportedRxAntenna***
  Estes objectos estão presentes numa lista de antenas suportadas a *dot11AntennasListTable*. Uma antena pode ter atribuída uma funcionalidade específica como transmitir,

receber ou para participar em diversidade de recepção. Cada entrada na tabela representa uma única antena em conjunto com as suas propriedades. No caso do primeiro objecto, se o valor do booleano for verdadeiro, indica que a antena representada na tabela pode ser usada como antena de transmissão. No caso do segundo objecto, se o valor indicado for verdadeiro, quer dizer que a antena em questão pode ser utilizada como antena de recepção.

- ***dot11DiversitySelectionRx***
  Este objecto, do tipo booleano, quando o valor é verdadeiro indica que a antena pode ser usada para diversidade de recepção, e apenas poderá tomar esse valor caso a antena seja obviamente uma antena de recepção.

- ***dot11SupportedDataRatesTxValue e dot11SupportedDataRatesRxValue***
  Estes dois objectos, também presentes em tabelas, respectivamente a *dot11SupportedDataRatesTxTable* e a *dot11SupportedDataRatesRxTable*, representam as taxas de dados suportadas pelo PLCP e PMD do equipamento. Em ambas as tabelas, existe uma contagem de X'02 a X'7f, correspondentes a incrementos de taxas em 500kbps no intervalo de 1Mbps a 63.5Mbps, sujeitos a limitações do equipamento.

De seguida são analisados mais dois grupos no âmbito de uma das técnicas de modulação, a FHSS (*Frequency Hopping Spread Spectrum*) nos quais são definidos objectos de forma a haver cooperação com a tecnologia em questão. Os grupos analisados são o *dot11PhyFHSSComplianceGroup* cujos objectos tornam possível a configuração da técnica FHSS e o *dot11PhyFHSSComplianceGroup2* com a mesma tarefa, mas com a particularidade de ser específico para uso quando a opção de domínios múltiplos está implementada. Tal como em grupos anteriores existem objectos que estão presentes em grupos distintos, neste caso, os primeiros sete objectos pertencem a ambos os grupos, sendo os restantes pertencentes ao segundo grupo. Assim, estes dois grupos são apresentados com as suas variáveis e analisados em detalhe de seguida.

```
dot11HopTime
dot11CurrentChannelNumber
dot11MaxDwellTime
dot11CurrentDwellTime
dot11CurrentSet
dot11CurrentPattern
dot11CurrentIndex

dot11EHCCPrimeRadix
dot11EHCCNumberofChannelsFamilyIndex
dot11EHCCCapabilityImplemented
dot11EHCCCapabilityEnabled
```

`dot11HopAlgorithmAdopted`
`dot11RandomTableFlag`
`dot11NumberofHoppingSets`
`dot11HopModulus`
`dot11HopOffset`
`dot11RandomTableFieldNumber`

- ***dot11HopTime***
  Este objecto define o tempo em nanossegundos para o PMD mudar do canal 2 para o canal 80.

- ***dot11CurrentChannelNumber***
  Este objecto contém o número do canal actual da frequência de saída do sintetizador RF (Rádio-Frequência).

- ***dot11MaxDwellTime e dot11CurrentDwellTime***
  O primeiro objecto define o tempo máximo em unidades de tempo que um transmissor pode operar num canal. O segundo objecto define o tempo em que um transmissor pode operar num canal, tal como configurado na camada MAC, sendo o seu valor por omissão 19 TUs.

- ***dot11CurrentSet e dot11CurrentPattern***
  Estes dois objectos pouco diferem um do outro. O primeiro define o conjunto de padrões que o PLME (Physical Layer Management Entity) usa para determinar a sequência de salto entre canais. Já o segundo simplesmente indica o padrão em uso para determinar a sequência de salto entre canais.

- ***dot11CurrentIndex***
  Este objecto simplesmente indica o valor do índice que o PLME usa para determinar o número do canal actual.

- ***dot11EHCCPrimeRadix***
  Este objecto indica o valor a ser usado como raíz prima (N) nos algoritmos HCC (Hyperbolic Congruence Codes) e EHCC (Extended Hyperbolic Congruence Codes).

- ***dot11EHCCNumberofChannelsFamilyIndex***
  Este objecto representa o valor a ser usado como máximo do índice de família (a) nos algoritmos HCC e EHCC. O valor deste campo não deve ser menor que a raiz prima menos três (N - 3). Os valores permitidos são (N-1), (N-2) e (N-3).

- ***dot11EHCCCapabilityImplemented e dot11EHCCCapabilityEnabled***
  O primeiro objecto do tipo booleano, quando toma o valor de verdadeiro, indica que a implementação da estação consegue gerar os algoritmos HCC e EHCC para determinar

padrões de salto entre canais. Caso o valor seja falso, esta capacidade não está disponível. O segundo objecto tem exactamente a mesma função, mas em vez de determinar se está implementado ou não, simplesmente indica se está activado ou desactivado.

- ***dot11HopAlgorithmAdopted***
  Este objecto define quais os algoritmos que irão ser usados para gerar padrões de salto entre canais. Os valores válidos são os seguintes:
  1 - Padrões de salto definidos na cláusula 14 da MIB.
  2 - Método de salto por índice, como ou sem tabela.
  3 - Método HCC ou EHCC.

- ***dot11RandomTableFlag***
  Com este objecto booleano, usando o valor verdadeiro é possível indicar que uma tabela aleatória está presente. Quando o valor é falso, para além de indicar que uma tabela aleatória não está presente, indica também que o método de salto por índice será usado para determinar a sequência de salto.

- ***dot11NumberofHoppingSets***
  Este objecto determina o número total de conjuntos nos padrões de salto.

- ***dot11HopModulus e dot11HopOffset***
  O primeiro objecto representa o número de canais permitidos para o conjunto de saltos. Este valor é definido pelas regulamentações em vigor conforme o código do país no qual este equipamento opera. Já o segundo objecto, apenas indica a próxima posição no conjunto de saltos.

- ***dot11RandomTableFieldNumber***
  Este objecto é usado em conjunto com um índice na tabela dot11HoppingPatternTable, e define o número do canal inicial na sequência de saltos da sub-banda para o domínio associado.

Na continuação das técnicas de modulação, surgem os próximos dois grupos desta vez relacionados com a técnica DSSS (*Direct-Sequence Spread Spectrum*). Ambos os grupos tratam dos aspectos de configuração do equipamento, embora com alvos diferentes. O primeiro é usado para DSSS simples enquanto que o segundo é usado para HRDSSS (*High-Rate Direct-Sequence Spread Spectrum*). Tal como nos grupos anteriores, o segundo grupo contém os mesmos objectos que o primeiro mas adiciona outros objectos mais específicos para as suas funções. São então agora apresentados os grupos, seguido da análise dos seus objectos individualmente.

```
dot11CurrentChannel
dot11CCAModeSupported
dot11CurrentCCAMode
```

```
dot11EDThreshold

dot11ShortPreambleOptionImplemented
dot11PBCCOptionImplemented
dot11ChannelAgilityPresent
dot11ChannelAgilityEnabled
dot11HRCCAModeSupported
```

- ***dot11CurrentChannel***
  Este objecto representa o canal do PHY DSSS na frequência operacional.

- ***dot11CCAModeSupported***
  Este objecto pode tomar vários valores, representando os modos CCA suportados em PHY. De seguida são apresentados esses valores. Este valor não deve ser usado para indicar os modos CCA suportados em extensões PHY de alto débito. Para isso deverá ser usado o objecto *dot11HRCCAModeSupported* analisado mais à frente.
  ED-ONLY: 01 (apenas detecção de energia).
  CS-ONLY: 02 (apenas *carrier sense*).
  ED-and-CS: 04 (detecção de energia e *carrier sense*)

- ***dot11CurrentCCAMode***
  Este objecto informa qual o modo CCA em uso operacional. Para além dos modos do objecto anterior, é possível ainda usar mais dois, tendo então os seguintes modos:
  edonly: 01 (apenas detecção de energia).
  csonly: 02 (apenas *carrier sense*).
  edandcs: 04 (detecção de energia e *carrier sense*) cswithtimer: 08 (*carrier sense* com temporizador) hrcsanded: 16 (*carrier sense* de alto débito e detecção de energia)

- ***dot11EDThreshold***
  Com este objecto é definido o limite de detecção de energia a ser usado pelo PHY DSSS.

- ***dot11ShortPreambleOptionImplemented e dot11PBCCOptionImplemented***
  Estes dois objectos do tipo booleano indicam opções que podem ou não estar implementadas. O primeiro, tomando valor de verdadeiro indica que a opção de preâmbulo curto está implementada e disponível para utilização. O segundo, quando assume o valor de verdadeiro, indica que a opção de modulação PBCC (*Packet Binary Convolutional Code*) está implementada no equipamento.

- ***dot11ChannelAgilityPresent e dot11ChannelAgilityEnabled***
  Os próximos dois objectos são relacionados com a capacidade de agilidade de canal.

O primeiro indica que a camada PHY suporta esta capacidade, enquanto o segundo indica que a capacidade está activa. Obviamente, isto é válido para quando os objectos do tipo booleano tomam o valor de verdadeiro.

- ***dot11HRCCAModeSupported***
  Objecto idêntico a *dot11CCAModeSupported* cujo uso é mais indicado com PHY de alto débito. Os modos suportados são os seguintes:
  ED-ONLY: 01 (apenas detecção de energia).
  CS-ONLY: 02 (apenas *carrier sense*).
  ED-and-CS: 04 (detecção de energia e *carrier sense*)
  CS-and-Timer: 08 (*carrier sense* com temporizador)
  HRCS-and-ED(*carrier sense* de alto débito e detecção de energia)

Ainda nas técnicas de modulação usadas pela tecnologia IEEE 802.11, surge a OFDM (*Orthogonal Frequency-Division Multiplexing*). Para esta técnica de modulação existe um grupo dedicado à sua configuração, o *dot11PhyOFDMComplianceGroup*. De seguida são apresentados os objectos que fazem parte desse grupo.

```
dot11CurrentFrequency
dot11TIThreshold
dot11FrequencyBandsSupported
```

- ***dot11CurrentFrequency***
  Este objecto representa a frequência operacional do PHY OFDM.

- ***dot11TIThreshold***
  Este objecto indica o limite a ser usado para detectar ocupação do meio, por frequência. O CCA deverá acusar o meio como estando ocupado quando detectar o valor de RSSI (*Received Signal Strength Indication*) acima deste limite.

- ***dot11FrequencyBandsSupported***
  Este objecto representa a capacidade da implementação de PHY OFDM operar nas três bandas U-NII, sendo codificada como um valor inteiro em três campos de bit.
  Bit 0 - Opera na banda U-NII mais baixa (5.15-5.25 GHz).
  Bit 1 - Opera na banda U-NII média (5.25-5.35 GHz).
  Bit 2 - Opera na banda U-NII mais alta (5.725-5.825 GHz).
  Por exemplo, numa implementação que opere na banda baixa e média, o valor deste objecto seria 3.

Antes de terminar a análise desta MIB com o último grupo convém fazer referência ao grupo *dot11PhyIRComplianceGroup* que não é analisado aqui devido à sua quase inexistente utilização em redes sem fios da actualidade. Assim segue-se para a análise do último grupo, o *dot11MultiDomainCapabilityGroup*, que fornece suporte à gestão dos canais utilizáveis por uma estação quando a opção de múltiplos domínios está implementada.

```
dot11FirstChannelNumber
dot11NumberofChannels
dot11MaximumTransmitPowerLevel
```

- ***dot11FirstChannelNumber***
  Este objecto indica o número do canal de valor mais baixo na sub-banda para a *string* do país do domínio associada.

- ***dot11NumberofChannels***
  Este objecto indica o número total de canais permitidos na sub-banda para a *string* do país do domínio associada.

- ***dot11MaximumTransmitPowerLevel***
  Com este objecto é representado o valor máximo de potência de transmissão em dBm permitido na sub-banda para a *string* do país do domínio associada.

B.2 CISCO-DOT11-IF-MIB

Na continuação do estudo das MIBs surge a *CISCO-DOT11-IF-MIB* sendo uma extensão da *IEEE802dot11-MIB*, já abordada anteriormente. Ao contrário da MIB anterior, contém objectos proprietários, específicos para utilização em equipamento *Cisco*, facilitando o suporte a tecnologias desenvolvidas pela empresa, bem como melhoramentos à MIB original do IEEE. As diferenças ou novidades mais notórias são a compatibilidade de trabalhar com LANs virtuais (VLAN) e capacidade de monitorização remota. De seguida inicia-se o estudo pormenorizado desta MIB, seguindo os mesmos moldes da MIB anterior. Desta forma, os grupos serão apresentados individualmente com os seus respectivos objectos, sendo estes analisados em maior detalhe posteriormente.

O primeiro agrupamento desta MIB é o *cd11IfManagementGroup* sendo a sua nomenclatura uma abreviação de "*Cisco dot 11 interface*", tal como em todos os grupos e tabelas desta MIB. Este grupo é o mais extenso em termos de número de objectos. Estes objectos têm como função fornecer informação para suporte à gestão de interfaces IEEE 802.11. Contudo, devido a todos os objectos deste grupo pertencerem simultaneamente a outros grupos, a análise será feita aos 3 grupos separados, pois ao estar a fazer a análise a cada um destes grupos, automaticamente o grupo 1 estará a ser analisado. Assim, o grupo 1 é constituído por objectos que também fazem parte dos grupos 8, 9 e 10. De certa forma pode-se afirmar que estes três grupos são na verdade sub-grupos do grupo 1 devido a estes estarem relacionados com a gestão de interfaces, mais especificamente e respectivamente a gestão de atributos rádio, gestão de associação e gestão de SSID. Estes três grupos serão apresentados de seguida.

O primeiro destes três grupos é o grupo com índice 8 na MIB, o *cd11IfRadioManageGroup*. O

seu objectivo é fornecer informação para gestão de configurações de interfaces rádio. Os últimos 3 objectos apesar de não pertencerem ao grupo 1, são aqui apresentados por pertencer ao grupo 8.

```
cd11IfStationRole
cd11IfCiscoExtensionsEnable
cd11IfAllowBroadcastSsidAssoc
cd11IfPrivacyOptionMaxRate
cd11IfEthernetEncapsulDefault
cd11IfBridgeSpacing
cd11IfAuxiliarySsidLength
cd11IfVoipExtensionsEnable
cd11IfDesiredBssAddr
cd11IfAssignedSta

cd11IfWorldMode
cd11IfWorldModeCountry
cd11IfMobileStationScanParent
```

- ***cd11IfStationRole***
  Com este objecto é definido o papel da estação no BSS a que pertence através de um inteiro, podendo assim alterar a funcionalidade do equipamento conforme necessário. Para além do papel por omissão (ponto de acesso), os papéis que uma estação (ponto de acesso) pode desempenhar são os seguintes:
  roleWgb(1) - cliente de infra-estrutura (*Work Group Bridge*)
  roleBridge(2) - *root bridge*
  roleClient(3) - *bridge* cliente ou de *work group*
  roleRoot(4) - ponto de acesso *root*
  roleRepeater(5) - repetidor
  roleApBridge(6) - ponto de acesso e *root bridge*
  roleApRepeater(7) - ponto de acesso e repetidor
  roleIBSS(8) - BSS independente
  roleNrBridge(9) - *bridge* não *root*
  roleApNrBridge(10) - ponto de acesso e *bridge* não *root*
  roleScanner(11) - *scanner* para pontos de acesso e clientes ilegais ou suspeitos de actividades não permitidas.

- ***cd11IfCiscoExtensionsEnable***
  Este objecto do tipo booleano, quando assume o valor de verdadeiro, indica que as extensões *Cisco Aironet* estão activadas. Com estas extensões é possível obter melhor

performance no BSS, bem como *roaming* mais rápido entre pontos de acesso. Obviamente, se o valor deste objecto for falso, apenas o protocolo 802.11 básico é utilizado. Com isto, é assegurada a compatibilidade com equipamento de marca diferente.

- ***cd11IfAllowBroadcastSsidAssoc***
  Se o papel assumido pelo equipamento for *roleRoot* ou *roleRepeater* e se o valor deste objecto for verdadeiro, a driver do dispositivo rádio irá responder a pedidos de sonda de SSID difundidos (*Broadcast SSID Probe Requests*) e difundir o seu SSID.

- ***cd11IfPrivacyOptionMaxRate***
  Este objecto indica a taxa máxima de transmissão suportada pelo dispositivo rádio quando é utilizada encriptação WEP. Esta taxa de transmissão é representada em incrementos de 500Kbps.

- ***cd11IfEthernetEncapsulDefault***
  Este objecto especifica o tipo de encapsulação usado no BSS. As encapsulações permitidas são a IEEE 802.1H ou a RFC-1042. O primeiro designa o mecanismo *SubNetwork Access Protocol* (SNAP) como o protocolo de encapsulação. O segundo especifica uma tradução de tramas *Ethernet* de forma a que possam ser trocadas entre estações que não forneçam serviço *Ethernet*.

- ***cd11IfBridgeSpacing***
  Se o equipamento em questão for uma *bridge* sem fios, este valor representa a distância em quilómetros entre esse equipamento e o cliente mais distante.

- ***cd11IfAuxiliarySsidLength***
  Este objecto especifica o número máximo de SSIDs permitidos para uma interface rádio, ou o número de entradas de SSID por interface rádio na tabela *cd11IfAuxSsidTable*

- ***cd11IfVoipExtensionsEnable***
  Este objecto activa os elementos proprietários de respostas de sinalização e sonda (*beacon e probe responses*), de forma a suportar telefones VoIP.

- ***cd11IfDesiredBssAddr***
  Este objecto é utilizado em conjunto com um índice na tabela *cd11IfDesiredBssTable*. Esta tabela é utilizada na situação em que a interface rádio não funciona como ponto de acesso, por exemplo um repetidor ou uma *bridge*. Sendo assim, a tabela contém uma lista dos pontos de acesso preferidos com os quais a interface rádio deverá associar-se. Cada entrada na tabela contém o endereço MAC de um ponto de acesso preferido, existindo um máximo de 4 BSSs configuráveis por interface.

- ***cd11IfAssignedSta***
  Este objecto define o endereço MAC da estação cliente. Quando um cliente associa-se

com esta interface rádio, deverá ser atribuído o valor do objecto *cd11IfAssignedAid* como o seu AID (Association ID).

- ***cd11IfWorldMode***
  Este objecto activa a função *World-Mode* de forma a permitir que o equipamento opere em países diferentes do qual foi fabricado. Existem três modos possíveis, activados através de um inteiro neste objecto. O valor *none*(1) corresponde ao funcionamento normal sem a configuração de *World-Mode*. O valor legacy(2) permite compatibilidade com equipamento de legado. O valor dot11d(3) utiliza o mecanismo IEEE 802.11d para o efeito.

- ***cd11IfWorldModeCountry***
  Com este objecto é possível identificar o país a partir do qual a estação está a operar. Os primeiros dois octetos desta *string* correspondem ao código de dois caracteres descrito no documento ISO/IEC 3166-1. O terceiro octeto poderá tomar um dos seguintes valores. Um espaço ASCII, caso os regulamentos do país onde opera envolvem todos os ambientes, exteriores ou interiores. Um caracter ASCII 'O', caso os regulamentos sejam apenas para ambientes exteriores (*Outdoor*). Um caracter ASCII 'I', caso os regulamentos sejam apenas para ambientes interiores (*Indoor*).

- ***cd11IfMobileStationScanParent***
  Este objecto permite que a interface de rádio procure por uma fonte-origem melhor quando o equipamento é não-root e móvel, por exemplo quando o valor de *cd11IfStation-Role* é *roleWgb(1)*.

O segundo grupo, o *cd11IfAssociationManageGroup* tem o índice 9 nesta MIB. A sua função é fornecer informação de gestão de uma interface IEEE 802.11 e todos os aspectos relacionados com o processo de associação e configurações de encriptação. Aqui, os últimos dois objectos pertencem ao grupo 7 e 9, por isso é feita a inclusão do estudo do grupo 7, o *cd11IfAuthAlgMethodListGroup*. Este grupo tem como função disponibilizar os objectos necessários para configuração e especificação do método de autenticação aplicado a endereços MAC ou em autenticação EAP (*Extensible Authentication Protocol*). Por fim, convém informar que este grupo contém quatro objectos, dois dos quais são analisados em conjunto com o grupo 9, os restantes dois serão analisados conjuntamente com o grupo 10 visto pertencerem simultaneamente a esses dois grupos. Prossegue-se então à análise destes objectos.

```
cd11IfDesiredSsidMaxAssocSta
cd11IfDesiredSsidMicAlgorithm
cd11IfDesiredSsidWepPermuteAlg
cd11IfAuthAlgRequireEap
cd11IfAuthAlgRequireMacAddr
cd11IfAuthAlgDefaultVlan
```

```
cd11IfWepDefaultKeyLen
cd11IfWepDefaultKeyValue

cd11IfAuthAlgEapMethod
cd11IfAuthAlgMacAddrMethod
```

- ***cd11IfDesiredSsidMaxAssocSta***
  Este objecto define o número máximo de estações que se podem associar a esta interface rádio através do objecto *dot11DesiredSSID* presente em *IEEE802dot11-MIB*, a MIB estudada anteriormente. Se o seu valor for zero, o número máximo é limitado apenas pela norma IEEE 802.11 ou por limitações de *hardware* e/ou *firmware* do ponto de acesso.

- ***cd11IfDesiredSsidMicAlgorithm***
  Este objecto define o MIC (*Message Integrity Check*) calculado em pacotes codificados com WEP, de estações associadas com esta interface rádio através do objecto *dot11DesiredSSID* presente em *IEEE802dot11-MIB*.

- ***cd11IfDesiredSsidWepPermuteAlg***
  Com este objecto é possível especificar a função através da qual a chave de encriptação WEP é permutada entre períodos de renovação de chaves para estações associadas com esta interface de rádio.

- ***cd11IfAuthAlgRequireEap e cd11IfAuthAlgRequireMacAddr***
  O primeiro objecto, do tipo booleano, quando assume o valor de verdadeiro indica que as estações que se autentiquem com o algoritmo presente em *dot11AuthenticationAlgorithm* da MIB *IEEE802dot11-MIB* devem completar a autenticação EAP a nível de rede antes das suas tentativas de associação serem desbloqueadas. Se o valor for falso, o processo é o mesmo mas sem ser necessária a autenticação EAP, sendo as estações desbloqueadas logo que a autenticação 802.11 esteja completa. Com o segundo objecto, o procedimento é exactamente o mesmo mas em vez de autenticação EAP, é requerida autenticação por endereço MAC.

- ***cd11IfAuthAlgDefaultVlan***
  Este objecto define o valor por omissão do identificador de VLAN para estações associadas com esta interface rádio. Se o valor for zero, pode indicar duas situações, que a VLAN por omissão não está definida para uma dada autenticação nesta interface, ou que a VLAN por omissão é o VLAN ID nativo.

- ***cd11IfWepDefaultKeyLen e cd11IfWepDefaultKeyValue***
  Estes dois objectos trabalham em conjunto com um índice na tabela *cd11IfWepDefaultKeysTable* que é usada para guardar chaves WEP de tamanho superior a 40 bits,

tamanho que não é permitido em *IEEE802dot11-MIB*. Assim são permitidas quatro chaves por interface, com tamanhos de 40 até 128 bits. Caso o equipamento implemente esta tabela, não deverá implementar a tabela *dot11WEPDefaultKeysTable* da MIB IEEE. O primeiro objecto define simplesmente o tamanho em octetos da chave WEP associada a uma entrada na tabela. O segundo objecto contém o valor da chave WEP. Qualquer tentativa de ler este objecto por parte do NMS (*Network Management System*) resultará no envio de uma *string* nula ou vazia.

- ***cd11IfAuthAlgEapMethod e cd11IfAuthAlgMacAddrMethod***
  Ambos os objectos, do tipo booleano, quando assumem o valor de verdadeiro, indicam o método de autenticação em uso. No caso do primeiro é o método de autenticação EAP, e no segundo o método de autenticação por endereço MAC.

Por fim, e para terminar este agrupamento de 3 "sub-grupos", vem o grupo 10, o cd11IfSsidAssociationGroup. O objectivo deste grupo é fornecer informação de gestão de interfaces ao nível da autenticação e associação de SSIDs. Conforme explicado no grupo anterior, este grupo contém dois objectos do grupo 7 (cd11IfAuthAlgMethodListGroup) que serão analisados em conjunto com o grupo 10. De seguida é feita a análise pormenorizada de cada um dos objectos em questão.

```
cd11IfAuxSsid
cd11IfAuxSsidBroadcastSsid
cd11IfAuxSsidMaxAssocSta
cd11IfAuxSsidMicAlgorithm
cd11IfAuxSsidWepPermuteAlg
cd11IfAuxSsidAuthAlgEnable
cd11IfAuxSsidAuthAlgRequireEap
cd11IfAuxSsidAuthAlgRequireMac
cd11IfAuxSsidAuthAlgDefaultVlan

cd11IfAuxSsidAuthAlgEapMethod
cd11IfAuxSsidAuthAlgMacMethod
```

- ***cd11IfAuxSsid***
  Este objecto é utilizado em conjunto com um índice na tabela cd11IfAuxSsidTable. Cada entrada desta tabela contém um conjunto de atributos que definem um conjunto de serviço (*Service Set*) cujas estações-cliente podem associar a uma interface específica. Uma interface pode ter múltiplos BSS auxiliares enquanto na norma IEEE 802.11 é definido que apenas um BSS preferido pode ser utilizado por cada interface. Cada interface rádio suporta até 25 SSIDs e o objecto *cd11IfAuxiliarySsidLength* especifica o máximo configurado. O objecto em si, especifica um SSID reconhecido pela interface.

A interface deverá responder a pedidos de sonda com este SSID mas não o deve utilizar na transmissão de tramas de sinalização.

- ***cd11IfAuxSsidBroadcastSsid***
  Este objecto indica se um SSID auxiliar é o SSID de difusão, logicamente existindo apenas um SSID de difusão por interface.

- ***cd11IfAuxSsidMaxAssocSta***
  Este objecto define o número máximo de estações que se poderão associar com esta interface rádio, através do objecto *cd11IfAuxSsid*. Se o seu valor for zero, o número máximo é limitado apenas pela norma IEEE 802.11 ou por quaisquer limitações a nível de *hardware* ou *firmware* do equipamento.

- ***cd11IfAuxSsidMicAlgorithm***
  Este objecto define o algoritmo MIC auxiliar aplicado aos pacotes encriptados por WEP das estações associadas a esta interface através de *cd11IfAuxSsid*.

- ***cd11IfAuxSsidWepPermuteAlg***
  Com este objecto é possível definir a função através da qual a chave WEP é permutada entre intervalos de renovação de chave para estações associadas com esta interface.

- ***cd11IfAuxSsidAuthAlgEnable***
  Este objecto, do tipo booleano, quando assume o valor de verdadeiro, indica que é possível autenticar uma associação usando o SSID através do algoritmo descrito no objecto correspondente de *dot11AuthenticationAlgorithmsIndex* na MIB *IEEE802dot11-MIB*.

- ***cd11IfAuxSsidAuthAlgRequireEap e cd11IfAuxSsidAuthAlgRequireMac***
  Caso os valores tanto do primeiro objecto como de *cd11IfAuxSsidAuthAlgEnable*, ambos do tipo booleano, assumirem o valor de verdadeiro, a autenticação de uma associação deverá passar por uma autenticação EAP a nível de rede antes de as tentativas de desbloqueio do cliente prosseguirem. Se o valor deste objecto for falso mas o valor do objecto *cd11IfAuxSsidAuthAlgEnable* for verdadeiro, as estações cliente não necessitam de autenticação adicional após o processo normal segundo a norma IEEE 802.11. De forma idêntica, o segundo objecto segue a mesma lógica do primeiro, sendo a única diferença a utilização de autenticação adicional por endereço MAC em vez da autenticação EAP.

- ***cd11IfAuxSsidAuthAlgDefaultVlan***
  Tal como o objecto *cd11IfAuthAlgDefaultVlan*, este objecto define o valor por omissão do identificador de VLAN para estações associadas com esta interface rádio. Se o valor for zero, pode indicar duas situações, que a VLAN por omissão não está definida para

uma dada autenticação nesta interface, ou que a VLAN por omissão é o VLAN ID nativo

- ***cd11IfAuxSsidAuthAlgEapMethod e cd11IfAuxSsidAuthAlgMacMethod***
  Ambos os objectos, do tipo booleano, quando assumem o valor de verdadeiro, indicam o método de autenticação em uso. No caso do primeiro é o método de autenticação EAP, e no segundo o método de autenticação por endereço MAC.

Na continuação da análise dos grupos desta MIB surge o grupo *cd11IfPhyConfigGroup* tendo o índice 2. Tal como os grupos anteriores, todos os objectos presentes neste grupo pertencem também ao grupo 13 intitulado *cd11IfPhyConfigGroupRev1*. Os objectos dos grupos 2 e 13 tem como funcionalidade fornecer informação para suporte da configuração da camada física da norma IEEE 802.11. Por sua vez, o grupo 13 contém objectos dos grupos 5 e 6, respectivamente *cd11IfDomainCapabilityGroup* e *cd11IfPhyMacCapabilityGroup*. Será então feita a análise dos objectos do grupo 2 que também pertencem ao grupo 13, seguidos dos restantes objectos do grupo 13 que por sua vez, englobam a totalidade dos objectos dos grupos 5 e 6.

```
cd11IfCurrentCarrierSet
cd11IfModulationType
cd11IfPreambleType
cd11IfPhyFhssMaxCompatibleRate
cd11IfPhyDsssMaxCompatibleRate
cd11IfPhyDsssChannelAutoEnable
cd11IfPhyDsssCurrentChannel
cd11IfSuppDataRatesPrivacyValue
cd11IfSuppDataRatesPrivacyEnabl
cd11IfChanSelectEnable
```

- ***cd11IfCurrentCarrierSet***
  Este objecto define o conjunto de portadores (*carrier set*) da interface rádio. De seguida são apresentados os valores possíveis para este objecto.
  usa(0), europe(1), japan(2), spain(3), france(4), belgium(5), israel(6), canada(7), australia(8), japanWide(9), usa5GHz(11), europe5GHz(12), japan5GHz(13), singapore5-GHz(14), taiwan5GHz(15), china(16)

- ***cd11IfModulationType***
  Este objecto especifica o tipo de modulação de rádio-frequência (RF) da interface, com dois valores possíveis. O primeiro, *standard(1)*, é a modulação por omissão definida na norma IEEE 802.11. O segundo, *mok(2)* utiliza a técnica MOK (*M-ary Orthogonal Keying*) da qual a técnica CCK (*Complementary Code Keying*) é uma variação. Esta técnica foi usada antes do IEEE ter terminado a norma 802.11 de alta velocidade.

- ***cd11IfPreambleType***
  Este objecto especifica o tipo de preâmbulo actualmente em uso pela estação, com os valores possíveis *long(1)* e *short(2)*. Com um preâmbulo longo é assegurada compatibilidade entre pontos de acesso e modelos mais antigos de adaptadores de rede (dispositivos cliente) *Cisco Aironet Wireless*. Com um preâmbulo curto, é melhorado o desempenho do rendimento.

- ***cd11IfPhyFhssMaxCompatibleRate e cd11IfPhyDsssMaxCompatibleRate***
  Estes objectos indicam a taxa de dados máxima a que uma estação pode transmitir dados, contendo um valor representante dessa mesma taxa num octeto. A taxa deverá encontrar-se num intervalo entre 2 e 127, correspondendo a incrementos na taxa de dados de 500 kbps, desde 1 Mbps até 63.5 Mbps. Obviamente, o primeiro objecto aplica-se no caso de utlização da técnica FHSS e o segundo objecto utilização de DSSS.

- ***cd11IfPhyDsssChannelAutoEnable***
  Este objecto, do tipo booleano, quando assume o valor de verdadeiro e o papel da interface rádio é ponto de acesso com o valor de *cd11IfStationRole* sendo *roleRoot* a interface irá procurar por actividade de outros BSS em todos os canais disponíveis antes de estabalecer o seu próprio BSS. Depois de fazer esta procura, a interface irá estabalecer o seu próprio BSS no canal com menos probabilidade de congestão de sinal de rádio, evitando sobreposição de canais.

- ***cd11IfPhyDsssCurrentChannel***
  Este objecto indica o canal de frequência operacional, tal como escolhido através de procura selectiva ou através do valor presente em *dot11CurrentChannel* na MIB *IEEE802dot11-MIB*. Os números válidos de canais estão definidos na norma IEEE 802.11. Na América do Norte, os canais 802.11b permitidos são do 1 ao 11 e os canais 802.11a permitidos são 36, 40, 44, 48, 52, 56, 60 e 64.

- ***cd11IfSuppDataRatesPrivacyValue e cd11IfSuppDataRatesPrivacyEnabl***
  Estes dois objectos relacionam-se entre si na tabela *cd11IfSuppDataRatesPrivacyTable*. A funcionalidade desta tabela é definir a taxa de transmissão e recepção em cada interface, bem como o suporte de encriptação WEP destas taxas. Então, o primeiro objecto define as taxas de transmissão e recepção suportadas pelo PLCP e PMD representadas em incrementos de de 500 kbps, desde 1 Mbps até 63.5 Mbps. O segundo objecto indica se com a taxa de dados correspondente na tabela, é suportada encriptação WEP tanto para transmissão como recepção.

- ***cd11IfChanSelectEnable***

Este objecto pertence à tabela *cd11IfChanSelectTable*, que especifica para cada canal se o processo de procura controlado por *cd11IfPhyDsssChannelAutoEnable* pode seleccionar um canal para utilização. Se o valor deste objecto, do tipo booleano, for verdadeiro indica que o objecto *cd11IfChanSelectChannel* está disponível para ser utilizado pelo sistema depois de procurar por ocupação do canal.

Conforme explicado anteriormente, continua-se agora com os restantes objectos do grupo 13 (*cd11IfPhyConfigGroupRev1*) que por sua vez englobam os grupos 5 e 6, respectivamente *cd11IfDomainCapabilityGroup* e *cd11IfPhyMacCapabilityGroup*. O grupo 5 tem como funcionalidade disponiblizar objectos de forma a gerir os canais e potência de transmissão de uma interface rádio no seu domínio regulamentar. O grupo 6 disponibiliza objectos essenciais à gestão da taxa de transmissão de dados.

```
cd11IfDomainCapabilitySet
cd11IfPhyBasicRateSet
cd11IfPhyMacSpecification
cd11IfPhyConcatenation
cd11IfClientNumberTxPowerLevels
cd11IfClientTxPowerLevel1 -- cd11IfClientTxPowerLevel8
cd11IfClientCurrentTxPowerLevel
```

- ***cd11IfDomainCapabilitySet***
  Este objecto indica a configuração de multi-domínio escolhida no momento. A configuração controla o número do primeiro canal operacional, o número de canais operacionais e o nível de potência máxima de transmissão da interface.

- ***cd11IfPhyBasicRateSet***
  Com este objecto é possível especificar se uma taxa de dados presente em *dot11OperationalRateSet* da MIB IEEE802dot11 é uma taxa de dados básica (*Basic Rate*) para esta interface. Caso seja esse o caso, o octeto correspondente deste objecto irá conter o valor 128. Caso contrário, o octeto correspondente deste objecto será zero.

- ***cd11IfPhyMacSpecification***
  Este objecto simplesmente indica qual a norma IEEE 802.11 em uso nesta interface, com as seguintes opções:
  ieee802dot11a(1) - Norma IEEE 802.11a
  ieee802dot11b(2) - Norma IEEE 802.11b
  ieee802dot11g(3) - Norma IEEE 802.11g

- ***cd11IfPhyConcatenation***
  Caso o valor de *cd11IfStationRole* seja *roleBridge*, ou *roleNrBridge*, este objecto define o tamanho máximo de concatenação de pacotes para todos os pacotes de saída. Para os produtos *Cisco* de 5GHz, o intervalo de valores situa-se entre 1600 e 4000.

- ***cd11IfClientNumberTxPowerLevels***
  Este objecto representa o número de níveis de potência disponíveis para os clientes, variando entre 1 e 8.

- ***cd11IfClientTxPowerLevel1 – cd11IfClientTxPowerLevel8***
  Estes objectos permitem a definição de níveis de potência pré-determinados, para alteração quando necessário, estando distribuídos do nível 1 ao nível 8. Cada nível representa a potência em mW ou dBm.

- ***cd11IfClientCurrentTxPowerLevel***
  Este objecto representa o nível de potência escolhido para operação.

Continuando com a análise desta MIB, surgem agora os objectos dedicados a estatísticas. Tais objectos estão presentes nos grupos *cd11IfMacStatisticsGroup* e *cd11IfMgmtStatisticsGroup*. O primeiro fornece informação estatística sobre a camada MAC da interface, o segundo fornece informação estatística relacionada com pontos de acesso ilegais (*rogue AP*).

```
cd11IfTransDeferEnerDetects
cd11IfRecFrameMacCrcErrors
cd11IfSsidMismatches

cd11IfLastRogueApMacAddr
cd11IfRogueApListSize
```

- ***cd11IfTransDeferEnerDetects***
  Este contador incrementa o seu valor quando a transmissão de uma trama é deferida devido a detecção de energia.

- ***cd11IfRecFrameMacCrcErrors***
  O valor deste contador é incrementado sempre que é recebida uma trama com um erro CRC a nível MAC.

- ***cd11IfSsidMismatches***
  Este contador será incrementado quando uma trama de resposta de sinalização ou sonda (*beacon e probe response frame*) recebida mas o seu SSID não corresponde ao valor contido em *dot11DesiredSSID* da MIB *IEEE802dot11*.

- ***cd11IfLastRogueApMacAddr***
  Este objecto contém o endereço MAC do ponto de acesso ilegal mais recentemente detectado. O seu valor será zero se não foi detectado nenhum ponto de acesso ilegal desde o reinício do sistema.

- ***cd11IfRogueApListSize***
  Este objecto contém o comprimento da lista onde são guardados os pontos de acesso

ilegais detectados, em conjunto com o seu endereço MAC desde o momento de reinicia-ção do sistema. O valor zero neste objecto indica que não está implementada tal lista ou que desde o momento de reiniciação do sistema não foi detectado nenhum ponto de acesso ilegal.

No seguimento deste estudo vêm mais dois grupos, desta vez relacionados com a gestão de VLANs. O grupo *cd11IfVlanEncryptKeyConfigGroup* com índice 4, é responsável pela configuração de chaves WEP para VLANs específicas em operação na interface rádio. O grupo *cd11IfVlanManageGroup* com índice 11 é dedicado ao fornecimento de informação para gestão de configurações de VLANs e respectiva encriptação. Como todos os objectos pertencentes a *cd11IfVlanEncryptKeyConfigGroup* estão presentes também no grupo *cd11IfVlanManageGroup*, serão logicamente analisados em conjunto. Os primeiros três objectos são os que ambos os grupos têm em comum.

```
cd11IfVlanEncryptKeyLen
cd11IfVlanEncryptKeyValue
cd11IfVlanEncryptKeyStatus

cd11IfVlanEncryptKeyTransmit
cd11IfVlanSecurityVlanEnabled
cd11IfVlanBcastKeyChangeInterval
cd11IfVlanBcastKeyCapabilChange
cd11IfVlanBcastKeyClientLeave
cd11IfVlanSecurityCiphers
cd11IfVlanSecurityRowStatus
```

- ***cd11IfVlanEncryptKeyLen, cd11IfVlanEncryptKeyValue e cd11IfVlanEncryptKeyStatus***
  Estes três objectos em conjunto com o respectivo identificador e índice, formam uma entrada na tabela *cd11IfVlanEncryptKeyTable*, que contém chaves WEP partilhadas para todos os pacotes transmitidos e recebidos numa VLAN, caso as opções de VLAN e WEP estejam estejam activadas na interface. Se a encriptação WEP está activa, a chave por omissão será a chave usada para encriptar tramas de difusão e *multicast* associados com o identificador dessa VLAN. Assim, o primeiro objecto especifica o comprimento em octetos da chave correspondente na tabela, com valores normais de 5 octetos para chaves WEP de 40 bits e 13 octetos para chaves WEP de 128 bits. Se o valor deste objecto for zero, indica que uma chave não foi configurada, mas a VLAN está activa. O segundo objecto contém simplesmente o valor secreto da chave WEP, sendo sempre devolvido uma *string* de comprimento zero quando é feita uma tentativa de leitura do objecto, por razões de segurança. O terceiro objecto controla e reflecte o estado das entradas da tabela. Quando a entrada está no estado activo, a aplicação

de gestão pode alterar o comprimento e o valor da chave. Para fins de eliminação de uma entrada na tabela, pode-se alterar o valor deste objecto para *'destroy'*.

- ***cd11IfVlanEncryptKeyTransmit***
  Este objecto indica qual a chave de transmissão da VLAN em uso. Apenas uma das quatro chaves de uma VLAN podem ser a chave de transmissão, pelo que quando é activada outra chave diferente da corrente, o agente automaticamente irá alterar o valor das outras três chaves para inactivo, caso elas existam.

- ***cd11IfVlanSecurityVlanEnabled***
  Este objecto, do tipo booleano, quando assume o valor de verdadeiro indica que a sub-interface desta VLAN está activa em todas as portas *trunk* e híbridas. Em plataformas de gestão que suportem a criação de sub-interfaces VLAN, a alteração deste objecto para verdadeiro irá criar as sub-interfaces VLAN em todas as portas. Caso o valor seja alterado para falso, a VLAN será removida.

- ***cd11IfVlanBcastKeyChangeInterval***
  e cd11IfVlanBcastKeyCapabilChange
  Ambos os objectos configuram aspectos relacionados com a chave de difusão. O primeiro objecto indica o período de rotação da chave de difusão, e se o seu valor for zero, indica que não há rotação da chave. O segundo objecto, do tipo booleano, quando verdadeiro indica que uma nova chave será utilizada todas as vezes que a encriptação dos clientes for alterada na VLAN.

- ***cd11IfVlanBcastKeyClientLeave***
  Caso o valor deste objecto seja verdadeiro, uma nova chave será utilizada quando um cliente da VLAN for desassociado.

- ***cd11IfVlanSecurityCiphers***
  Este objecto indica qual o tipo de combinações de encriptação possíveis, que não sejam as implementadas na norma IEEE 802.11. Algumas plataformas poderão suportar apenas um sub-conjunto destas combinações. Assim, as combinações possíveis são:
  none: Sem encriptação
  aesccm: WPA (*WiFi Protected Access*) AES (*Advanced Encryption Standard*) CCMP (*CCM Mode Protocol*)
  ckip: *Cisco Per Packet Key*
  cmic: *Cisco MMH (Multi-Modal Hashing) MIC (Message Integrity Check)*
  ckip|cmic: *Cisco Per Packet Key* e MIC
  mmic: *Michael* MIC
  tkip: *WPA Temporal Key*
  wep128: 128-bit WEP
  wep40: 40-bit WEP

tkip|wep128: *WPA Temporal Key* e 128-bit WEP
tkip|wep40: *WPA Temporal Key* e 40-bit WEP

- ***cd11IfVlanSecurityRowStatus***
  Este objecto é utilizado para criar uma nova entrada, modificar ou eliminar uma configuração de encriptação de uma VLAN existente na tabela *cd11IfVlanSecurityTable*. A criação de entradas pode ser feita através do parâmetro *createAndGo* com o preenchimento obrigatório das colunas *cd11IfVlanSecurityVlanEnabled* e *cd11IfVlanSecurityCiphers*. A modificação e eliminação poderão ser efectuadas através dos parâmetros *createAndGo* e *delete*, respectivamente.

Para finalizar o estudo desta MIB, serão analisados os dois restantes grupos, o *cd11IfRemoteMonitoringGroup* e o cd11IfPhyErpConfigGroup. O primeiro grupo tem como objectivo fornecer informação de gestão para configuração de monitorização remota. O segundo grupo contém informação para configurar os níveis de potência para OFDM de taxa estendida ERP (*Extended Rate PHY*). Apesar de não pertencer ao grupo de monitorização, o objecto cd11IfRemoteRadioMacAddr será mencionado, pelo que convém esclarecer a sua função que é simplesmente conter o endereço MAC da interface a ser monitorizada. São apresentados de seguida os objectos que fazem parte destes grupos e é feita a análise detalhada de cada um dos objectos.

```
cd11IfRadioMonitorPollingFreq
cd11IfRadioMonitorPollingTimeOut
cd11IfLocalRadioMonitorStatus
cd11IfRadioMonitorRowStatus

cd11IfErpOfdmNumberTxPowerLevels
cd11IfErpOfdmTxPowerLevel1 -- cd11IfErpOfdmTxPowerLevel8
cd11IfErpOfdmCurrentTxPowerLevel
```

- ***cd11IfRadioMonitorPollingFreq***
  Este objecto indica a frequência em segundos com que a interface especificada em *cd11IfRemoteRadioMacAddr* é consultada com a finalidade de saber a condição do seu estado.

- ***cd11IfRadioMonitorPollingTimeOut***
  Este objecto representa o número total em segundos que a estação de monitorização pode tolerar a falha de consultas a interface alvo de monitorização. Depois deste tempo ter passado, a estação de monitorização assume o papel de interface activa. Então irá interromper toda a actividade de monitorização e colocar a instância de *cd11IfLocalRadioMonitorStatus* da estação que entrou em falha para o estado activo.

- ***cd11IfLocalRadioMonitorStatus***
  Este objecto pode ser alterado pela aplicação de gestão para o valor monitor(2). Neste estado a interface escolhida irá monitorizar a interface especificada em *cd11IfRemoteRadioMacAddr*. Quando a interface alvo de monitorização falhar esta interface irá assumir as suas funções e tornar-se activa. Obviamente quando esta interface está activa, não consegue monitorizar outras interfaces, por isso, todas as outras instâncias de *cd11IfLocalRadioMonitorStatus* serão colocadas com inactivas.

- ***cd11IfRadioMonitorRowStatus***
  Este objcto é utilizado para criar uma nova entrada, modificar ou eliminar entradas na tabela *cd11IfRadioMonitoringTable*. A criação de entradas pode ser feita através do parâmetro *createAndGo*. A modificação e eliminação poderá ser efectuada através dos parâmetros *createAndGo* e *destroy*, respectivamente.

- ***cd11IfErpOfdmNumberTxPowerLevels***
  Este objecto representa o número de níveis de potência disponíveis para os clientes, variando entre 1 e 8.

- ***cd11IfErpOfdmTxPowerLevel1 – cd11IfErpOfdmTxPowerLevel8***
  Estes objectos permitem a definição de níveis de potência pré-determinados, para alteração quando necessário, estando distribuídos do nível 1 ao nível 8. Cada nível representa a potência em mW ou dBm.

- ***cd11IfErpOfdmCurrentTxPowerLevel***
  Este objecto representa o nível de potência operacional.

B.3 CISCO-DOT11-ASSOCIATION-MIB

Para finalizar o estudo de MIBs proposto é analisada a *CISCO-DOT11-ASSOCIATION-MIB*, que tal como a MIB anterior, é uma extensão da *IEEE802dot11-MIB* é obviamente também foi criada pela *Cisco* pelo que os seus objectos são proprietários. Assim, existe um melhor suporte as tecnologias desenvolvidas e criadas pela empresa, para utilização nos seus produtos. Esta MIB não é tão extensa como as anteriores, pois ocupa-se de uma área muito especifica da operação de pontos de acesso sem fios, a gestão da associação de clientes. Contém ainda alguns objectos para suporte a encaminhamento de pacotes de dados. De seguida é iniciado o estudo desta MIB, na mesma estrutura apresentada anteriormente.
Para começar, é analisado o grupo *ciscoDot11AssocGlobalGroup* que tem como objectivo gerir o número de dispositivos associados, bem como alguns objectos estatísticos, muito úteis ao administrador de rede. De seguida são listados os objectos pertencentes a este grupo, seguido de uma análise mais detalhada de cada um dos seus constituintes.

`cDot11ParentAddress`

```
cDot11ActiveWirelessClients
cDot11ActiveBridges
cDot11ActiveRepeaters
cDot11AssStatsAssociated
cDot11AssStatsAuthenticated
cDot11AssStatsRoamedIn
cDot11AssStatsRoamedAway
cDot11AssStatsDeauthenticated
cDot11AssStatsDisassociated
```

- ***cDot11ParentAddress***
  Este objecto contém o endereço MAC do ponto de acesso pai para este dispositivo. Caso o dispositivo esteja a operar como um ponto de acesso raiz, o valor deste objecto será zero, indicando logicamente que o dispositivo não tem um nodo pai.

- ***cDot11ActiveWirelessClients, cDot11ActiveBridges e cDot11ActiveRepeaters***
  O primeiro objecto representa simplesmente o número de clientes associados com este dispositivo na interface em questão. Os outros dois objectos têm exactamente a mesma função, mas para número de *bridges* associadas e número de repetidores associados, respectivamente,

- ***cDot11AssStatsAssociated e cDot11AssStatsAuthenticated***
  Estes dois contadores estatísticos representam a contagem de estações, respectivamente, associadas e autenticadas com este dispositivo na interface em questão desde a sua reiniciação.

- ***cDot11AssStatsRoamedIn e cDot11AssStatsRoamedAway***
  O primeiro objecto é um contador que indica o número de estações que mudaram de outro dispositivo para este desde a sua reiniciação. O segundo objecto tem a mesma função mas no sentido inverso, ou seja, conta o número de dispositivos que mudaram deste dispositivo para outro desde o momento da sua reiniciação.

- ***cDot11AssStatsDeauthenticated e cDot11AssStatsDisassociated***
  Estes dois objectos representam a contagem de estações, respectivamente, desautenticadas e desassociadas com este dispositivo na interface em questão desde a sua reiniciação.

O grupo *ciscoDot11AssocGlobalGroup* é o próximo grupo a analisar nesta MIB. Os seus objectos são dedicados sobretudo à configuração de clientes associados bem como alguma informação relevante para o sistema de gestão. De seguida são apresentados os objectos pertencentes ao grupo, e posteriormente é efectuada a análise individual de cada um deles.

```
cDot11ClientParentAddress
cDot11ClientRoleClassType
cDot11ClientDevType
cDot11ClientRadioType
cDot11ClientWepEnabled
cDot11ClientWepKeyMixEnabled
cDot11ClientMicEnabled
cDot11ClientPowerSaveMode
cDot11ClientAid
cDot11ClientDataRateSet
```

- ***cDot11ClientParentAddress***
  Este objecto contém o endereço MAC do ponto de acesso pai para este dispositivo, caso o dispositivo esteja a operar como um repetidor. Se este dispositivo não tiver um nodo pai, o valor deste objecto será zero.

- ***cDot11ClientRoleClassType, cDot11ClientDevType e cDot11ClientRadioType***
  Estes três objectos categorizam o dispositivo cliente em três aspectos respectivamente, a classificação do papel ou função do cliente, o tipo de dispositivo do cliente e a classificação da interface rádio do cliente.

- ***cDot11ClientWepEnabled***
  Este objecto, do tipo booleano, quando verdadeiro indica que o mecanismo de chaves WEP é utilizado para transmitir tramas para este cliente. Obviamente, quando o objecto assume o valor de falso, indica que o mecanismo não é utilizado.

- ***cDot11ClientWepKeyMixEnabled***
  Quando este objecto, do tipo booleano, assume o valor de verdadeiro, indica que este cliente está a utilizar mistura de chaves WEP.

- ***cDot11ClientMicEnabled***
  Este objecto, quando assume o valor de verdadeiro, indica que MIC está activado para este cliente.

- ***cDot11ClientPowerSaveMode***
  Este objecto representa o modo de gestão de energia do cliente, existindo dois modos possíveis. O modo *active(1)* indica que o cliente não está em modo de poupança de energia e está a enviar ou receber dados. O modo *powersave(2)* indica que o cliente está em modo de poupança de energia e acorda de vez em quando para verificar se existem dados para receber ou enviar.

- ***cDot11ClientAid***
  Este objecto contém o número do identificador de associação de clientes ou de endereços *multicast* do cliente. Para um endereço *multicast* com clientes, o seu valor será zero. Para a associação deste dispositivo com o dispositivo pai, o seu valor será 1.

- ***cDot11ClientDataRateSet***
  Este objecto representa o conjunto de taxas de dados com que o cliente pode enviar ou receber dados, podendo cada cliente suportar até 126 taxas de dados distintas. Cada octeto contém um valor inteiro que representa cada uma destas 126 taxas de dados e cada uma delas será de um intervalo de 2 a 127, correspondendo a taxas de dados em incrementos de 500kbps desde 1Mbps até 63.5 Mbps. Este valor é descrito nas tramas de sinalização, sonda, associação e reassociação, e é utilizado para determinar se o cliente que se deseja associar tem uma taxa de dados compatível.

O grupo seguinte é o *ciscoDot11ClientStatGroup*. Este grupo fornece informação estatística sobre os clientes associados a um ponto de acesso sem fios. De notar que os objectos aqui analisados são relativos a dispositivos cliente presentes na tabela *cDot11ClientConfigInfoTable*, ou seja, cada um dos valores destes objectos será correspondente a apenas um cliente associado.

```
cDot11ClientCurrentTxRateSet
cDot11ClientUpTime
cDot11ClientSignalStrength
cDot11ClientSigQuality
cDot11ClientPacketsReceived
cDot11ClientBytesReceived
cDot11ClientPacketsSent
cDot11ClientBytesSent
cDot11ClientAgingLeft
cDot11ClientDuplicates
cDot11ClientMsduRetries
cDot11ClientMsduFails
cDot11ClientWepErrors
cDot11ClientMicErrors
cDot11ClientMicMissingFrames
```

- ***cDot11ClientCurrentTxRateSet***
  Este objecto representa o conjunto de taxas de dados com que o cliente pode enviar ou receber dados, podendo cada cliente suportar até 126 taxas de dados distintas. Cada octeto contém um valor inteiro que representa cada uma destas 126 taxas de dados e cada uma delas será de um intervalo de 2 a 127, correspondendo a taxas de dados em

incrementos de 500kbps desde 1Mbps até 63.5 Mbps. Este valor é descrito nas tramas de sinalização, sonda, associação e reassociação,

- ***cDot11ClientUpTime***
  Este objecto indica o tempo em segundos que o dispositivo cliente está associado.

- ***cDot11ClientSignalStrength e cDot11ClientSigQuality***
  Ambos os objectos são uma medida dependente do dispositivo, sendo o primeiro a medida da força do sinal do pacote mais recente, e o segundo uma medida da qualidade do sinal do pacote mais recente,

- ***cDot11ClientPacketsReceived e cDot11ClientBytesReceived***
  Estes dois objectos são contadores de tráfego de entrada do cliente. O primeiro indica o número de pacotes recebidos e o segundo indica o número de bytes recebidos.

- ***cDot11ClientPacketsSent e cDot11ClientBytesSent***
  Estes dois objectos são contadores de tráfego de saída do cliente. O primeiro indica o número de pacotes enviados e o segundo indica o número de bytes enviados.

- ***cDot11ClientAgingLeft***
  Este objecto representa o número de segundos de tempo de envelhecimento restante para este cliente. Isto significa que passado este tempo, se não foi recebido nenhum pacote no cliente, o ponto de acesso a funcionar como *bridge* irá remover a entrada do cliente na sua tabela.

- ***cDot11ClientDuplicates***
  Este objecto é um contador que incrementa o seu valor quando um pacote é recebido pelo cliente e o campo de controlo de sequência do seu cabeçalho indica que o pacote é um duplicado.

- ***cDot11ClientMsduRetries e cDot11ClientMsduFails***
  O primeiro objecto é um contador que incrementa o seu valor quando um MSDU é transmitido com sucesso depois de uma ou mais retransmissões para este cliente. O segundo objecto, também um contador, é incrementado quando um MSDU não é transmitido com sucesso devido ao número de tentativas de transmissão ter excedido o seu limite.

- ***cDot11ClientWepErrors***
  Este objecto contém o número de pacotes recebidos pelo cliente que falharam a desencriptação através do mecanismo de segurança WEP.

- ***cDot11ClientMicErrors e cDot11ClientMicMissingFrames***
  O primeiro objecto indica o número de erros de MIC para o cliente, enquanto que o segundo indica o número de pacotes MIC em falta para o cliente.

Para terminar o estudo desta MIB falta analisar o último grupo desta MIB, o *ciscoDot11ClientInfoGroup* que tem como função fornecer informação básica sobre o equipamento do cliente.

```
cDot11ClientSoftwareVersion
cDot11ClientName
cDot11ClientAssociationState
cDot11ClientIpAddressType
cDot11ClientIpAddress
```

- ***cDot11ClientSoftwareVersion***
  Este objecto representa a versão de *software Cisco IOS*, caso do outro lado da associação estiver uma *bridge*, ponto de acesso ou repetidor. Caso seja um cliente este objecto representa a versão do *firmware* da placa do cliente.

- ***cDot11ClientName***
  Este objecto representa o nome de *host* do dispositivo, caso do outro lado da associação estiver uma *bridge*, ponto de acesso ou repetidor. Caso seja um cliente este objecto representa o nome de configuração do cliente.

- ***cDot11ClientAssociationState***
  Este objecto indica o estado do processo de autenticação e associação, com os seguintes valores possíveis:
  initial(1) - pedido de associação recebido pelo cliente.
  authenNotAssociated(2) - autenticado mas não associado.
  assocAndAuthenticated(3) - autenticado e associado.
  assocNotAnuthenticated(4) - associado mas não autenticado.

- ***cDot11ClientIpAddressType e cDot11ClientIpAddress***
  O primeiro objecto representa o tipo de endereço IP do cliente. O segundo objecto representa o endereço estático ou o endereço atribuído por DHCP do cliente.

Assim termina-se a análise das MIBs mais relevantes para este trabalho. Com isto foi possível recolher informação com mais facilidade para a tarefa proposta da criação de uma pequena aplicação de gestão. Conforme as decisões tomadas, foram escolhidos objectos SNMP específicos que contribuiram para a resolução do problema. Proporcionou também um melhor conhecimento sobre o que se passa nas camadas física e *MAC* em SNMP e respectivas implementações em produtos comerciais. Deste modo será possível desenvolver ferramentas ou aplicações reutilizando objectos já existentes, mas utilizados com outros fins.