PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE TECNOLOGIAS

ARTURO JOSÉ FENILE PERIS

CONTROLE DE VAZÃO EM REDES IEEE 802.11 COM PRESENÇA DE OFENSORES

CAMPINAS

2012

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS

GRÃO-CHANCELER

Dom Airton José dos Santos

MAGNÍFICA REITORA

Prof. Dra. Angela de Mendonça Engelbrecht

VICE-REITOR

Prof. Dr. Eduard Prancic

PRÓ-REITORA DE PESQUISA E PÓS-GRADUAÇÃO

Prof. Dra. Vera Engler Cury

DIRETOR DO CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE TECNOLOGIAS

Prof. Ricardo Luís de Freitas

COORDENADOR DO PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO SENSU* EM ENGENHARIA ELÉTRICA

CURSO DE MESTRADO PROFISSIONAL EM GESTÃO DE REDES DE TELECOMUNICAÇÕES

ÁREA DE CONCENTRAÇÃO: GESTÃO DE REDES E SERVIÇOS

Prof. Dr. Marcelo Luis Francisco Abbade

ARTURO JOSÉ FENILE PERIS

CONTROLE DE VAZÃO EM REDES IEEE 802.11 COM PRESENÇA DE OFENSORES

Dissertação apresentada como exigência para obtenção do Título de Mestre em Gestão de Redes de Telecomunicações, ao Programa de Pós-Graduação em Engenharia Elétrica, Pontifícia Universidade Católica de Campinas.

Orientador: Prof. Dr. Alexandre de Assis Mota

PUC CAMPINAS

2012

Ficha Catalográfica
Elaborada pelo Sistema de Bibliotecas e
Informação – SBI – PUC-Campinas

t621.3845
P446c

Peris, Arturo José Fenile.
Controle de vazão em redes IEEE 802.11 com presença de ofensores / Arturo José Fenile Peris. - Campinas: PUC-Campinas, 2012.
131p.

Orientador: Alexandre de Assis Mota.
Dissertação (mestrado) – Pontifícia Universidade Católica de Campinas, Centro de Ciências Exatas, Ambientais e de Tecnologias, Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Sistemas de comunicação sem fio. 2. Sistemas de telecomunicações. 3. Redes de sensores sem fio. 4. Inovações tecnológicas. I. Mota, Alexandre de Assis. II. Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologias. Pós-Graduação em Engenharia Elétrica. III. Título.

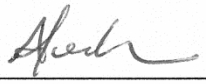
22.ed.CDD – t621.3845

ARTURO JOSÉ FENILE PERIS

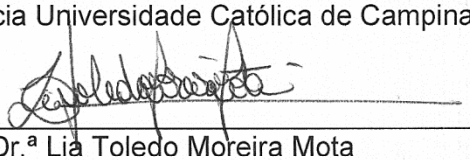
CONTROLE DE VAZÃO EM REDES IEEE 802.11 COM PRESENÇA DE OFENSORES.

Dissertação apresentada ao Curso de Mestrado Profissional em Gestão de Redes de Telecomunicações do Centro de Ciências Exatas, Ambientais e de Tecnologias da Pontifícia Universidade Católica de Campinas como requisito parcial para obtenção do título de Mestre em Gestão de Redes de Telecomunicações.
Área de Concentração: Gestão de Redes e Serviços.
Orientador: Prof. Dr. Alexandre de Assis Mota

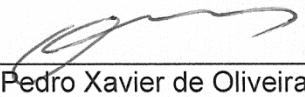
Dissertação defendida e aprovada em 22 de junho de 2012 pela Comissão Examinadora constituída dos seguintes professores:



Prof. Dr. Alexandre de Assis Mota
Orientador da Dissertação e Presidente da Comissão Examinadora
Pontifícia Universidade Católica de Campinas



Prof.ª Dr.ª Lia Toledo Moreira Mota
Pontifícia Universidade Católica de Campinas



Prof. Dr. Pedro Xavier de Oliveira
Universidade Estadual de Campinas

Dedico este trabalho à minha Mãe Ronysa, cujo incentivo, exemplo de vida e ensinamentos foram fundamentais ao sucesso deste projeto.

Dedico também este trabalho à minha esposa Adriana, cuja compreenção e apoio tornaram possíveis este momento.

Finalmente, dedico à minhas filhas Laura e Flávia, que com seu amor incondicional deram a energia necessária para a finalização desta empreitada.

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Alexandre de Assis Mota,
Por ter viabilizado este trabalho e cuja orientação e conselhos foram fundamentais para sua conclusão.

Ao Prof. Dr. Omar Carvalho Branquinho,
Pelas informações, experiências e materiais compartilhados.

Aos Professores Dr. Marcelo Luís Francisco Abbade, Dra. Lia Toledo Moreira Mota, Dr. Eric Alberto de Mello Fagotto e Dr. Davi Bianchini
Pela dedicação e paciência em compartilhar conhecimentos e experiências durante as aulas.

Aos meus companheiros de mestrado,
Pela ajuda e pela amizade construída durante este desafio.

RESUMO

PERIS, Arturo José Fenile. *Controle de vazão em redes IEEE 802.11 com presença de ofensores*. 131f. Dissertação (Mestrado em Gestão de Redes de Telecomunicações) – Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologias, Programa de Pós-Graduação em Engenharia Elétrica, Campinas, 2012.

A concepção do padrão IEEE 802.11 permite que, em determinadas situações, a competição pelo acesso ao meio de transmissão resulte em uma anomalia. Como resultado dessa anomalia, estações móveis que deveriam conseguir altas taxas de transmissão são prejudicadas e acabam por não conseguir essas taxas. A consequência é o baixo aproveitamento da capacidade de transmissão do sistema sem fio e o natural aumento do tempo em que as estações móveis permanecem ligadas e consumindo energia; por conta disso, o aumento no consumo de energia é outro efeito dessa anomalia. A partir desse cenário, uma forma de combater essa anomalia é a melhoria da utilização do meio de transmissão sem fio. Nesse contexto, este trabalho apresenta uma proposta de bancada para estudos de controle de vazão em redes IEEE 802.11; essa bancada permite gerenciar a largura de banda e coletar dados estatísticos relativos ao tráfego das sessões dos usuários conectados à rede sem fio. Como estudo resultante da utilização da bancada, há uma proposta de mitigação da anomalia por meio da gestão do tráfego dos usuários. Os resultados indicam que o controle do tráfego de usuários que estão em más condições de transmissão sem fio pode melhorar o aproveitamento da capacidade de transmissão de um ponto de acesso IEEE 802.11.

Termos de indexação: IEEE 802.11, Wi-Fi, gerenciamento de largura de banda, WISPr

ABSTRACT

PERIS, Arturo José Fenile. *Bandwidth Management on IEEE 802.11 networks in the presence of offending users.* 131f. Dissertation (Master in Telecommunications Networks Management) – Pontifícia Universidade Católica de Campinas. Centro de Ciências Exatas, Ambientais e de Tecnologias, Programa de Pós-Graduação em Engenharia Elétrica, Campinas, 2012.

The design of the IEEE 802.11 standard allows, in certain situations, competition for access to the transmission medium results in an anomaly. As a result of this anomaly, mobile stations that should achieve high rates of transmission are impaired and don't get those rates. The consequence is the transmission capacity low utilization of the wireless system and the natural increasing of the time mobile stations stays on (and consuming energy); because of this, the increase in energy consumption is another effect of this anomaly. From this scenario, one way to deal with this anomaly is to have a better utilization of the wireless transmission medium. In this context, this work proposes a bench for studies of flow control in IEEE 802.11; this bench allows the management of bandwidth and the collection of traffic statistics from users sessions connected to the wireless network. As a resulting study from the bench using, there is a proposal to mitigate the anomaly through the user traffic management. The results indicate that the controlling the traffic from users who are in poor wireless transmission may produce a better utilization of transmission capacity of an IEEE 802.11 access point.

Indexing terms: *IEEE 802.11, Wi-Fi, Bandwidth Management, WISPr*

LISTA DE FIGURAS

Figura 1 - Exemplo de rede sem fio 21
Figura 2 - Localização da MAC, DCF e PCF no modelo de referência OSI 22
Figura 3 – Situação com possibilidade de ocorrer a anomalia da MAC 25
Figura 4 – Constatação da anomalia da MAC. Fonte: (GUIRARDELLO, 2008) 27
Figura 5 - Arquitetura utilizada nos experimentos 29
Figura 6 - NAS na arquitetura RADIUS 31
Figura 7 - AAA na arquitetura RADIUS 33
Figura 8 – Gerencia de Rede na arquitetura utilizada nos experimentos 35
Figura 9 – BWM na arquitetura utilizada nos experimentos 36
Figura 10 - Autenticação/Autorização com RADIUS 38
Figura 11 - Exemplo de mensagem Access-Request 39
Figura 12 - Exemplo de mensagem Access-Accept 39
Figura 13 - "accounting" RADIUS 40
Figura 14 – Exemplo de mensagem de "accounting" 41
Figura 15 – Protocolo DHCP 42
Figura 16 – CoA – "Change of Authorization" 43
Figura 17 - Controle da vazão dos usuários 44
Figura 18 – Estabelecimento da sessão 46
Figura 19 – Registro de Dados CSV e Monitor RSSI 46
Figura 20 – Variação da Vazão Permitida 47
Figura 21 - Componentes da Bancada de Testes 49
Figura 22 - Conexões físicas entre os componentes 52
Figura 23 - Tela de login do Grase/CoovaChilli 56
Figura 24 – Confirmação de conexão no Grase/CoovaChilli 57
Figura 25 - RSSI x Vazão no procedimento de validação 58
Figura 26 - Experimento com controle da vazão de Download do ofensor com AP em 802.11g 59
Figura 27 - Experimento com controle da vazão de Download do ofensor com AP em 802.11b 60
Figura 28 - RSSI durante o experimento 4.3.1 (802.11g). 61
Figura 29 - RSSI durante o experimento 4.3.1 (802.11b). 61
Figura 30 - Experimento com controle de vazão de Upload do ofensor 62
Figura 31 - RSSI durante o experimento 4.3.2. 63
Figura 32 – Evolução dos dados a partir do controle da anomalia da MAC (download, IEEE 802.11g) 64
Figura 33 - Evolução dos dados a partir do controle da anomalia da MAC (download, IEEE 802.11b) 65

Figura 34 - Evolução dos dados a partir do controle da anomalia da MAC (upload)..... 66

Figura 35 – Aproximação para dois segmentos de reta em em torno dos momentos de diminuição da taxa de crescimento da vazão total (download, IEEE 802.11g). 67

Figura 36 – Aproximação para dois segmentos de reta em em torno dos momentos de diminuição da taxa de crescimento da vazão total (download, IEEE 802.11b). 68

Figura 37 - Aproximação para dois segmentos de reta em em torno dos momentos de diminuição da taxa de crescimento da vazão total (upload) 68

Figura 38 – Tendência da variação da vazão - download, IEEE 802.11g. 126

Figura 39 – Tendência da variação da vazão - download, IEEE 802.11b. 126

Figura 40 – Tendência da variação da vazão - upload, IEEE 802.11g..... 127

Figura 41 – Índice de eficiência na mitigação da anomalia da MAC, Download- IEEE802.11g. 129

Figura 42 – Índice de eficiência na mitigação da anomalia da MAC, Download- IEEE802.11b. 129

Figura 43 - Índice de eficiência na mitigação da anomalia da MAC, Upload- IEEE802.11g 130

LISTA DE TABELAS

Tabela 1 - Atributos RADIUS "WISPr" *43*

Tabela 2 - Vazão máxima dos Laptops c1 e c2 nos experimentos *63*

Tabela 3 - Vazão inicial dos Laptops c1 e c2 nos experimentos *66*

Tabela 4 - Comparação da vazão no início e final do experimento e no momento de diminuição da taxa de crescimento da vazão total *69*

Tabela 5 – Inclinação antes e depois do momento de diminuição da taxa de crescimento da vazão total. *70*

Tabela 6 – Sensibilidade da vazão (Laptop c1 e Vazão Total) em relação à vazão do Laptop c2 antes e depois do momento de diminuição da taxa de crescimento da vazão total. *71*

Tabela 7 – Equações da tendência da variação da vazão *124*

Tabela 8 – Indice de eficiência na mitigação da anomalia da MAC *127*

Tabela 9 – Valores de vazão e restrição para o instante em que a eficiência ficou abaixo de 1. ..... *131*

LISTA DE ABREVIATURAS E SIGLAS

3G	=	"3rd Generation"
AAA	=	"Authentication, Authorization & Accounting"
ACK	=	"Acknowledgement"
AP	=	"Access Point"
BRAS	=	"Broadband Remote Access Server"
BWM	=	"BandWidth Management"
CoA	=	"Change of Authorization"
CPU	=	"Central Processing Unit"
CSV	=	"Comma Separated Values"
DCF	=	"Distributed Coordination Function"
DHCP	=	"Dynamic Host Configuration Protocol"
GB	=	"Gigabyte"
GGSN	=	"Gateway GPRS Support Node"
GPRS	=	"General Packet Radio Service"
HCF	=	"Hybrid Coordination Function"
HD	=	"Hard Disk"
IEEE	=	"Institute of Electrical and Electronics Engineers"
IEEE-SA	=	"IEEE Standards Association"
IP	=	"Internet Protocol"
Kbps	=	"Kilobits per second"
LAN	=	Local Access Network
MAC	=	"Media Access Control"
Mbps	=	"Megabits per second"
NAS	=	"Network Access Server"
NTP	=	"Network Time Protocol"
OSI	=	"Open Systems Interconnection"
p2p	=	"peer-to-peer"
PCF	=	"Point Coordination Function"
PDA	=	"personal digital assistant"
QoS	=	"Quality of Service"
RADIUS	=	"Remote Authentication Dial In User Service"
RAM	=	"Random Access Memory"

RAS	=	“Remote Access Server”
RF	=	Rádio Frequência
RFC	=	“Request For Comment”
RSSI	=	“Received Signal Strength Indication”
SCP	=	“Secure CoPy”
SFTP	=	“SSH File Transfer Protocol”
SSH	=	“Secure Shell”
UDP	=	“User Datagram Protocol”
UMTS	=	”Universal Mobile Telecommunication System”
VoIP	=	“Voice over IP”
WAN	=	“Wilde Area Network”
WAP	=	“Wireless Application Protocol”
WEP	=	“Wired Equivalent Privacy”
WISPr	=	“Wireless Internet Service Provider Roaming”
WLAN	=	“Wireless LAN”

SUMÁRIO

1. INTRODUÇÃO 17
1.1. Contextualização do Problema 17
1.2. Justificativa Para o Desenvolvimento do Trabalho 17
1.3. Objetivo do Trabalho 18
1.4. Resultados Esperados 19
1.5. Delimitação da Pesquisa 19
1.6. Organização da Dissertação 20
2. REDES SEM FIO IEEE 802.11 21
2.1. Anomalia da MAC 24
2.2. Priorização da Anomalia 25
2.3. Efeitos da Presença de Ofensores 26
3. PROPOSTA DE ARQUITETURA E MÉTODO PARA O CONTROLE DE VAZÃO 28
3.1. “Traffic Shaping” 28
3.2. Arquitetura dos Experimentos 28
3.3. Componentes do Núcleo 30
3.3.1. NAS e “Captive Portal”. 30
3.3.2. AP 32
3.3.3. AAA 33
3.3.4. Gerência de Rede 34
3.3.5. BWM 35
3.4. Componentes dos Usuários 37
3.4.1. Usuários 37
3.4.2. Servidor de arquivos 37
3.5. Protocolos da Arquitetura Proposta 38
3.5.1. RADIUS 38
3.5.2. NTP 41
3.5.3. DHCP 42
3.6. Método de Controle de Vazão 42
3.6.1. Fluxogramas do Controle de Vazão 45
4. METODOLOGIA E AMBIENTE DE TESTES 48
4.1. Bancada de Testes 48
4.1.1. Computador 1 49
4.1.2. Computador 2 50
4.1.3. Laptop c1 50
4.1.4. Laptop c2 50

4.1.5. AP....51
4.1.6. Conexões físicas entre os componentes....51
4.2. Procedimento de Validação....52
4.3. Experimentos....53
4.3.1. Experimento com controle da vazão de Download do ofensor....53
4.3.2. Experimento com controle de vazão de Upload do ofensor....54
5. RESULTADOS OBTIDOS....56
5.1. Procedimento de Validação....57
5.2. Experimentos....58
5.2.1. Experimento com controle da vazão de Download do ofensor....58
5.2.2. Experimento com controle de vazão de Upload do ofensor....62
5.2.3. Análise dos resultados....63
6. CONCLUSÃO....72
7. REFEFÊNCIAS....74
8. ANEXO A – PROCEDIMENTOS DE INSTALAÇÃO/CONFIGURAÇÃO....79
8.1. NAS....79
8.2. AP....83
8.3. Gestor de Autenticação e Largura de Banda....83
8.3.1. 8950AAA - PolicyFlow....83
8.3.2. Scripts utilizados....115
8.3.3. 8950AAA - Aprovisionamento do usuário “teste”....116
9. ANEXO B - DETERMINAÇÃO DA TENDÊNCIA DOS PONTOS COLETADOS....120
10. ANEXO C – ESTUDO DA INCLINAÇÃO DAS CURVAS TENDÊNCIA DE VAZÃO ATRAVÉS DAS DERIVADAS DESSAS CURVAS....124

1. INTRODUÇÃO

1.1. Contextualização do Problema

A utilização de redes locais sem fio (WLAN – “Wireless LAN (Local Area Network)”) está altamente disseminada. O ganho da produção em escala barateou os preços e hoje é grande o número de pontos de acesso sem fio. Uma evidência disso é o número de “hot-spots” existentes hoje no Brasil atingir quase 2500. É grande também o número de dispositivos móveis equipados com acesso às redes sem fio: desde laptops até “Tablets” e telefones celulares do tipo “Smartphone” podem ser equipados com acesso a redes sem fio. Uma evidência da existência de um grande número desse tipo de dispositivos é o de fato que em 2011 foram vendidos 491,4 milhões de smartphones no mundo (TELECO, 2012).

Esse cenário cria condições para a competição pelo meio de transmissão sem fio e permite o aparecimento de um fenômeno conhecido como “anomalia da MAC (“Media Access Control”)” (BRANQUINHO et al., 2006), em que, sob determinadas condições, dispositivos sem fio transmitindo dados a taxas baixas degradam as transmissões dos demais dispositivos conectados ao mesmo “Access Point” (AP) (HEUSSE et al., 2003) (BRANQUINHO et al., 2006).

1.2. Justificativa Para o Desenvolvimento do Trabalho

Estudos no sentido de minimizar essa anomalia alcançaram resultados ao alterar a configuração do AP como um elemento isolado (BRANQUINHO et al., 2006) (GUIRARDELLO, 2008) ou então a forma de classificação do tráfego para o padrão IEEE 802.11e (HYOGON et al., 2005) (FONTOLAN, 2010) (MOTA et al 2011). A implementação destas soluções pode ser dificultada pela necessidade de se alterar uma diversidade muito grande de clientes, ou então pela dificuldade de aplicar a solução somente para um dispositivo móvel. Porém, em todos os

casos, a ideia central se mostrou eficiente: dificultar o tráfego de dispositivos sem fio que estão provocando a anomalia da MAC.

Uma rede sem fio IEEE 802.11 necessariamente está conectada a outros equipamentos que fazem o acesso à Internet e que, por sua vez, podem estar sob a gestão de algum sistema de gerência (VANHATUPA, 2008). Estes equipamentos de acesso podem ter sua gerência integrada à gerência da rede sem fio. Essa integração de gerências pode ser eficiente nas tarefas de identificar quem está causando a anomalia da MAC, assim como mitigar os efeitos dessa anomalia.

A vantagem da integração das gerências é a visão da rede como um todo, com a possibilidade de melhorar a utilização dos recursos disponíveis, em especial o acesso à internet. Numa situação em que a anomalia da MAC está presente, o acesso à internet pode estar subutilizado; combater a anomalia representa melhorar a utilização desse acesso.

1.3. Objetivo do Trabalho

Este trabalho propõe uma arquitetura de integração de gerências sem fio e da rede de acesso que, através de ações dessa gerência integrada, tem a capacidade de mitigar a anomalia da MAC (BRANQUINHO et al., 2006). Não é objetivo desse trabalho a identificação do causador da anomalia, mas o combate aos seus efeitos.

Para isso, foi especificada uma bancada de testes em que estudos de longa duração em redes sem fio podem ser feitos de forma automática. Os componentes dessa bancada utilizam protocolos padronizados e sistemas de controle de tráfego abertos ou comerciais.

Estudos de validação da bancada foram executados e, em seguida, foi feito um estudo de mitigação dos efeitos da anomalia da MAC através da gerência integrada dos elementos da bancada.

1.4. Resultados Esperados

O registro de dados coletados durante uma comunicação sem fio é esperado como resultado da bancada de testes. Os dados coletados devem incluir:

- Tempo de sessão;
- Bytes transferidos (download e upload) e vazão (download e upload);
- RSSI (Received Signal Strength Indication) observado pelo AP para a conexão;
- Limitação da vazão (download e upload).

Nos experimentos feitos, os dispositivos sem fio fizeram uso máximo da vazão no download ou upload (dependendo do experimento). Ao impor um limite para a vazão, o valor observado para a vazão total deve ficar próxima ao valor desse limite, indicando que o limite da vazão é o fator que está controlando o tráfego.

Por meio da comparação da vazão de transmissões sem fio espera-se a comprovação de que limitar a vazão do ofensor pode melhorar a utilização dos recursos de um AP.

1.5. Delimitação da Pesquisa

Neste trabalho, é empregado o controle de tráfego por meio do “Traffic Shaping” do sistema CoovaChilli (CoovaChilli, 2011). Para a coleta de informações de tráfego, foram utilizadas mensagens RADIUS de “accounting”; essas mensagens foram enviadas pelo CoovaChilli, que foi configurado para enviar uma mensagem “accounting-Request” a cada minuto.

Os testes foram feitos com um número restrito de clientes sem fio (dois) e todos utilizaram o sistema operacional Windows (Windows XP e Windows 7 SE).

Um dos dispositivos não tinha capacidade de processamento (*CPU*) suficiente para suportar alto tráfego sem fio com criptografia WAP (“Wireless Application Protocol”) e isso limitou os testes para a utilização da criptografia WEP (“Wired Equivalent Privacy”).

Além disso, esse trabalha assume que o ofensor em uma rede Wi-Fi já tenha sido detectado. Os experimentos partem do fato que que o ofensor já é conhecido e verificam a eficária da arquitetura na mitigação dos efeitos de sua presença.

1.6. Organização da Dissertação

O Capítulo 1 dessa dissertação – INTRODUÇÃO – traz a contextualização do trabalho, as justificativas para seu desenvolvimento e os objetivos do trabalho. Os resultados esperados também são colocados nesse capítulo.

O Capítulo 2 - REDES SEM FIO IEEE 802.11 – expõe a Anomalia da MAC, a Priorização da Anomalia e os Efeitos da Presença de Ofensores.

O Capítulo 3 - PROPOSTA DE CONTROLE DE VAZÃO – define os conceitos de Traffic Shaping, a Arquitetura dos Experimentos, os Componentes do Núcleo da arquitetura do experimento (NAS, AAA, “Captive Portal”, AP e BWM), os Componentes de Usuários (Laptops e Servidor), os Protocolos da Arquitetura Proposta (RADIUS, NTP e DHCP) e o Método de Controle de Vazão.

O Capítulo 4 - METODOLOGIA E AMBIENTE DE TESTES – define os Componentes da Bancada de Testes (Computador 1, Computador 2, Laptop c1, Laptop c2, conexões) e o Procedimento de Validação e Experimentos.

O Capítulo 5 descreve os Resultados Obtidos e a Análise dos resultados.

Finalmente, o Capítulo 6 explicita a Conclusão deste trabalho.

2. REDES SEM FIO IEEE 802.11

Considera-se rede sem fio aquela em que dispositivos, tais como laptops e PDAs (“Personal Digital Assistant”), tem acesso à Internet (ou alguma outra rede de dados) através de alguma tecnologia de acesso a rede de dados sem fio. Uma das tecnologias mais difundidas atualmente é a Wi-Fi (que permite a transmissão sem fio de pacotes de dados). O meio de transmissão é um canal RF (Rádio Frequência) com frequência e largura de banda definidas pelo padrão IEEE 802.11 (VANHATUPA, 2008).

Redes Wi-Fi também são conhecidas como redes IEEE 802.11, que é um conjunto de padrões para acesso a redes sem fio mantido pelo IEEE Standards Association (IEEE-SA, 2012).

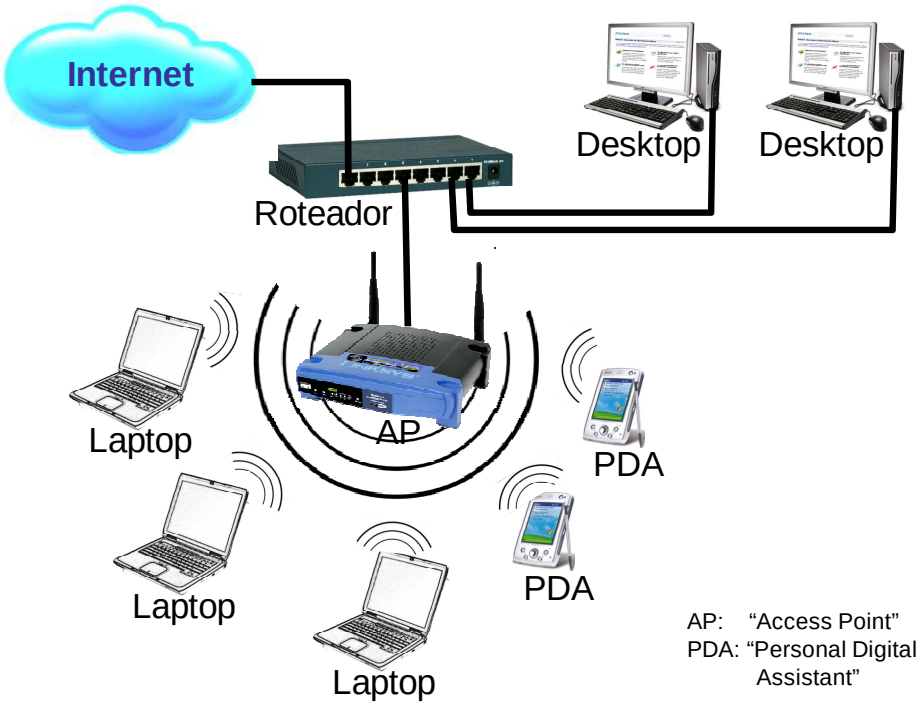


Figura 1 - Exemplo de rede sem fio

Uma rede Wi-Fi típica (ROSHAN; LEARY, 2003) pode ser vista na Figura 1, onde é possível ver o AP (Access Point) e os dispositivos sem fio. Nessa figura, é possível perceber que o AP está conectado ao roteador e este dá o acesso à

Internet. É bastante comum equipamentos que reúnam as funcionalidades de AP e roteador, porém funcionalmente são entidades distintas.

A Figura 2 permite visualizar as camadas do modelo de referência OSI (“Open Systems Interconnection”) com ênfase para alguns componetes de redes IEEE 802.11: MAC, “Distributed Coordination Function” (DCF) e “Point Coordination Function” (PCF) (ROSHAN; LEARY, 2003). É possível perceber que o padrão IEEE 802.11 diz respeito às camadas 1 e 2. Os equipamentos responsáveis por essas camadas em uma transmissão de dados são chamados de “Access Point” (AP). Os equipamentos responsáveis pela camada 3 são conhecidos como reteadores; em alguns casos os roteadores são responsáveis pela camada 4 também.

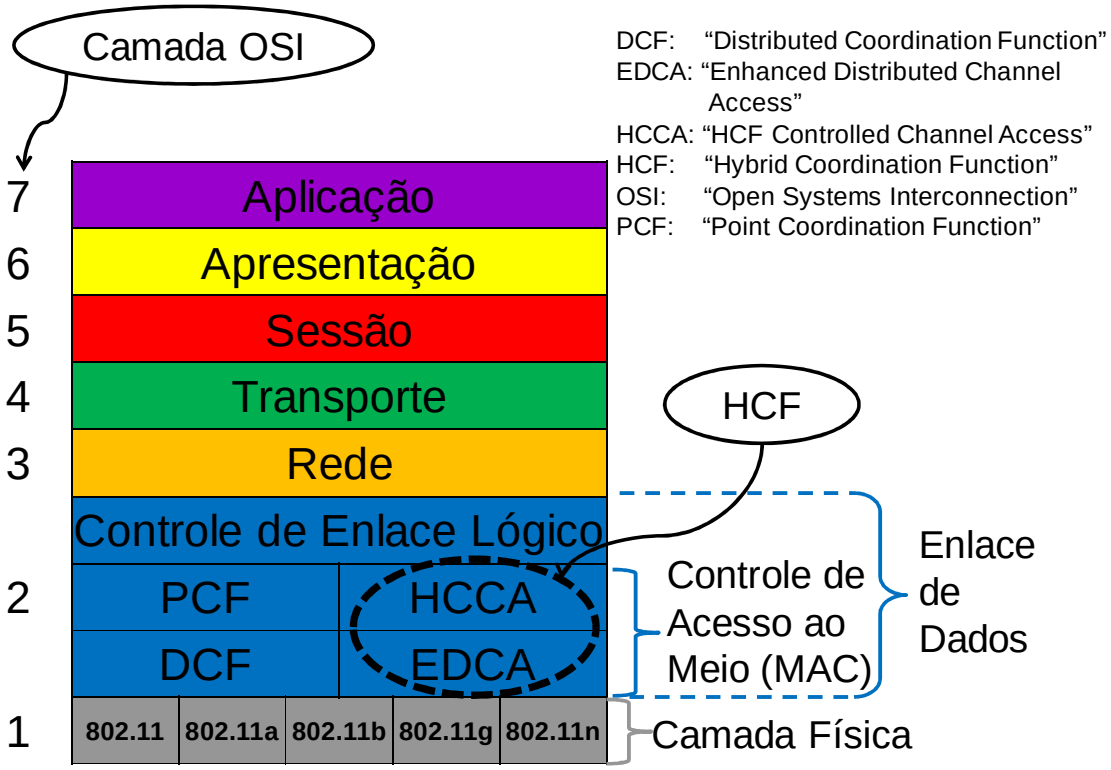


Figura 2 - Localização da MAC, DCF e PCF no modelo de referência OSI

A Figura 1 mostra as camadas 1 (os sinais sem fio), 2 (o AP) e 3 (roteador) de uma conexão sem fio típica de acesso à internet.

O acesso ao meio de transmissão estabelecido pelo IEEE 802.11 é disciplinado por um conjunto de regras conhecido como MAC ou, simplesmente, “camada MAC” (IEEE Std 802.11, 2007).

A camada MAC de redes IEEE 802.11 provê mecanismos de controle de acesso que permitem o compartilhamento do meio de transmissão por vários usuários. A técnica básica utilizada para esse controle é a DCF (IEEE Std 802.11, 2007).

A DCF prevê que uma estação que deseja iniciar a transmissão observa um canal de status, por um período de tempo pré-estabelecido, para se certificar que o meio de transmissão está desocupado. Só então a estação tem permissão de transmitir um pacote de dados. Durante a transmissão, um indicador de status indica que o canal está ocupado. Após a transmissão, a estação espera por uma confirmação de recepção (ACK) do receptor; se o ACK não chegar após um tempo pré-definido, há a retransmissão do pacote de dados. Para evitar colisões, as estações verificam o status de ocupação antes de iniciar o protocolo de ocupação de canal. Se o status indica que o canal está ocupado, estas estações aguardam um tempo aleatório fazem uma nova verificação.

Além do DCF, a camada MAC prevê uma técnica opcional chamada PCF. Através do PCF, o AP desempenha o papel de gerenciador de acesso ao canal de comunicação dentro de uma rede Wi-Fi. A ideia básica do PCF é que o AP envia para cada um das estações conectadas ao AP um pacote de dados, indicando que a estação tem permissão para transmitir. Caso a estação com permissão não tenha dados para transmitir, ela envia um quadro nulo. É importante observar que nem toda estação implementa o PCF, uma vez que este é opcional.

Os protocolos DCF e PCF podem ser substituídos pelo HCF (“Hybrid Coordination Function”), como definido pelo padrão IEEE 802.11e (IEEE Std 802.11e, 2005). Porém, as estações não são obrigadas a honrá-lo. Por conta disso, este trabalho baseia-se num AP com DCF e PCF.

2.1. Anomalia da MAC

Os protocolos de controle de acesso à MAC (DCF e PCF) garantem para todas as estações móveis o acesso ao meio de transmissão em igualdade de condições (sem levar em consideração para isso as condições de propagação do sinal Wi-Fi das estações). Isso significa que uma estação em más condições de propagação RF tem a mesma chance de conseguir o acesso ao canal RF que uma estação em ótimas condições (BRANQUINHO et al., 2006).

Porém, uma estação em más condições de propagação transmite seus dados a uma taxa de transmissão baixa (conforme define o padrão IEEE 802.11) e, por conta disso, fica com o canal RF alocado por mais tempo (em comparação com outra estação com taxa mais alta e que esteja transmitindo a mesma quantidade de dados).

Dependendo das condições de transmissão do AP e das estações sem fio conectadas a ele, é possível que haja situações em que um canal fique alocado a maior parte do tempo para taxas de transmissões baixas, prejudicando as transmissões em altas taxas.

Dependendo da quantidade de informação que as estações em boas condições de propagação desejam transmitir, os dados dessas estações podem ficar represados por conta da existência de estações em baixas taxas (por conta de más condições de propagação). Isso se configura no que é conhecido por “Anomalia da MAC”.

Essa “Anomalia da MAC” foi descrita, inicialmente, por (HEUSSE et al., 2003). Posteriormente, houve estudos para mitigar seus efeitos, como em (BRANQUINHO et al., 2006), (GUIRARDELLO, 2008) e (FONTOLAN, 2010). A Figura 3 reproduz essa situação, na qual há possibilidade de ocorrer a anomalia da MAC, conforme descrito posteriormente no item 2.3 deste documento.

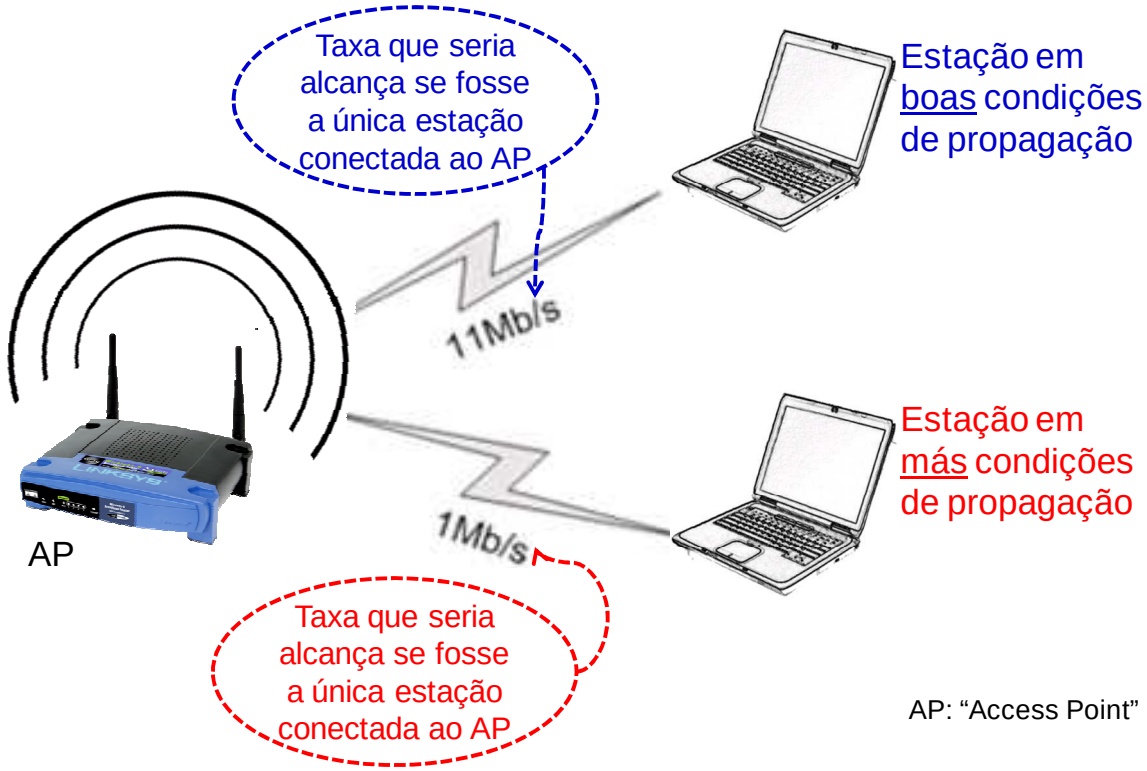


Figura 3 – Situação com possibilidade de ocorrer a anomalia da MAC

Neste trabalho, define-se como ofensor uma estação sem fio em más condições de propagação que está prejudicando outra estação sem fio em boas condições de propagação. Entende-se aqui “prejudicar” por diminuir a capacidade de transmissão/recepção da estação.

Estudos mostram que em situações em que a anomalia da MAC está acontecendo, a taxa disponível para cada estação fica próxima à taxa mais baixa (BRANQUINHO et al., 2006). Esse fenômeno, além de prejudicar os usuários de uma rede Wi-Fi, prejudica também o provedor do acesso, que tem o recurso colocado à disposição dos usuários subaproveitado (com possíveis consequências no faturamento desse provedor).

2.2. Priorização da Anomalia

O padrão IEEE 802.11e (IEEE Std 802.11e, 2005) foi definido para implementar QoS (“Quality of Service”) em redes Wi-Fi. Este padrão define que os pacotes de dados são pré classificados (como vídeo, voz etc) pelas camadas

superiores e priorizados pela MAC, privilegiando o tráfego pré-classificado na alocação do canal de transmissão. É importante salientar que as condições de propagação das estações não são levadas em consideração na priorização da alocação.

Assim, pode ocorrer a priorização do tráfego que provoca a anomalia. Essa combinação potencializa a ocorrência da anomalia da MAC e, consequentemente, amplia seus efeitos (FONTOLAN, 2010).

Os dispositivos Wi-Fi não são obrigados a implementar o padrão IEEE 802.11e. Por conta disso, esse padrão não será utilizado nesse trabalho.

2.3. Efeitos da Presença de Ofensores

Na Figura 3 (vista anteriormente no item 2.1), a anomalia da MAC ocorreria se o AP tivesse a capacidade de tráfego na WAN de 11 Mbps (além das duas estações em diferentes condições de propagação do sinal Wi-Fi: 1 Mbps para a primeira estação e 11 Mbps para a segunda). Nessas condições, a primeira estação ocupa o meio de transmissão por um tempo 11 vezes maior que a segunda estação para transmitir a mesma quantidade de dados.

Se as duas estações têm demandas de tráfego suficientes (cada uma delas individualmente) para ocupar todo o tempo de canal (e levando-se em consideração que a MAC dá chances iguais para as duas estações de alocar o meio de transmissão), é possível que o tráfego da segunda estação seja até menor que o tráfego da primeira estação. Essa situação é bem estudada em (FONTOLAN, 2010).

Nesse cenário, a primeira estação é a ofensora e a segunda estação é a ofendida. Uma vez que a taxa da segunda estação está limitada a 1 Mbps, o tráfego máximo no AP é de 2 Mbps (bem menor que a capacidade teórica, que é de 11 Mbps). Essa situação foi demostrada em (GUIRARDELLO, 2008) e pode ser vista na Figura 4.

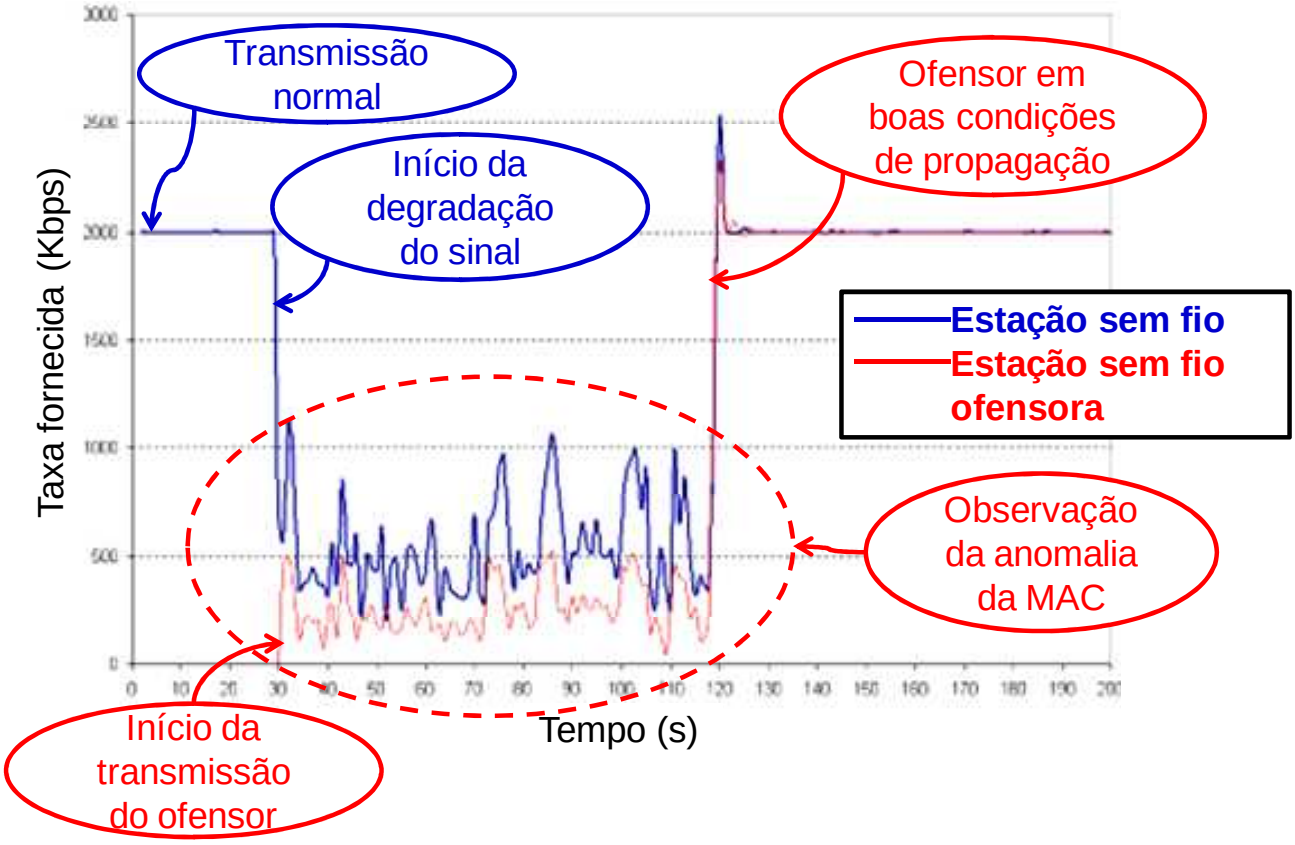


Figura 4 – Constatação da anomalia da MAC. Fonte: (GUIRARDELLO, 2008)

Nessa figura, uma das estações está transmitindo a 2000 Kbps (curva em azul) e tem sua vazão diminuída para ~500 Kbps logo que a segunda estação (curva em vermelho), com taxa de transmissão baixa, inicia a transmissão (em Tempo=30s). Isto indica que a presença de ofensores em redes Wi-Fi provoca o subaproveitamento dos recursos da rede sem fio. Isso faz com que os custos da rede sejam maiores e os gastos com energia também sejam maiores.

3. PROPOSTA DE ARQUITETURA E MÉTODO PARA O CONTROLE DE VAZÃO

3.1. “Traffic Shaping”

O termo “Traffic Shaping” será utilizado neste trabalho como o processo de controle de vazão do tráfego IP (“Internet Protocol”) de um usuário.

É possível definir “Traffic Shaping” incluindo também a priorização de tráfego. Isso permitiria a inibição de protocolos do tipo p2p (“peer-to-peer”) ou VoIP (“Voice over IP”).

Porém, nesse trabalho, não foi utilizado “Traffic Shaping” com priorização de protocolo. Isso significa que todos os protocolos terão a mesma chance de serem trafegados. O “Traffic Shaping” desse trabalho foi baseado exclusivamente na vazão do tráfego IP do usuário ultrapassar ou não um determinado limite.

3.2. Arquitetura dos Experimentos

A Figura 5 mostra a arquitetura utilizada nos experimentos realizados. Essa arquitetura inclui equipamentos que representam o núcleo desse trabalho, assim como equipamentos que representam os usuários, ou seja, os geradores e receptores de tráfego de uma rede de acesso.

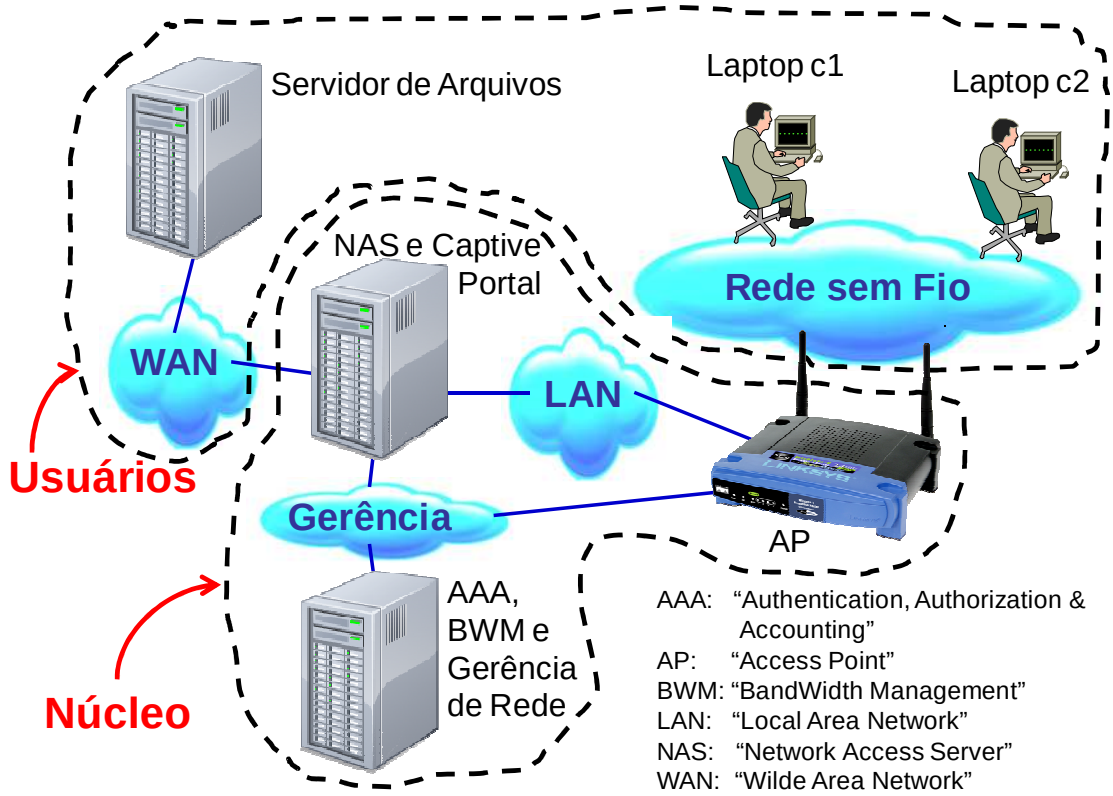


Figura 5 - Arquitetura utilizada nos experimentos

A área delimitada "Usuários" representa a Internet e usuários de uma rede Wi-Fi. A utilização de aplicações (por exemplo, um arquivo baixado pelo Laptop c1 e armazenado no servidor de arquivos) faz com que informações sejam trafegadas através do núcleo.

A área delimitada "Núcleo" é responsável por fazer todo o controle do tráfego dos usuários. Entende-se por controle, a gerência e a transmissão efetiva da informação entre os usuários.

Existem diversas formas de gerar tráfego na rede sem fio, Nessa arquitetura, os experimentos tiveram o tráfego gerado das seguintes formas:

1. Download de Arquivos.
2. Upload de Arquivos.

Essa forma de gerar o tráfego é interessante, pois gera uma demanda constante. Demandas por tráfego com taxas variáveis poderiam comprometer as medidas realizadas, já que poderia haver períodos de baixo tráfego em que a

restrição à vazão imposta pelo NAS (“Network Access Server”) não teria efeito sobre o tráfego dos usuários. A ideia da arquitura dos experimentos é que o NAS seja o ponto que determina a vazão dos usuários.

3.3. Componentes do Núcleo

O núcleo compõe-se de 5 entidades funcionais:

- NAS e “Captive Portal”: Network Access Server.
- AP: Access Point.
- AAA: “Authentication, Authorization & Accounting”.
- BWM: Bandwidth Management.
- Gerência de Rede.

Neste trabalho, estas entidades funcionais estão agrupadas em 1 servidor; porém, nada impede que residam em servidores distintos.

3.3.1. NAS e “Captive Portal”.

O conceito de NAS foi formalmente definido em (RFC 2881, 2000) como sendo o ponto de entrada inicial de uma rede para usuários de serviços de rede. Além disso, o NAS é um gateway para os demais serviços da rede.

Dentro da topologia do protocolo AAA RADIUS (RFC2865, 2000) (RFC2866, 2000), que é um protocolo do tipo Cliente-Servidor, o NAS desempenha o papel de Cliente dos serviços oferecidos pelo servidor AAA. Isso significa que o NAS solicita autenticação/autorização de usuários que desejam ter acesso aos serviços de rede e também solicita o registro detalhado das sessões de dados dos usuários que utilizam seus serviços de acesso à rede.

Tipicamente, o equipamento que desempenha o papel de NAS é um roteador e, dependendo da tecnologia de acesso, pode ter outros nomes dentro do ambiente dessa tecnologia. Exemplos desses outros nomes são:

- RAS (“Remote Access Server”) para redes de acesso por linha discada;
- BRAS (“Broadband Remote Access Server”) para redes de acesso DSL (“Digital Subscriber Line”);
- GGSN (“Gateway GPRS Support Node”) para redes de acesso UMTS (“Universal Mobile Telecommunication System”), também conhecido como 3G (“3rd Generation”).

Uma visão pictórica do NAS na arquitetura RADIUS pode ser vista na Figura 6.

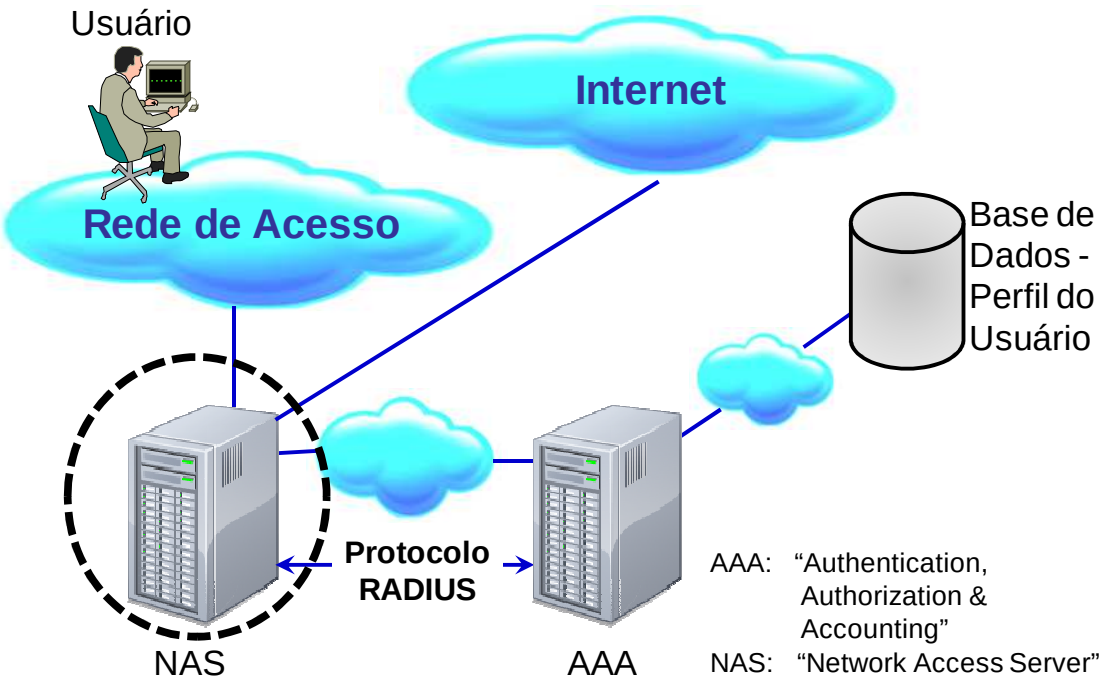


Figura 6 - NAS na arquitetura RADIUS

Neste trabalho, para desempenhar o papel de NAS, foi utilizado o pacote CoovaChilli (COOVACHILLI, 2011) foi utilizado. Este pacote está disponível para a versão do Linux-Ubuntu (UBUNTU, 2011) utilizado. Por conta disso, não foi necessário fazer qualquer adaptação ao pacote já disponível na distribuição Linux. Ao instalar o pacote CoovaChilli em um computador com 3 interfaces de rede, esse computador se transforma num roteador com capacidade de desempenhar, entre outros, os papeis de NAS e de “Captive Portal”.

A característica principal que levou à escolha do CoovaChilli para os papeis de NAS e de “Captive Portal” é o fato de ele implementar “Traffic Shaping” em conjunto com os atributos RADIUS WISPr (ANTON et al., 2003) e a mensagem RADIUS CoA (RFC 5176, 2008). Assim, foi possível alterar a vazão permitida para um determinado usuário conectado ao NAS através dos atributos WISPr (sem reiniciar a sessão desse usuário).

Essa alteração dinâmica da vazão de um usuário permite o controle de tráfego que passa pelo NAS por parte do BWM via protocolo RADIUS.

O pacote GRASE Hotspot (GRASE, 2011) foi utilizado para configurar o “Captive Portal” do CoovaChilli. Esse pacote tem total integração com o CoovaChilli e foi utilizado devido à facilidade de instalação.

Neste trabalho, o conceito de “Captive Portal” utilizado foi o de que consiste em um sistema que captura a navegação web do usuário e a redireciona para um Portal Web. Esse Portal Web solicita dados de autenticação do usuário (nome e senha) e submete esses dados ao servidor AAA. A navegação somente será liberada após a autenticação com sucesso por parte do servidor AAA.

3.3.2. AP

O AP (“Access Point”) implementa a interface sem fio padrão IEEE 802.11.

É importante observar que para a bancada de testes, esse equipamento não deve acumular as funções de roteador, mas somente a função de AP. Todas as funções de roteador são desempenhadas pelo NAS.

O AP pode ser visto como um conversor de mídia entre a rede sem fio Wi-Fi e a conexão FastEthernet entre o AP e o NAS. Isso é importante para a bancada de testes, pois solicitações de endereço IP dos Laptops serão encaminhadas para o NAS.

O AP deve também permitir a conexão Telnet por parte de algum outro componente. Através dessa conexão, é possível a coleta de informações das conexões ativas, em especial o RSSI (“Received Signal Strength Indication”) percebido pelo AP.

3.3.3. AAA

A sigla AAA significa “Authentication, Authorization & Accounting” e é utilizada neste trabalho como uma referência ao servidor que permite a autenticação, autorização e o registro de detalhes das sessões de dados dos experimentos realizados.

Não há uma definição formal de servidor AAA. Neste trabalho, define-se servidor AAA como o servidor que implementa o protocolo RADIUS (RFC 2865, 2000) dentro da arquitetura Cliente-Servidor proposta por esse protocolo.

Essa definição de servidor AAA é compatível com o conceito de NAS proposto anteriormente (Figura 7).

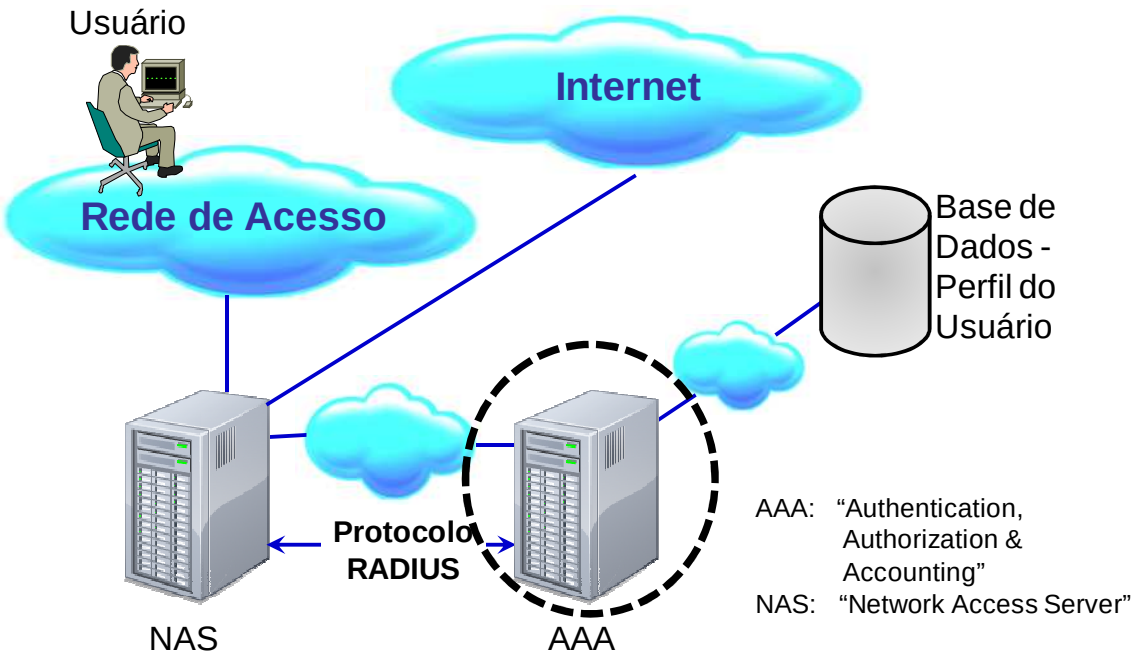


Figura 7 - AAA na arquitetura RADIUS

Um servidor AAA deve executar pelo menos as seguintes tarefas (RFC 2903, 2000):

- Ao receber uma mensagem RADIUS de autenticação, obtém o perfil do usuário na base de dados e processa a autenticação do usuário.
    - Verifica também os demais dados do usuário para verificar se o usuário está autorizado a ter a sessão.

- Ao receber uma mensagem RADIUS de “accounting”, registra as informações contidas na mensagem num repositório pré-estabelecido (arquivo texto, banco de dados etc.).

Vários softwares tem a capacidade de funcionar com um servidor AAA. Dentre eles pode-se destacar:

- FreeRADIUS (FreeRADIUS, 2012)
- 8950AAA (8950AAA, 2011)

Para este trabalho foi escolhido o 8950AAA para desempenhar o papel de AAA por conta da sua capacidade de implementar uma lógica de programação para cada mensagem RADIUS que chega ao servidor. Essa lógica de programação é escrita na linguagem “PolicyFlow” (8950AAA, 2011) e funciona como um software que transforma o servidor AAA num ponto inteligente da arquitetura RADIUS. No item “8.3.1 - 8950AAA - PolicyFlow” está o “PolicyFlow” que foi desenvolvido para este trabalho.

3.3.4. Gerência de Rede

A Gerência de Rede pode ser definida de diferentes maneiras (VANHATUPA, 2008). Neste trabalho, o conceito de Gerência de Rede utilizado é: a entidade que obtém informações de tráfego dos usuários e as disponibiliza para outras entidades. Além disso, gráficos de utilização e status podem ser obtidos com esses dados.

O PolicyFlow desenvolvido no 8950AAA faz o registro das informações de gerência a partir das mensagens de autenticação e de “accounting” recebidas do NAS. Isso significa que, neste trabalho, a Gerência de Rede é parte do PolicyFlow do 8950AAA e, portanto, está contido no servidor AAA (como pode ser visto na Figura 8). A lógica da Gerência de Rede será explicada mais adiante, no item “3.6 - Método de Controle de Vazão”.

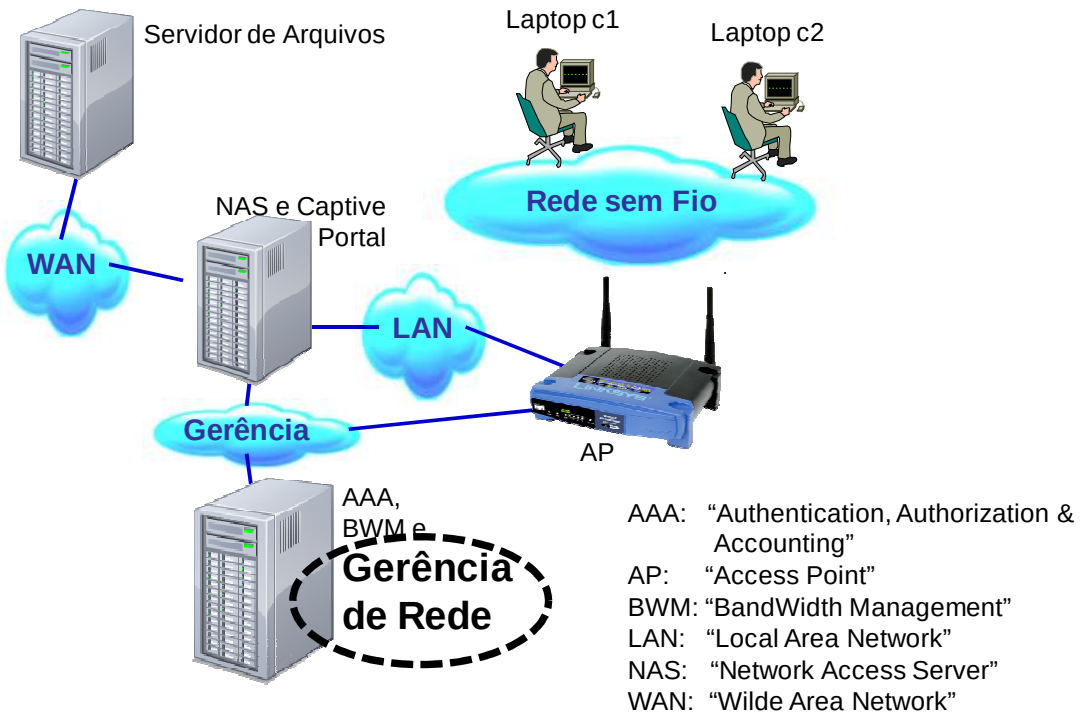


Figura 8 – Gerencia de Rede na arquitetura utilizada nos experimentos

Além das mensagens de "accounting", a Gerência de Rede coleta informações de RSSI das conexões dos usuários executando comandos por meio de linha de comando disponibilizada pelo firmware do AP (DD-WRT, 2011). A lógica dessa coleta também está contida no PolicyFlow desenvolvido, como pode ser observado do item "8.3.1 - 8950AAA - PolicyFlow".

A Gerência de Rede armazena todas as informações coletadas em arquivos do tipo CSV ("Comma Separated Values") e isso permite que essas informações sejam utilizadas para gerar gráficos de utilização da rede (conforme será feito no capítulo "5 - Resultados Obtidos"). Essas informações também são disponibilizadas internamente na memória do "PolicyFlow" para o BWM.

3.3.5. BWM

O conceito de BWM (ou BandWidth Management) utilizado neste trabalho é: a entidade que, de acordo com critérios pré-estabelecidos, faz o controle da vazão permitida aos usuários conectados ao NAS. Algumas vezes, a literatura propõe uma arquitetura em que o BWM aparece associado a outra entidade chamada "Call Admission Control" (NIYATO; HOSSAIN, 2007). Este é exatamente

o caso deste trabalho, em que o BWM reside no mesmo servidor do AAA e tem uma relação bastante próxima com ele (como será visto mais adiante). O servidor AAA da arquitetura proposta neste trabalho poderia ter a função ampliada para fazer controle de admissão (em que usuários poderiam ser bloqueados pelo fato do AP estar congestionado), porém isso não foi feito porque essa função não é necessária para os experimentos.

O PolicyFlow desenvolvido para o 8950AAA analisa os dados coletados pela Gerência de Rede e toma a decisão de enviar comandos ao NAS para alterar a vazão permitida aos usuários. Isso significa que, neste trabalho, o BWM é parte do PolicyFlow do 8950AAA e, portanto, está contido no servidor AAA (como pode ser visto na Figura 9). A lógica do BWM será explicada mais adiante, no item “3.6 - Método de Controle de Vazão”.

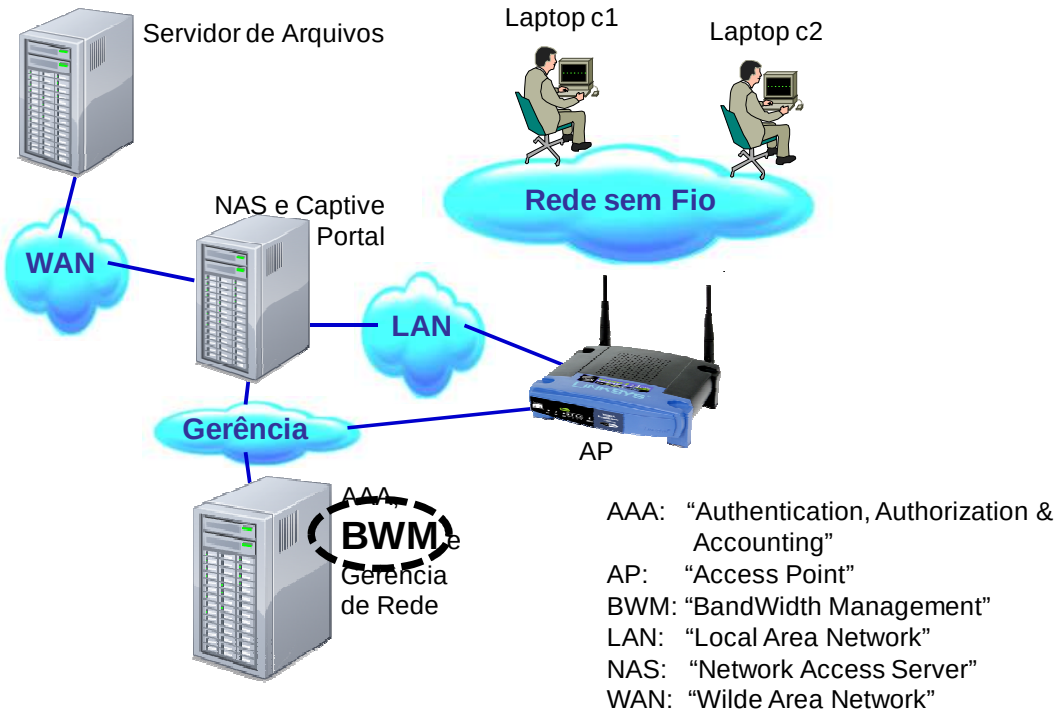


Figura 9 – BWM na arquitetura utilizada nos experimentos

A comunicação entre o BWM e o NAS é feita por meio de mensagens RADIUS CoA (RFC 5176, 2008). Como será visto mais adiante, essas mensagens CoA contêm atributos WISPr (ANTON et al., 2003) para indicar a vazão permitida para os usuários.

3.4. Componentes dos Usuários

Os componentes da região delimitada “Usuários” utilizam os serviços oferecidos pelo núcleo. Estes componentes podem ser divididos em servidor e clientes (ou usuários).

Esta divisão entre servidor e clientes corresponde ao que se vê normalmente em ambiente comercial. Os usuários, suas residências utilizam os serviços de algum servidor (por exemplo, net-banking). O acesso desses usuários ao servidor ocorre via Internet (aqui representada pelo núcleo).

3.4.1. Usuários

São representados por laptops conectados ao Núcleo (mais especificamente ao AP via IEEE 802.11). Esses Laptops foram configurados para obterem endereço IP automaticamente via protocolo DHCP.

Os usuários fazem cópias de arquivos de forma a gerar tráfego que passa pelo núcleo.

3.4.2. Servidor de arquivos

O servidor de arquivos fica conectado logicamente com os usuários. Isso significa que haverá troca de informações entre o servidor e os usuários; essas informações sempre vão passar pelo núcleo.

É importante observar que a capacidade do meio de transmissão (em termos de largura de banda) entre o NAS e o servidor de arquivos e de vídeo deve ser superior à capacidade da interface Wi-Fi entre o AP e os laptops. Com isso, o tráfego do experimento será limitado pela interface aérea ou pelo NAS. Isso é alcançado através de uma conexão dedicada entre o Servidor e o NAS.

3.5. Protocolos da Arquitetura Proposta

3.5.1. RADIUS

O protocolo RADIUS é um protocolo de AAA padronizado por (RFC 2865, 2000) (RFC 2866, 2000): são as funções de Autenticação, Autorização e “accounting” (registro de informações detalhadas das sessões de dados). Por meio desse protocolo, o núcleo (mais especificamente o AAA) sabe quais são as sessões ativas no NAS.

O protocolo RADIUS é do tipo Cliente-Servidor, em que o Cliente é o NAS e o Servidor é o servidor AAA. As mensagens RADIUS são transportadas em pacotes UDP (portas padrão 1812 e 1813 para autenticação e “accounting”, respectivamente).

Genericamente, a autenticação de uma sessão de dados vai permitir, ou não, ao usuário estabelecer essa sessão; essa permissão é normalmente baseada em usuário/senha. A autorização ocorre após a autenticação e verifica se existe alguma restrição (por exemplo, é possível que o usuário possa se conectar somente em algumas faixas de horário). A Figura 10 mostra a sinalização básica de uma autenticação/autorização com RADIUS.

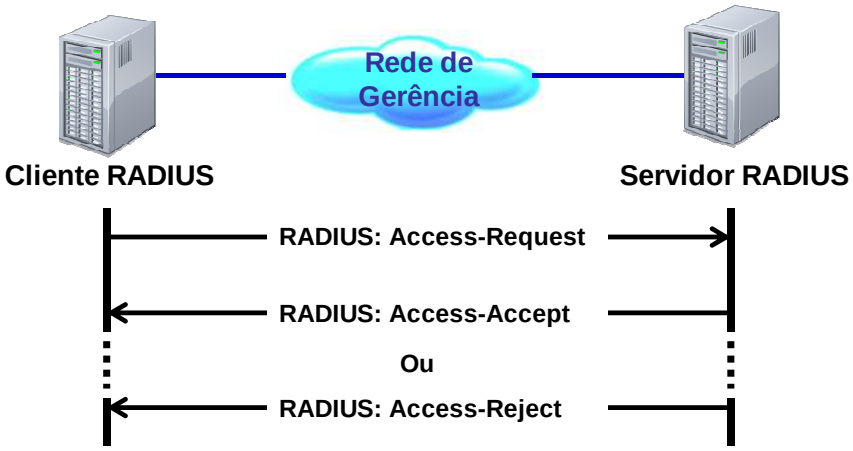


Figura 10 - Autenticação/Autorização com RADIUS

A mensagem RADIUS Access-Request (RFC 2865, 2000) deve conter dados suficientes para que o servidor AAA processe a autenticação e a

autorização. Já a mensagem Access-Accept deve conter dados suficientes para que o NAS estabeleça a sessão de dados (exemplo: endereço IP a ser atribuído para a sessão). A Figura 11 e a Figura 12 contêm exemplos de dados contidos em mensagens RADIUS Access-Request e na correspondente mensagem Access-Accept.

```
ChilliSpot-Version = "1.2.6"
User-Name = "teste"
CHAP-Challenge = 44534A7CB41C8F0862F14FF34A627766
CHAP-Password = "\$009424133D0245AB461EBE890A7829E076"
Service-Type = Login-User
Acct-Session-Id = "4dcbd88400000002"
Framed-IP-Address = 10.1.0.3
NAS-Port-Type = Wireless-IEEE-802-11
NAS-Port = 2
NAS-Port-Id = "00000002"
Calling-Station-Id = "00-0C-29-0C-9D-C5"
Called-Station-Id = "00-0C-29-3C-09-07"
NAS-IP-Address = 10.1.0.1
NAS-Identifier = "nas01"
WISPr-Location-ID = "isocc=,cc=,ac=,network=Grase,"
WISPr-Location-Name = "GRASE_HotSpot"
WISPr-Logoff-URL = "http://10.1.0.1:3990/logoff"
Message-Authenticator = "133926ED588BA33D5F15A09D99F9515A"
```

Figura 11 - Exemplo de mensagem Access-Request

```
Service-Type = Framed-User
Framed-Protocol = PPP
Framed-IP-Address = 255.255.255.254
Framed-IP-Netmask = 255.255.255.255
Framed-Routing = None
Framed-Compression = Van-Jacobson-TCP-IP
Idle-Timeout = 1200
WISPr-Bandwidth-Max-Up = 50000
WISPr-Bandwidth-Max-Down = 50000
```

Figura 12 - Exemplo de mensagem Access-Accept

Após a autenticação e a autorização (ou seja, após o cliente RADIUS receber o Access-Accept), requisições de “accounting” (RFC 2866, 2000) são enviadas para o servidor RADIUS. Essas requisições contêm informações

detalhadas com o status naquele instante da sessão de dados. A Figura 13 mostra a sinalização básica de “accounting” RADIUS.

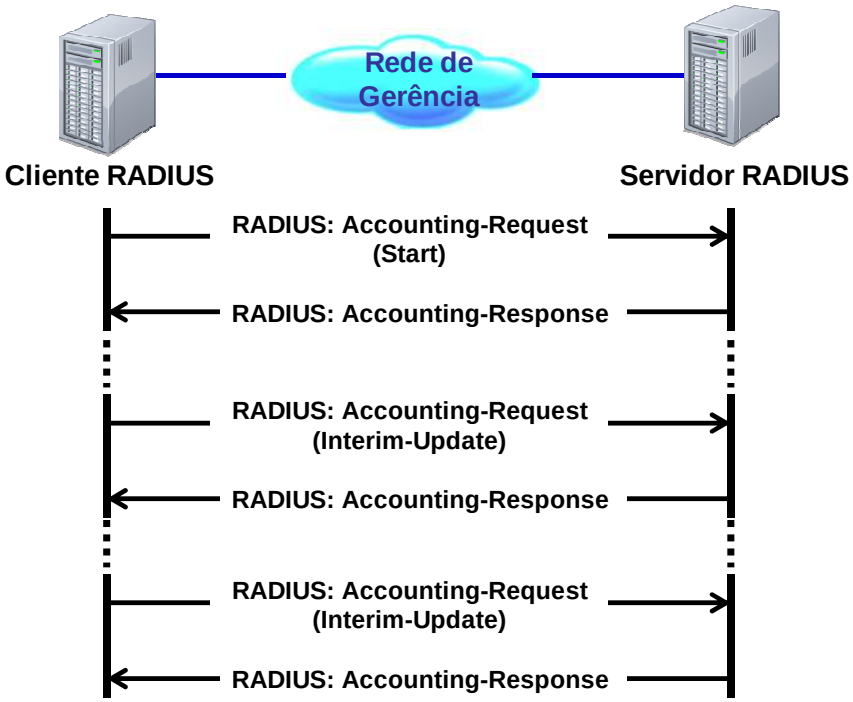


Figura 13 - “accounting” RADIUS

Geralmente, o servidor AAA registra o conteúdo das mensagens de “accounting” em um repositório específico para posterior processamento. Esse processamento posterior pode incluir funções de tarifação e, nesse caso, esse processo é chamado de tarifação off-line. No caso desse trabalho, as mensagens de “accounting” são utilizadas para registrar o histórico das sessões, já que contém informações de tráfego e de tempo de sessão. A Figura 14 contém um exemplo de dados contidos em uma mensagem “accounting-Request”.

```
Sex Jan 20 15:02:08 2012
        ChilliSpot-Version = "1.2.6"
        ChilliSpot-Acct-View-Point = Client-View-Point
        Event-Timestamp = "2011/05/12 10:26:44"
        Acct-Status-Type = Interim-Update
        User-Name = "teste"
        Acct-Input-Octets = 216942
        Acct-Output-Octets = 66766
        Acct-Input-Gigawords = 0
        Acct-Output-Gigawords = 0
        Acct-Input-Packets = 437
        Acct-Output-Packets = 476
        Acct-Session-Time = 1202
        Acct-Session-Id = "4dcbd88400000002"
        Framed-IP-Address = 10.1.0.3
        NAS-Port-Type = Wireless-IEEE-802-11
        NAS-Port = 2
        NAS-Port-Id = "00000002"
        Calling-Station-Id = "00-0C-29-0C-9D-C5"
        Called-Station-Id = "00-0C-29-3C-09-07"
        NAS-IP-Address = 10.1.0.1
        NAS-Identifier = "nas01"
        WISPr-Location-ID = "isocc=,cc=,ac=,network=Grase,"
        WISPr-Location-Name = "GRASE_HotSpot"
```

Figura 14 – Exemplo de mensagem de “accounting”

3.5.2. NTP

NTP (“Network Time Protocol”) (RFC 5905, 2010) (NTP, 2011) é um protocolo utilizado para sincronizar o relógio interno de equipamentos (clientes NTP) com um servidor de relógio (servidor NTP) centralizado. A utilização desse protocolo garante que todos os clientes estejam com os relógios sincronizados.

O NTP foi implementado em todos os servidores para garantir o sincronismo e coerência na coleta e no registro dos dados de tráfego. Como parte da arquitetura do NTP, o NAS foi configurado para desempenhar a função de servidor NTP e todos os demais componentes foram configurados para serem clientes NTP. Com isso, os equipamentos utilizados nos experimentos tiveram seus relógios sincronizados.

3.5.3. DHCP

DHCP (“Dynamic Host Configuration Protocol”) (RFC 2131, 1997) é um protocolo que permite, entre outras coisas, a atribuição de endereço IP a clientes DHCP. Tipicamente um cliente DHCP é um computador conectado a uma rede de acesso. Após a conexão do computador a uma rede, esse computador envia uma requisição de endereço IP dentro de uma mensagem DHCPDISCOVER. A resposta, contendo o endereço IP, é a mensagem DHCPOFFER. A Figura 15 mostra essa utilização do DHCP.

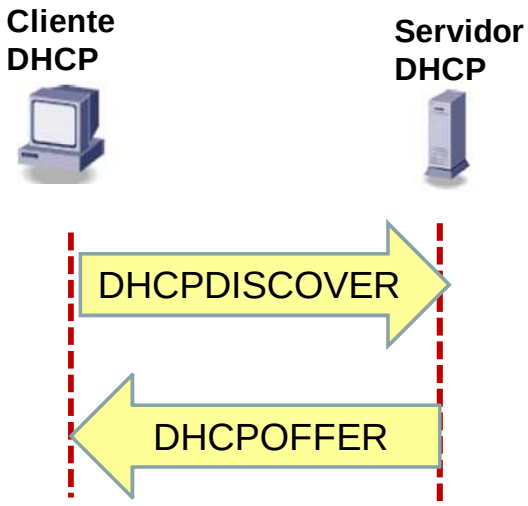


Figura 15 – Protocolo DHCP

Na arquitetura de experimentos proposta por esse trabalho, o NAS faz o papel de Servidor DHCP e os Laptops fazem o papel de Clientes DHCP. Dessa forma, o NAS mantém, em sua base de dados, quais são as sessões de dados conectadas ao AP.

3.6. Método de Controle de Vazão

A referência (RFC 5176, 2008) propõe a utilização de mensagens RADIUS para a alteração de características de sessões ativas no NAS. Esta funcionalidade é alcançada via mensagem RADIUS “CoA - Change of Authorization”.

De acordo com a referência (RFC 5176, 2008), a mensagem CoA é enviada pelo servidor RADIUS em direção ao cliente RADIUS (na arquitetura da Figura 6 é o NAS) e tem a capacidade de alterar os dados de uma sessão ativa sem que seja necessário reiniciar essa sessão. A Figura 16 mostra a sinalização do CoA.

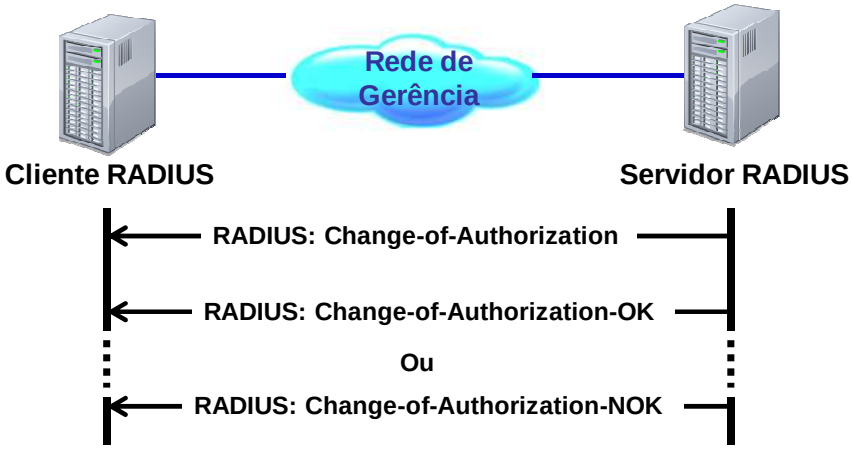


Figura 16 – CoA – “Change of Authorization”

Neste trabalho, o objetivo é alterar a largura de banda permitida para cada usuário com sessão ativa no NAS. Isso é alcançado por meio de atributos RADIUS que são inseridos dentro da mensagem CoA. Os atributos RADIUS que alteram a largura de banda das conexões estão descritos na Tabela 1. Esses atributos RADIUS estão definidos pela Wi-Fi Alliance em (ANTON et al., 2003).

Tabela 1 - Atributos RADIUS “WISPr”

Atributo	Descrição
WISPr-Bandwidth-Max-Up	Largura de banda máxima que o usuário terá no Upload. Ex. 500000 (500 Kbps)
WISPr-Bandwidth-Max-Down	Largura de banda máxima que o usuário terá no Download. Ex. 2000000 (2 Mbps)

Tendo em vista a arquitetura dos experimentos utilizada neste trabalho, o controle da vazão dos usuários é feito com os passos abaixo (ilustrados na Figura 17). O intervalo entre as leitura de RSSI foi colocado no valor mínimo que a bancada permite sem que haja perda de leitura: 3 segundos. Já o intervalo para o registro dos dados da sessão (bytes transferidos. vazão etc) foi fixado em 1

minuto (que é a periodicidade mínima que o NAS permite para o envio de mensagens de “accounting”).

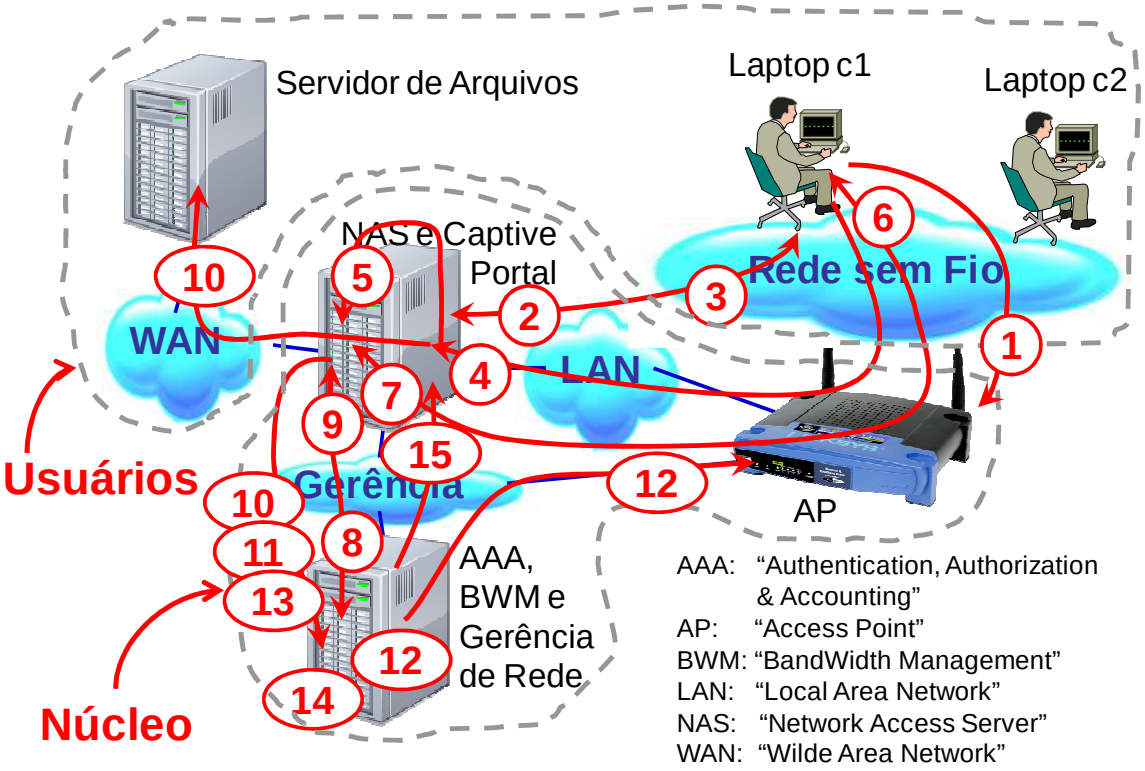


Figura 17 - Controle da vazão dos usuários

1) O Usuário inicia a conexão do laptop com o AP
2) Após a conexão com sucesso no AP, o laptop envia uma solicitação de endereço IP via protocolo DHCP.
3) O NAS responde com o IP para o usuário.
4) Usuário inicia navegação
5) NAS captura a sessão e redireciona para o “Captive Portal”
6) “Captive Portal” solicita o Nome do Usuário e a Senha
7) Usuário informa o Nome do Usuário e a Senha
8) “Captive Portal” solicita ao AAA autenticação/autorização da sessão.
9) AAA consulta a base de dados, obtém dados de autenticação e vazão inicial permitida e autentica a sessão: envia mensagem RADIUS Access-Accept incluindo a vazão inicial permitida para o usuário.

10) NAS libera a navegação do usuário, faz a limitação da vazão, de acordo com o indicado pelo AAA, e inicia o “accounting” via protocolo RADIUS.
11) Mensagem “accounting” start chega ao servidor AAA.
12) AAA armazena sessão internamente, registra informações de tráfego no arquivo CSV e inicia supervisão de RSSI a cada 3 segundos junto ao AP (para isso, utiliza o endereço MAC presente na mensagem RADIUS “accounting-Request).
13) NAS envia uma mensagem RADIUS de “accounting” a cada minuto.
14) AAA faz a média do RSSI e registra a média calculada juntamente com as informações de tráfego da mensagem RADIUS “accounting” no arquivo CSV.
15) A cada 5 mensagens de “accounting” (e consequentemente a cada 5 minutos), o AAA verifica se o usuário associado à sessão é o usuário que terá a largura de banda diminuída. Se for o caso, envia mensagem CoA para o NAS informando qual a nova vazão permitida para a sessão.
    a. A nova vazão permitida (caso a mensagem CoA seja enviada) é 10% da vazão corrente da sessão em questão. Ex.: se a vazão corrente é 4500 Kbps, a nova vazão indicada na mensagem CoA é 4050 Kbps.

3.6.1. Fluxogramas do Controle de Vazão

Os passos do controle da vazão de usuários foram implementados utilizando o PolicyFlow do 8950AAA. O PolicyFlow resultante está no “ANEXO A – PROCEDIMENTOS DE INSTALAÇÃO/CONFIGURAÇÃO”, item “8.3.1 - 8950AAA - PolicyFlow”. Os fluxogramas dessa implementação são mostrados na Figura 18, na Figura 19 e na Figura 20.

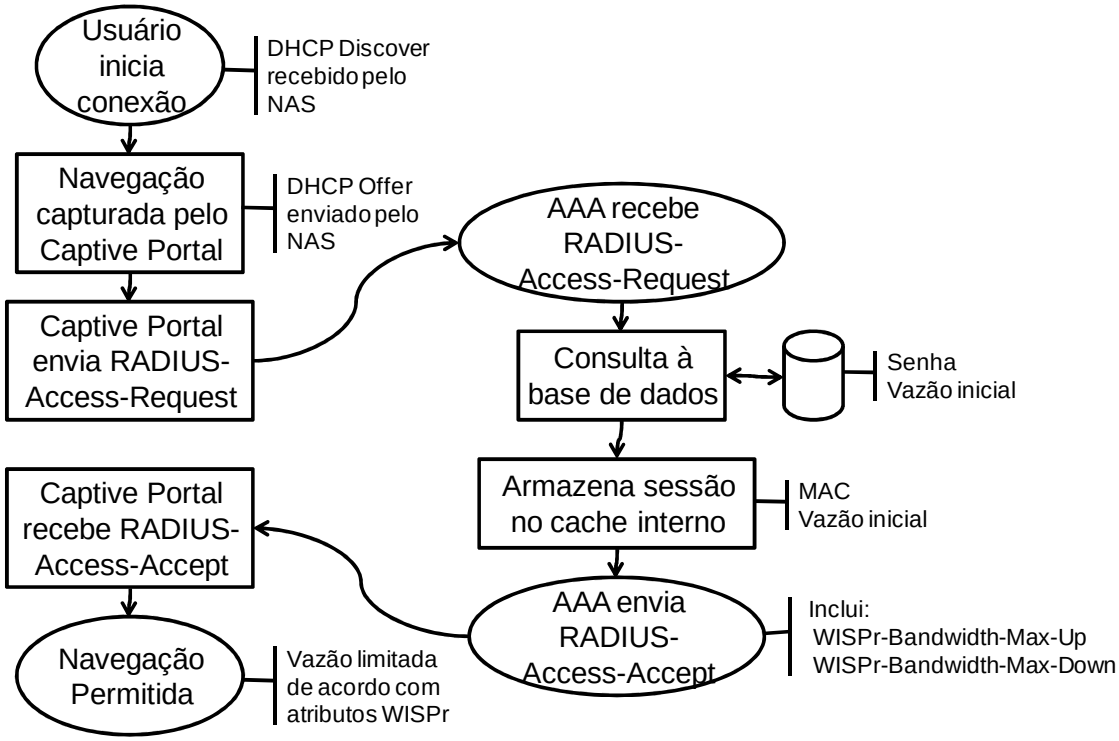


Figura 18 – Estabelecimento da sessão

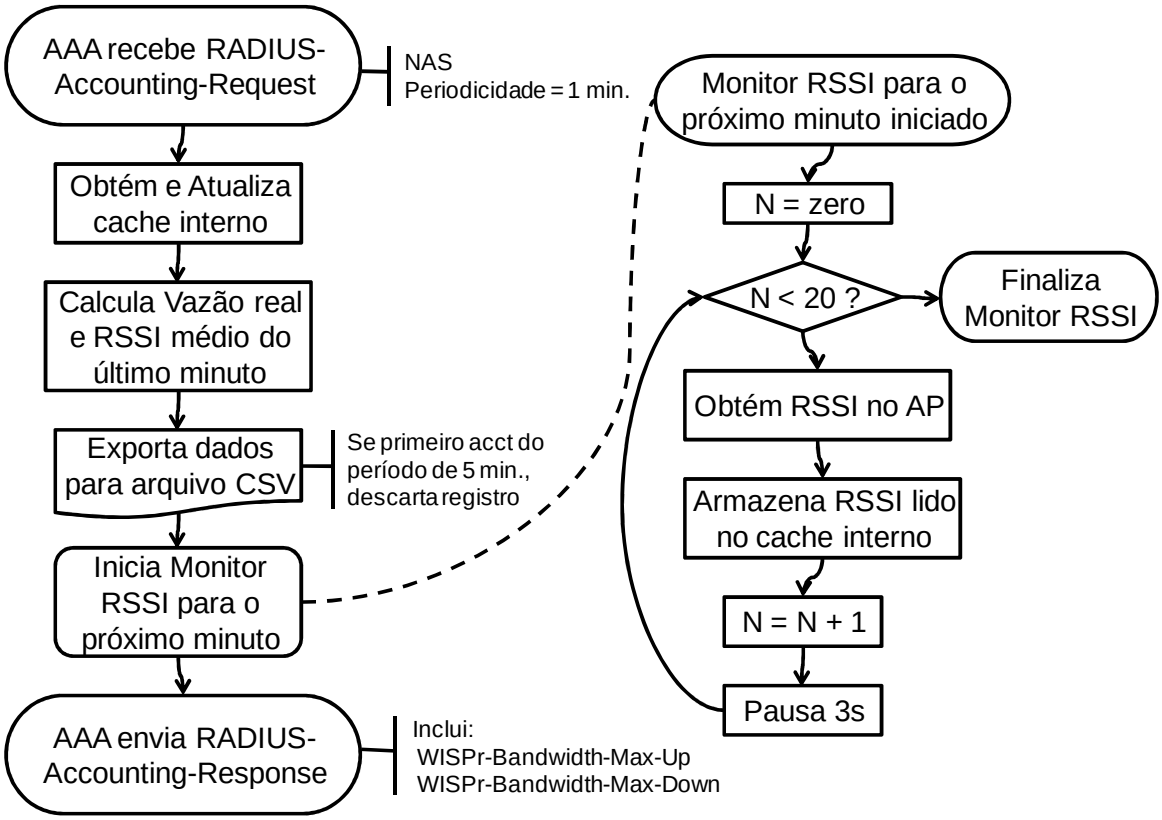


Figura 19 – Registro de Dados CSV e Monitor RSSI

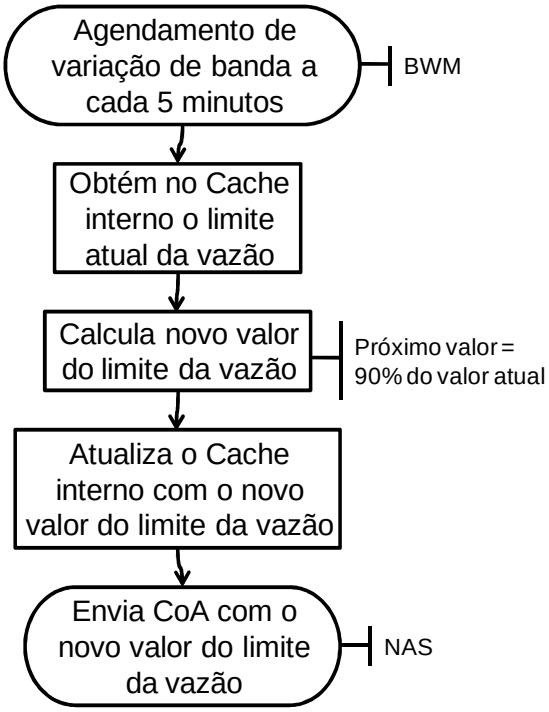


Figura 20 – Variação da Vazão Permitida

4. METODOLOGIA E AMBIENTE DE TESTES

Para realizar o controle de vazão proposto pelo item “3.6 - Método de Controle de Vazão”, foi construída uma bancada de testes que atende à arquitetura especificada no item “3.2 - Arquitetura dos Experimentos”.

A bancada de testes proposta em (PERIS et al., 2010) foi utilizada como referência, porém, os papeis de NAS e “Captive Portal” passaram a ser executados pelo CoovaChilli.

Esta bancada foi construída tendo em vista a utilização de equipamentos e soluções disponíveis comercialmente ou sem restrição de acesso. O resultado obtido foi a bancada mostrada no item 4.1.

4.1. Bancada de Testes

A bancada de testes foi construída tendo em vista a arquitetura estabelecida.

É importante observar que alguns componentes da arquitetura proposta no item “3.2 - Arquitetura dos Experimentos” foram reunidos em um único equipamento (computador 1), que concentrou as funções do núcleo NAS, “Captive Portal”, AAA, BWM e Gerência de Rede.

A bancada de testes foi construída utilizando os equipamentos e configurações mostrados na Figura 21. Os procedimentos de instalação destes componentes podem ser encontrados no item “8 - ANEXO A – PROCEDIMENTOS DE INSTALAÇÃO/CONFIGURAÇÃO”.

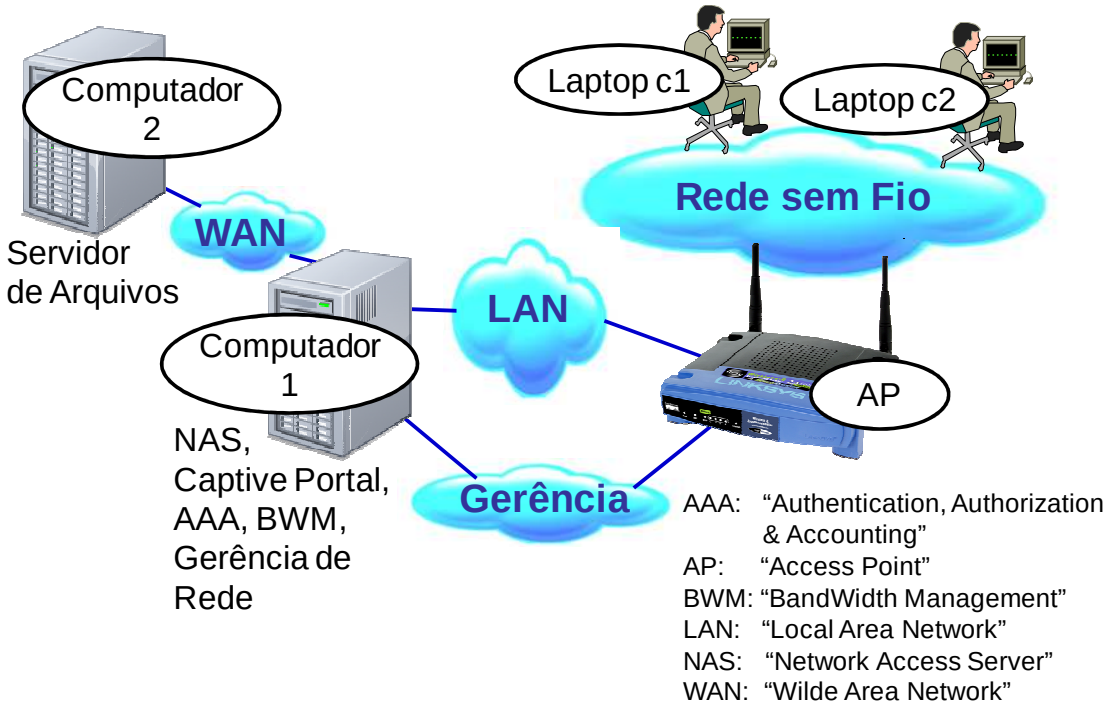


Figura 21 - Componentes da Bancada de Testes

4.1.1. Computador 1

Os Procedimentos de instalação do Sistema Operacional e dos componentes software do computador 1 estão descritos nos ítens:

- 8.1-NAS
- 8.3-Gestor de Autenticação e Largura de Banda
  - 8.3.1-8950AAA - PolicyFlow
  - 8.3.2-Scripts utilizados

A configuração do computador 1 é:

- Computador Virtual suportado pelo software VMWare Player versão 3.1.5 (VMware, 2011) instalado num computador AMD Athlon XP, 1,5 GB RAM, HD 160GB (Sistema operacional Windows XP SP3).
  - Linux Ubuntu-Server 11.11 (Ubuntu, 2011);
  - 8950AAA (8950AAA, 2011);
  - CoovaChilli (CoovaChilli, 2011);
  - Grase (Grase, 2011).

Com isto, este computador desempenha as seguintes funções da arquitetura dos experimentos:

- NAS
- “Captive Portal”
- AAA
- BWM
- Gerência de Rede

4.1.2. Computador 2

A configuração do computador 2 é:

- AMD Athlon XP, 1.0 GB RAM, HD 160GB.
    - Windows XP SP3.
    - OpenSSH (Incluindo SFTP e SCP)

Este computador desempenha as seguintes funções da arquitetura dos experimentos:

- Servidor de arquivos.
    - Os arquivos utilizados foram de tamanhos diversos, variando de 50 Kbytes a 2 Gbytes.

4.1.3. Laptop c1

A configuração do Laptop c1 é:

- Lenovo ThinkPad T60
    - Windows XP
    - Software WinSCP (WinSCP, 2012)
    - Cópia de arquivos em rede suportada pelo WinSCP.

4.1.4. Laptop c2

A configuração do Laptop c2 é:

- Asus EeeePc AMD Vision

- Windows 7 SE
- Software WinSCP (WinSCP, 2012)
- Cópia de arquivos em rede suportada pelo WinSCP.

4.1.5. AP

O Procedimento de instalação do AP está descrito no item 8.2-AP.

A configuração do AP é:

- Linksys wrt54G com o firmware DD-WRT (DD-WRT, 2011).
- Modo bridge: Todas as funções típicas de roteador desempenhadas por esse tipo de equipamento foram eliminadas.
- Conexão via Telnet habilitada: Este tipo de conexão foi utilizado para a coleta de informações de RSSI das conexões ativas.
  - O Anexo A (item “8.3.2 - Scripts utilizados”) mostra o script “ObtemRSSI.sh” utilizado para essa coleta.

4.1.6. Conexões físicas entre os componentes

As conexões físicas entre os componentes da bancada de testes são:

- Laptops c1 e c2 <--> AP: IEEE 802.11 (Wi-Fi)
- Computador 1 <--> AP: cabo FastEthernet.
- Computador 2 <--> AP: cabo FastEthernet.
- Computador 1 <--> Computador 2: Interface virtual VMWare.

Estas conexões podem ser vistas na Figura 22.

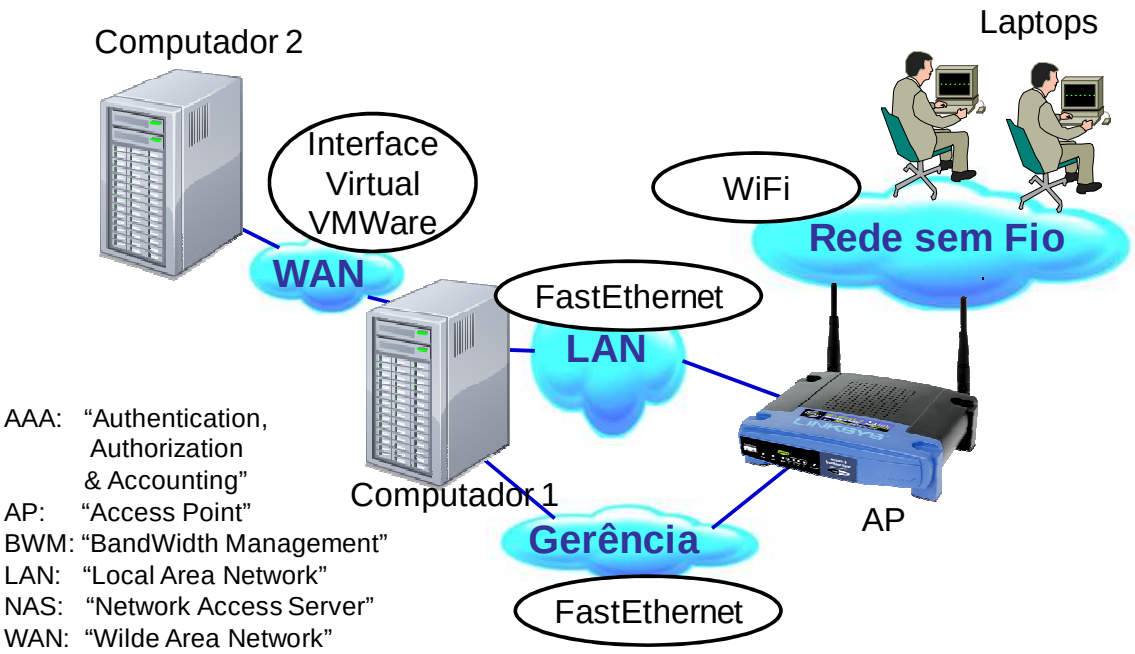


Figura 22 - Conexões físicas entre os componentes

4.2. Procedimento de Validação

O procedimento de validação teve como objetivo verificar se a bancada de testes descrita em "4.1 - Bancada de Testes" possuía os componentes funcionando, se as conexões estavam operacionais e se a coleta de dados ocorria de forma consistente.

Este procedimento consistiu em executar o controle de vazão do Laptop c1 (de acordo com os passos do item "3.6 - Método de Controle de Vazão"), utilizando o seguinte ambiente:

- O BWM faz o controle da largura de banda disponível para o Laptop c1.
- Somente o Laptop c1 gera tráfego
- O Laptop c1 gera tráfego suficiente para ocupar toda a vazão permitida pelo NAS.
- As informações de tráfego do Laptop c1 são coletadas.

O resultado esperado para o procedimento de validação era um arquivo CSV com as informações do tráfego transmitido e recebido pelo Laptop c1. A partir desse arquivo CSV foi possível gerar diversos gráficos como, por exemplo, vazão x RSSI, apresentado no capítulo “5 - Resultados Obtidos”.

4.3. Experimentos

Os experimentos desse trabalho consistiram em utilizar a bancada de teste descrita em “4.1 - Bancada de Testes” com dois laptops em condições diferentes de propagação do sinal Wi-Fi. Para isso, foi feito o controle de vazão dos Laptops c1 e c2 (de acordo com os passos do item “3.6 - Método de Controle de Vazão”), utilizando o seguinte ambiente:

- O Laptop c1 (em boas condições de propagação do sinal Wi-Fi) gera o tráfego máximo, ou seja, tenta transmitir ou receber dados com a maior vazão possível.
- O Laptop c2 (o ofensor em más condições de propagação do sinal Wi-Fi) tenta transmitir ou receber dados com a maior vazão possível..
- BWM atua no sentido de restringir o tráfego do ofensor (Laptop c2).
- As informações de tráfego são coletadas.

Dois experimentos foram realizados:

1) Experimento com controle da vazão de Download do ofensor
2) Experimento com controle de vazão de Upload do ofensor

4.3.1. Experimento com controle da vazão de Download do ofensor

Neste experimento, os Laptops c1 e c2 tentavam receber dados com a maior vazão possível.

Este experimento foi realizado utilizando o ambiente de testes definido em “4.3 - Experimentos”, complementado com as seguintes características:

- Laptop c1 em boas condições de propagação do sinal Wi-Fi.
- Laptop c2 em más condições de propagação do sinal Wi-Fi – ofensor.
- Tráfego liberado em ambos os laptops (sem controle de Largura de Banda) no início do experimento.
- O BWM controla o tráfego do ofensor no sentido de alterar o tráfego permitido para o ofensor de forma cada vez mais restritiva.

Durante o experimento, dados de tráfego e de RSSI de ambos os laptops foram coletados. O RSSI coletado permitiu comprovar as condições de sinal Wi-Fi dos laptops previstas no ambiente do experimento.

Em todo o experimento, os laptops demandaram tráfego para ocupar toda a capacidade de download oferecida pelo sistema para cada um deles.

Para possibilitar a investigação das condições entre as diferentes interfaces aéreas do padrão IEEE802.11, este experimento foi realizado duas vezes:

- AP configurado para utilizar o IEEE 802.11b (conhecido como Wi-Fi 11 Mbps)
- AP configurado para utilizar o IEEE 802.11g (conhecido como Wi-Fi 54 Mbps).

4.3.2. Experimento com controle de vazão de Upload do ofensor

Neste experimento, os Laptops c1 e c2 tentavam transmitir dados com a maior vazão possível.

Este experimento foi realizado utilizando o ambiente de testes definido em “4.3 - Experimentos”, complementado com as seguintes características:

- Laptop c1 em boas condições de propagação do sinal Wi-Fi.
- Laptop c2 em más condições de propagação do sinal Wi-Fi – ofensor.
- Tráfego liberado em ambos os laptops (sem controle de Largura de Banda) no início do experimento.

- O BWM controla o tráfego do ofensor no sentido de alterar o trafego permitido para o ofensor de forma cada vez mais restritiva.

Durante o experimento, dados de tráfego e de RSSI de ambos os laptops foram coletados. O RSSI coletado permitiu comprovar as condições de sinal Wi-Fi dos laptops previstas no ambiente do experimento.

Em todo experimento, os laptops demandaram tráfego para ocupar toda a capacidade de upload oferecida pelo sistema para cada um deles.

Este experimento foi realizado com o AP configurado para utilizar o IEEE 802.11g.

5. RESULTADOS OBTIDOS

Em todos os procedimentos, os 2 laptops conectaram-se ao AP via sinal Wi-Fi e conectaram-se ao CoovaChili via Web-Browser. O Laptop c1 sempre utilizou o usuário “teste” e o Laptop c2 sempre utilizou o usuário “teste2”. O procedimentos de aprovisionamento dos usuários na base de dados do NAS é mostrado no item “8.3.3-8950AAA - Aprovisionamento do usuário “teste””.

Além disso, o Laptop c1 sempre ficou próximo ao AP, enquanto o Laptop c2 sempre ficou distante e em locais desfavoráveis para a propagação de sinal RF. A cada 5 minutos, o AAA reduziu a vazão permitida para o Laptop c2 em 10% de seu valor.

A Figura 23 apresenta a tela de login do Grase/CoovaChilli e a Figura 24 mostra a confirmação da conexão do usuário.

Figura 23 - Tela de login do Grase/CoovaChilli

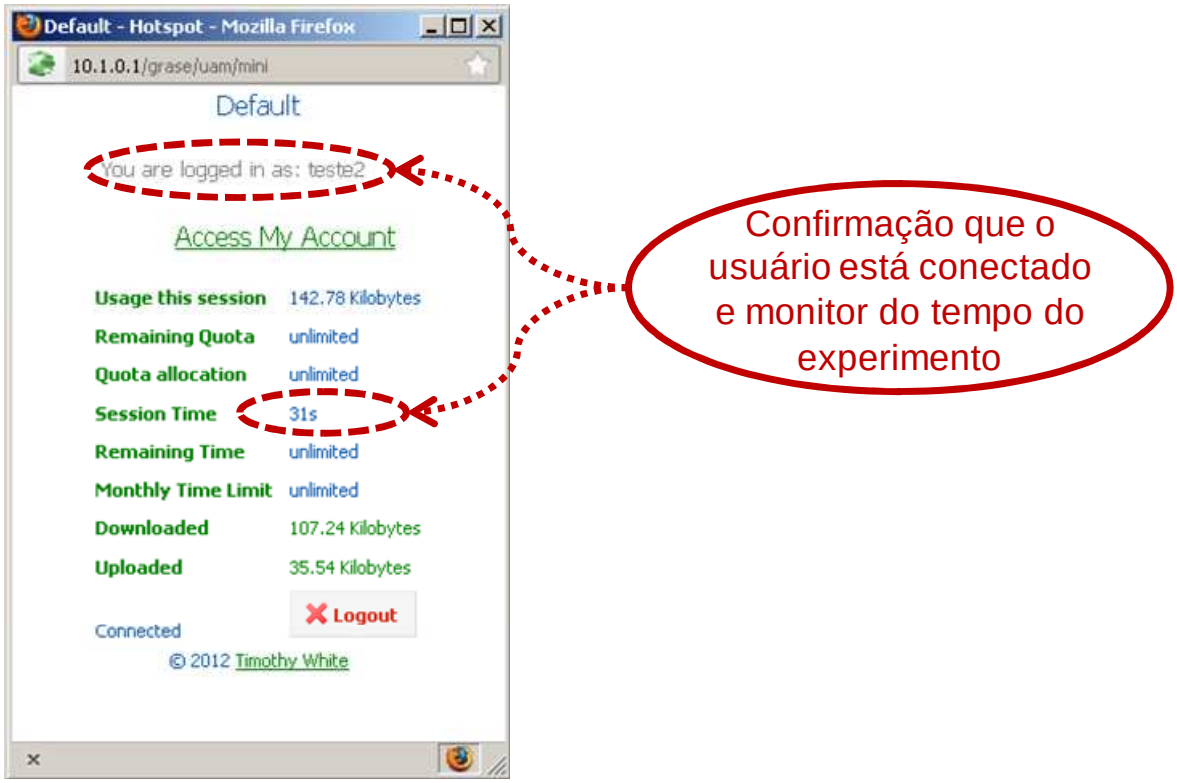


Figura 24 – Confirmação de conexão no Grase/CoovaChilli

5.1. Procedimento de Validação

Os resultados do procedimento de validação “4.2 - Procedimento de Validação” foram coletados e utilizados para gerar um gráfico, comparando o tráfego de download com o RSSI de uplink. Esse gráfico gerado pode ser visto na Figura 25.

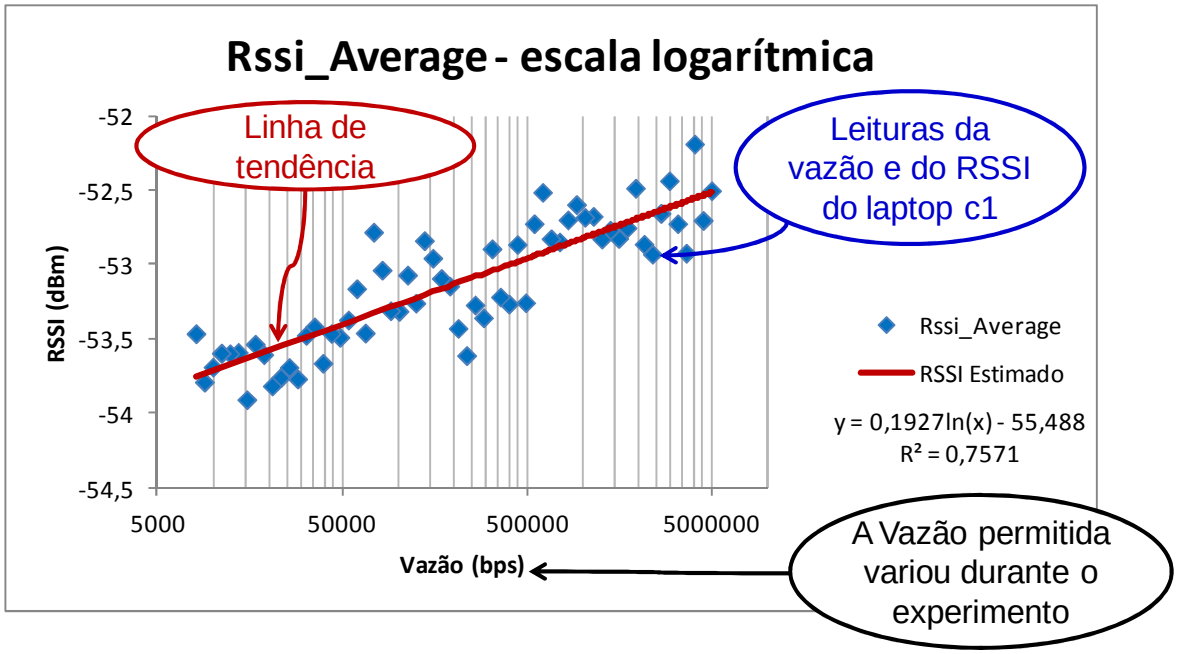


Figura 25 - RSSI x Vazão no procedimento de validação

Esse resultado confirmou que a banca estabelece as conexões dos usuários com a internete, coleta e consolida os dados (do NAS e do AP) corretamente. Além disso, o resultado confirmou que a limitação da vazão dos usuários ocorre como esperado.

Com isso foi possível inciar os experimentos previstos neste trabalho.

5.2. Experimentos

5.2.1. Experimento com controle da vazão de Download do ofensor

Os resultados do experimento descrito em “4.3.1 - Experimento com controle da vazão de Download do ofensor” foram coletados e utilizados para gerar gráficos que mostram as vazões (download) dos Laptops c1 e c2 ao longo do tempo, além da soma dessas vazões (vazão total). Nestes gráficos, a tendência do comportamento dos três conjuntos de pontos foi ajustada utilizando uma curva polinomial de ordem 6, conforme descrito no ANEXO B - Determinação da tendência dos pontos coletados.

O gráfico gerado na primeira parte do experimento (em que o AP foi configurado para utilizar somente 802.11g) pode ser visto na Figura 26. O gráfico gerado na segunda parte do experimento (em que o AP foi configurado para utilizar somente 802.11b) pode ser visto na Figura 27. Em ambos os gráficos, apenas o laptop c1 encontra-se em operação desde o início das medições. O laptop c2 inicia o tráfego somente 5 minutos após o início das medições.

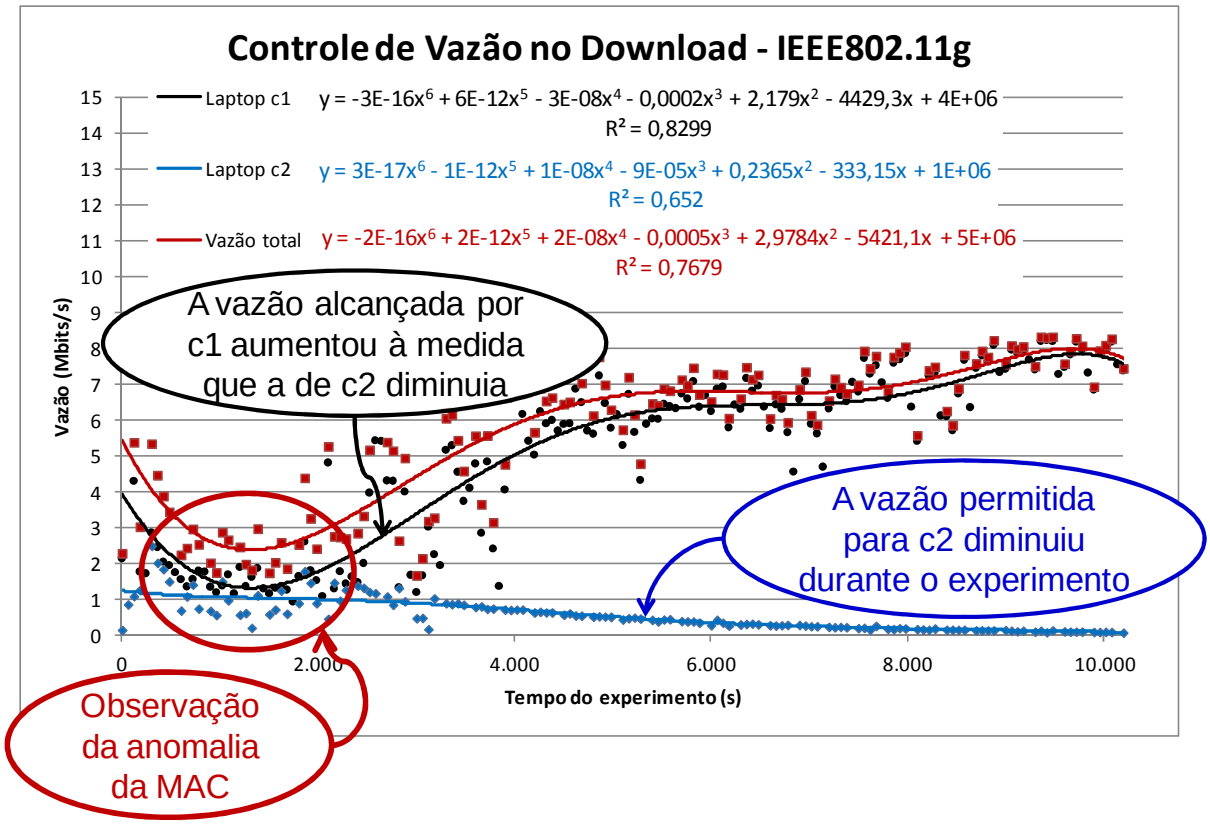


Figura 26 - Experimento com controle da vazão de Download do ofensor com AP em 802.11g

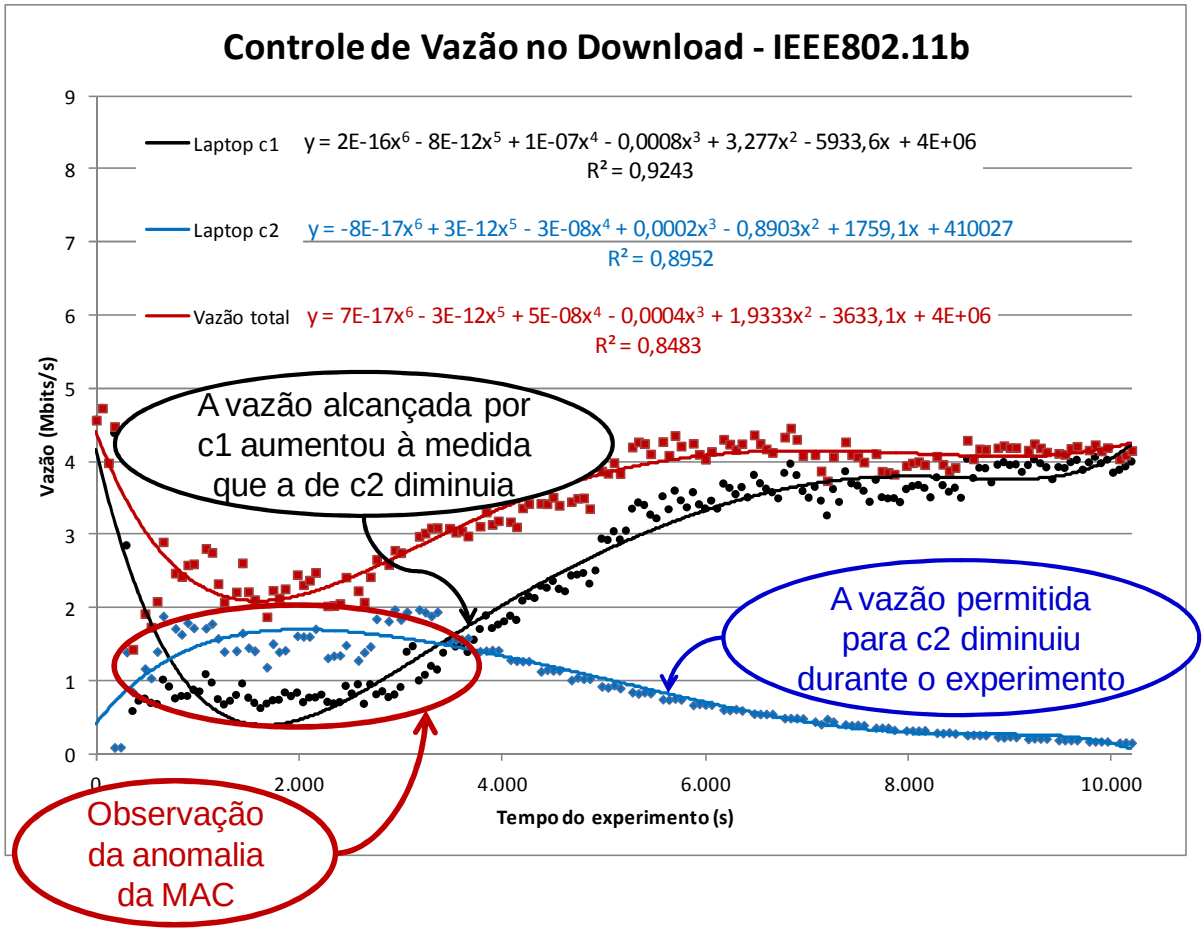


Figura 27 - Experimento com controle da vazão de Download do ofensor com AP em 802.11b

Durante estes experimentos, os valores do RSSI do uplink foram coletados e registrados conforme descrito em “3.6 - Método de Controle de Vazão”. Esses valores podem ser observados na Figura 28 e na Figura 29.

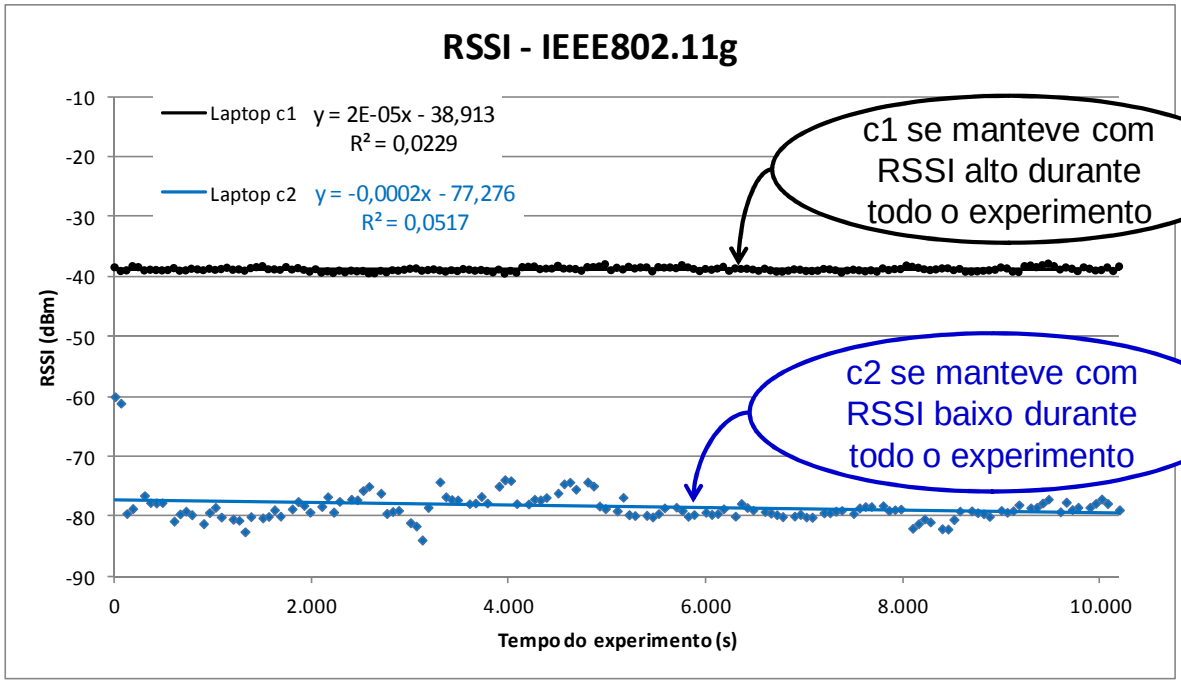


Figura 28 - RSSI durante o experimento 4.3.1 (802.11g).

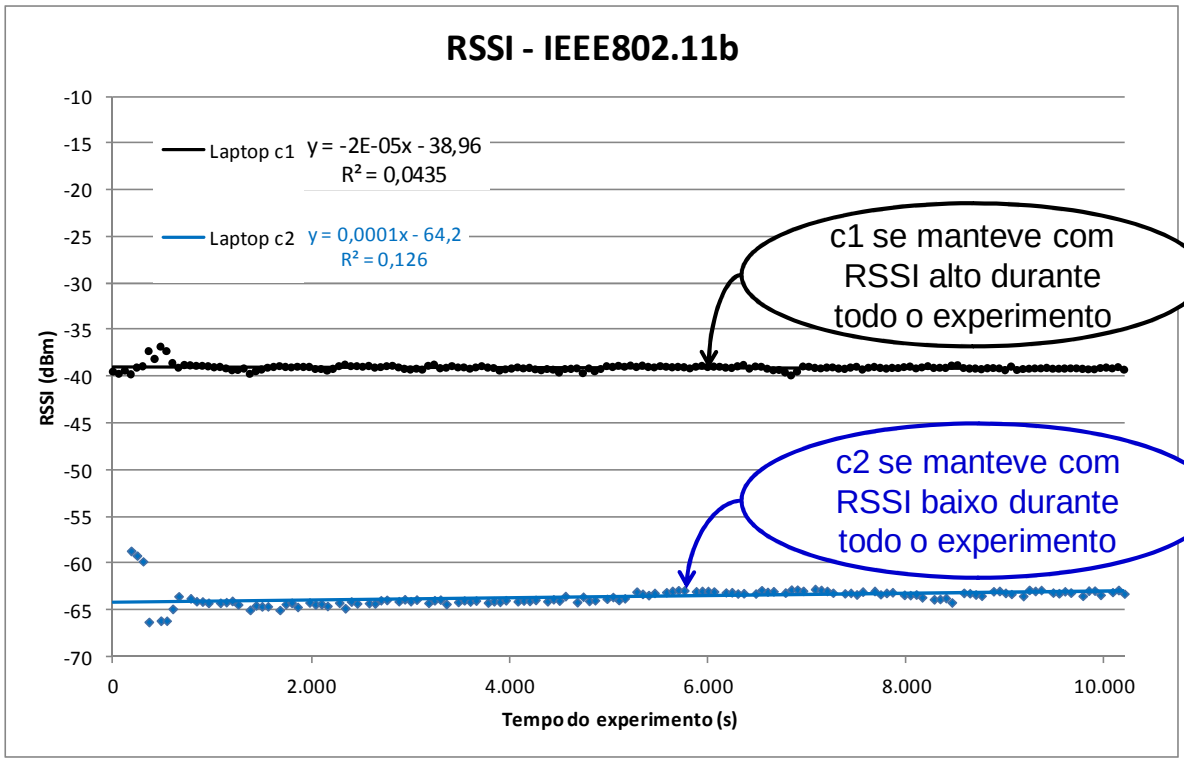


Figura 29 - RSSI durante o experimento 4.3.1 (802.11b).

5.2.2. Experimento com controle de vazão de Upload do ofensor

Os resultados do experimento descrito em “4.3.2 - Experimento com controle de vazão de Upload do ofensor” foram coletados e utilizados para gerar um gráfico que mostra a vazão (upload) dos laptops c1 e c2 ao longo do tempo, além da soma dessas vazões (vazão total). Também neste gráfico, a tendência do comportamento dos três conjuntos de pontos foi ajustada utilizando uma curva polinomial de ordem 6, conforme descrito no ANEXO B - Determinação da tendência dos pontos coletados.

O gráfico gerado nesse experimento pode ser visto na Figura 30.

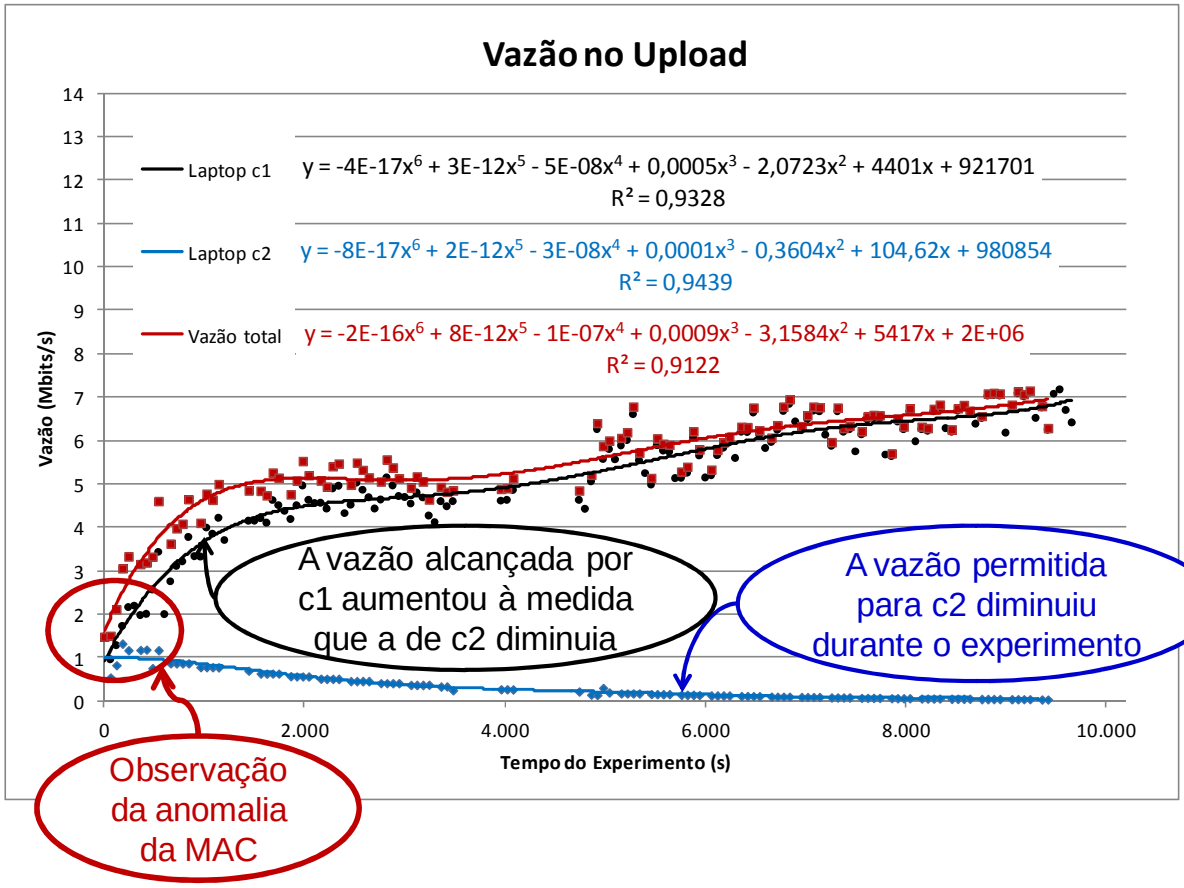


Figura 30 - Experimento com controle de vazão de Upload do ofensor

Durante este experimento, os valores do RSSI do uplink foram coletados e registrados conforme descrito em “3.6 - Método de Controle de Vazão”. Esses valores podem ser observados na Figura 31.

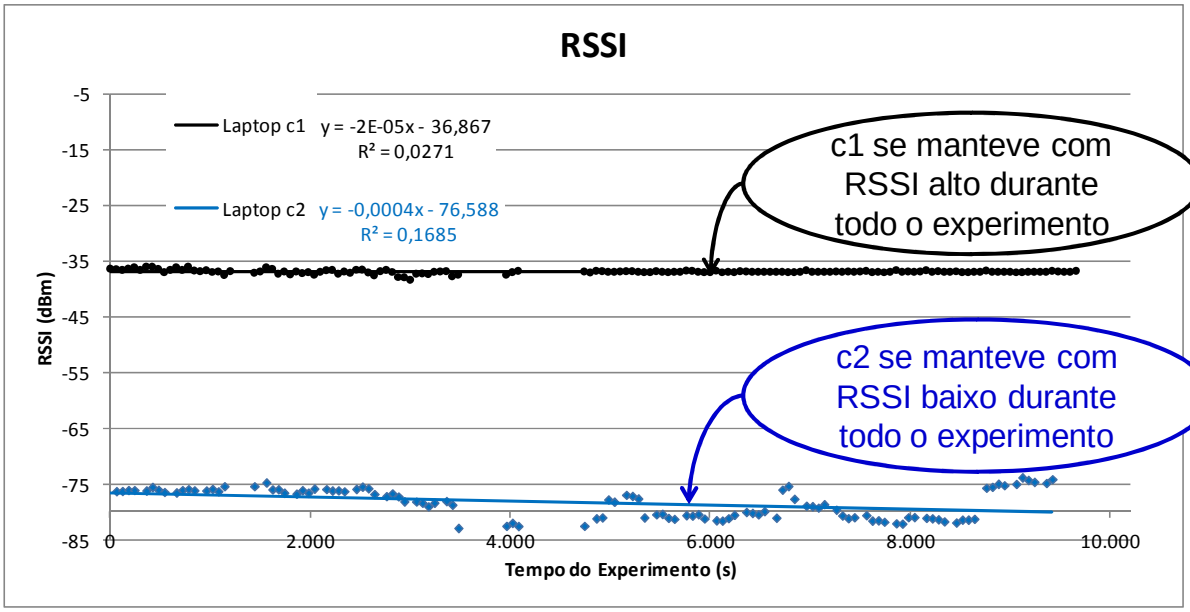


Figura 31 - RSSI durante o experimento 4.3.2.

5.2.3. Análise dos resultados

A partir dos resultados obtidos, foi possível observar a vazão do Laptop c1 sem a presença do ofensor, que é a vazão máxima desse Laptop. Também foi observado o valor da vazão máxima do Laptop c2. Esses valores estão registrados na Tabela 2.

Tabela 2 - Vazão máxima dos Laptops c1 e c2 nos experimentos

Experimento	Vazão Máxima c1 (bits/s)	Vazão Máxima c2 (bits/s)
Experimento com controle da vazão de Download do ofensor (802.11g)	8.219.592	1.777,664
Experimento com controle da vazão de Download do ofensor (802.11b)	4.731.400	1.944,928
Experimento com controle de vazão de Upload do ofensor	7.176.512	1.325,936

Os gráficos dos experimentos mostram que a vazão do Laptop c2 não começou a diminuir imediatamente após o controle de vazão do NAS começar a atuar. Isto se deve ao fato de que a vazão desse Laptop já era baixa devido à sua condição de propagação. Por exemplo, no gráfico do experimento “Experimento com controle da vazão de Download do ofensor-802.11b” (Figura 27), percebe-se que a vazão de c2 começou a diminuir por volta de 3.300 segundos após o início

do experimento. Esse é o momento em que o limite imposto pelo NAS ficou abaixo da vazão que o Laptop c2 alcançaria se não houvesse esse limite.

Com o objetivo de visualizar melhor a evolução dos dados a partir do momento em que o NAS começou a diminuir a vazão no Laptop c2 (ou seja, a partir do momento em que a anomalia da MAC começou a ser mitigada), os gráficos dos experimentos foram refeitos a partir desse ponto. Os gráficos resultantes podem ser vistos na Figura 32, na Figura 33 e na Figura 34.

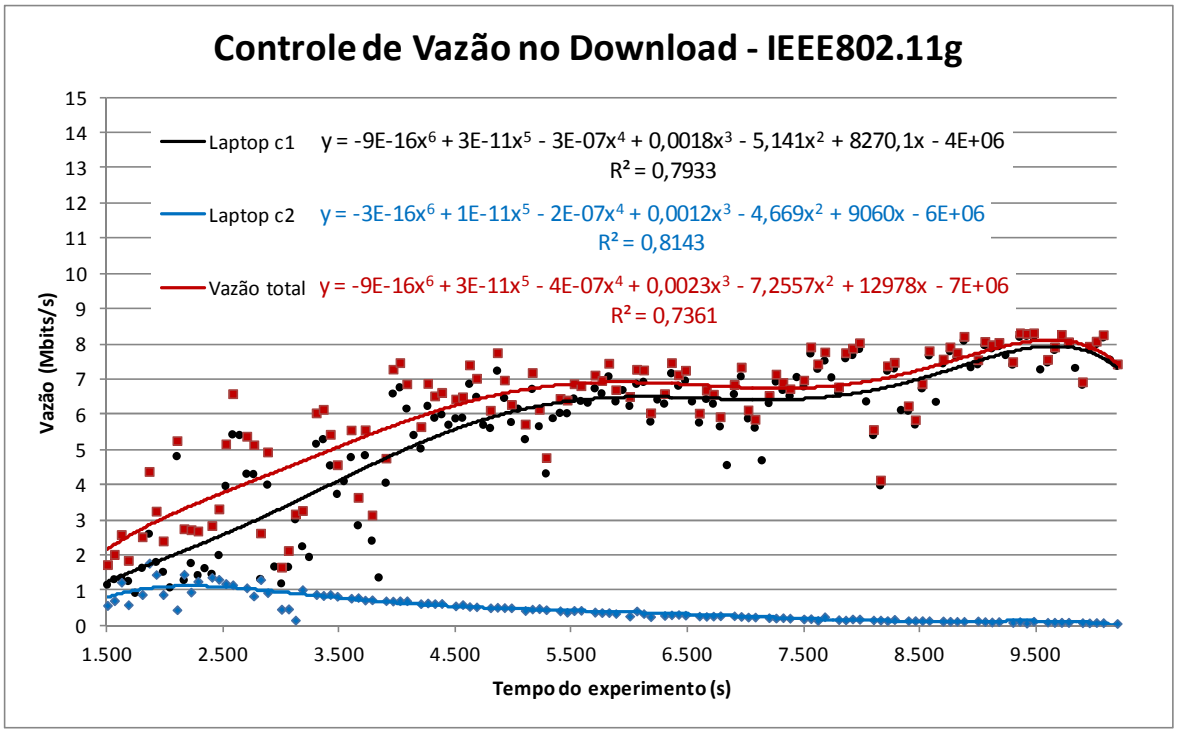


Figura 32 – Evolução dos dados a partir do controle da anomalia da MAC (download, IEEE 802.11g)

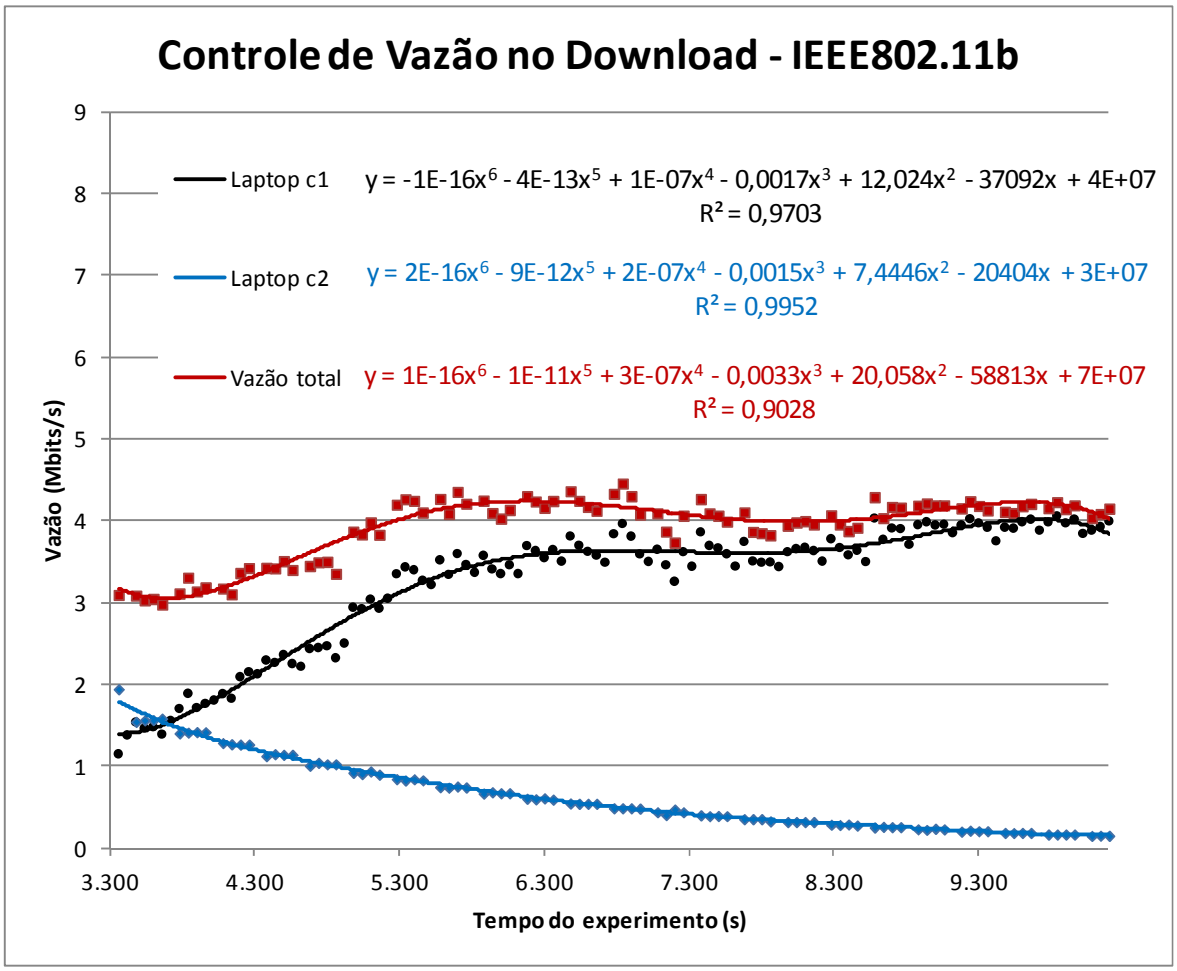


Figura 33 - Evolução dos dados a partir do controle da anomalia da MAC (download, IEEE 802.11b)

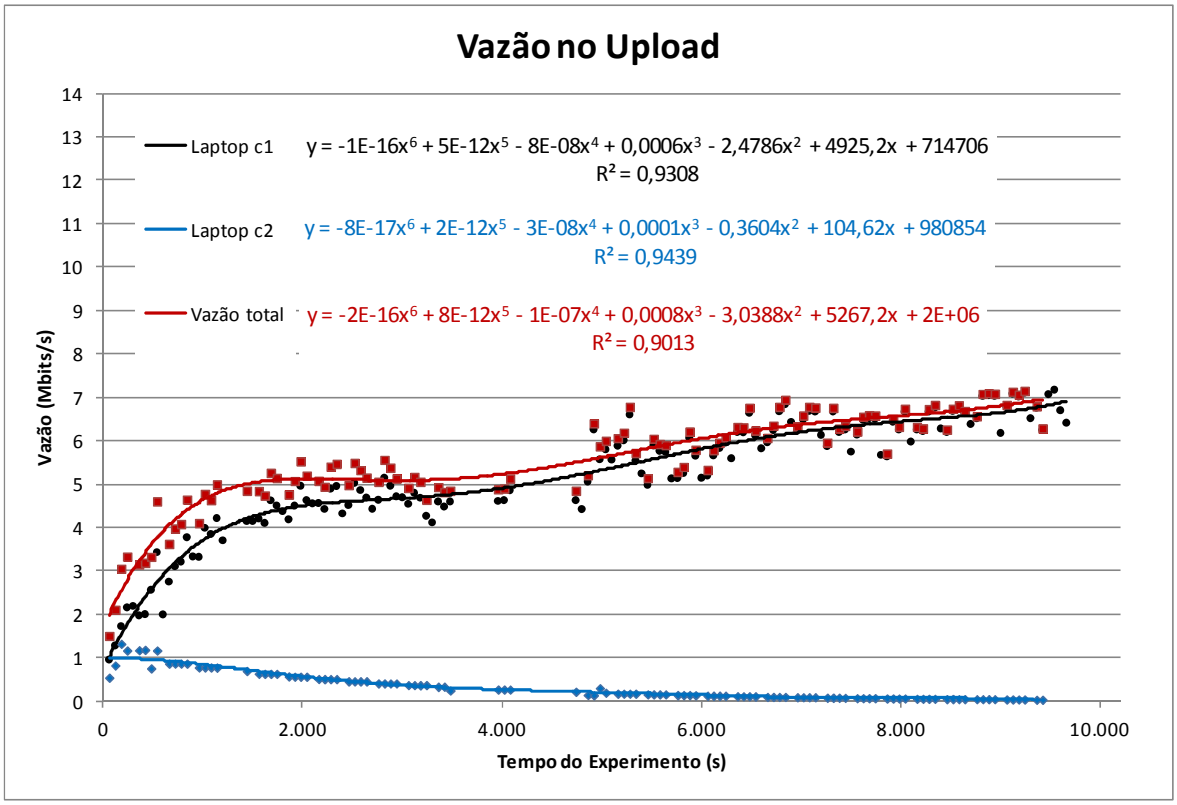


Figura 34 - Evolução dos dados a partir do controle da anomalia da MAC (upload)

A partir dos gráficos recortados, foi possível observar a vazão dos Laptops c1 e c2 no ponto inicial do controle de vazão. Para o Laptop c1 essa é a vazão mínima. Esses valores estão registrados na Tabela 3.

Tabela 3 - Vazão inicial dos Laptops c1 e c2 nos experimentos

Experimento	**Vazão Inicial (bits/s)**	
	Laptop c1	**Laptop c2**
Experimento com controle da vazão de Download do ofensor (802.11g)	1.158.608	578.288
Experimento com controle da vazão de Download do ofensor (802.11b)	1.153.000	1.944.928
Experimento com controle de vazão de Upload do ofensor	961.104	548.504

A taxa de crescimento da vazão total corresponde à inclinação das curvas correspondentes nos gráficos recortados. Observou que, nesses gráficos recortados, essa taxa de crescimento da vazão total é grande até certo ponto. Neste trabalho, esse ponto foi chamado de “ponto de diminuição da taxa de

crescimento da vazão total". O instante quem que esse ponto ocorre foi chamado de "momento de diminuição da taxa de crescimento da vazão total".

Com os "momentos de diminuição da taxa de crescimento da vazão total" estimados para os três experimentos, cada uma das curvas (vazões dos Laptops c1 e c2 e vazão total) foi aproximada para dois segmentos de reta em torno desses momentos. Os gráficos resultantes dessa aproximação estão na Figura 35, na Figura 36 e na Figura 37.

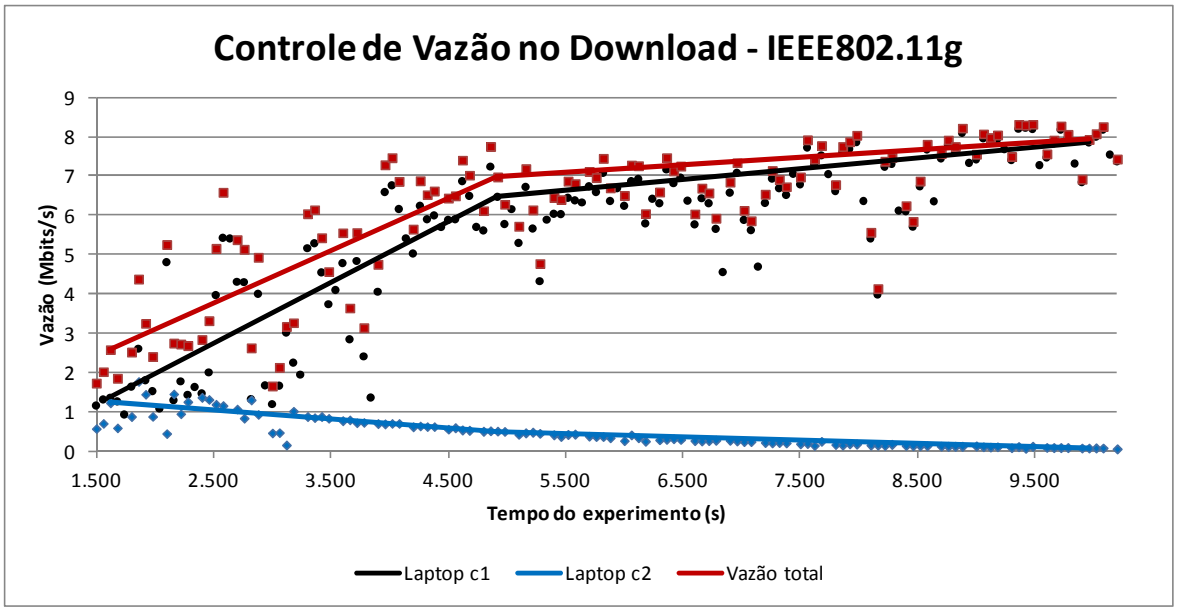


Figura 35 – Aproximação para dois segmentos de reta em em torno dos momentos de diminuição da taxa de crescimento da vazão total (download, IEEE 802.11g).

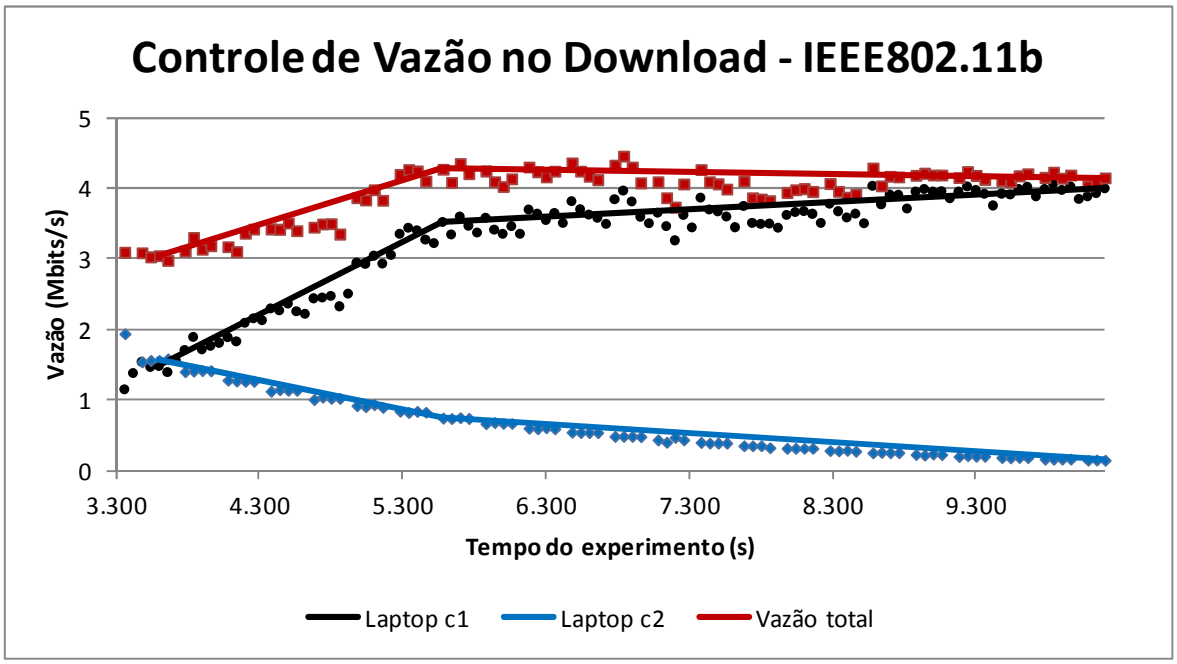


Figura 36 – Aproximação para dois segmentos de reta em em torno dos momentos de diminuição da taxa de crescimento da vazão total (download, IEEE 802.11b).

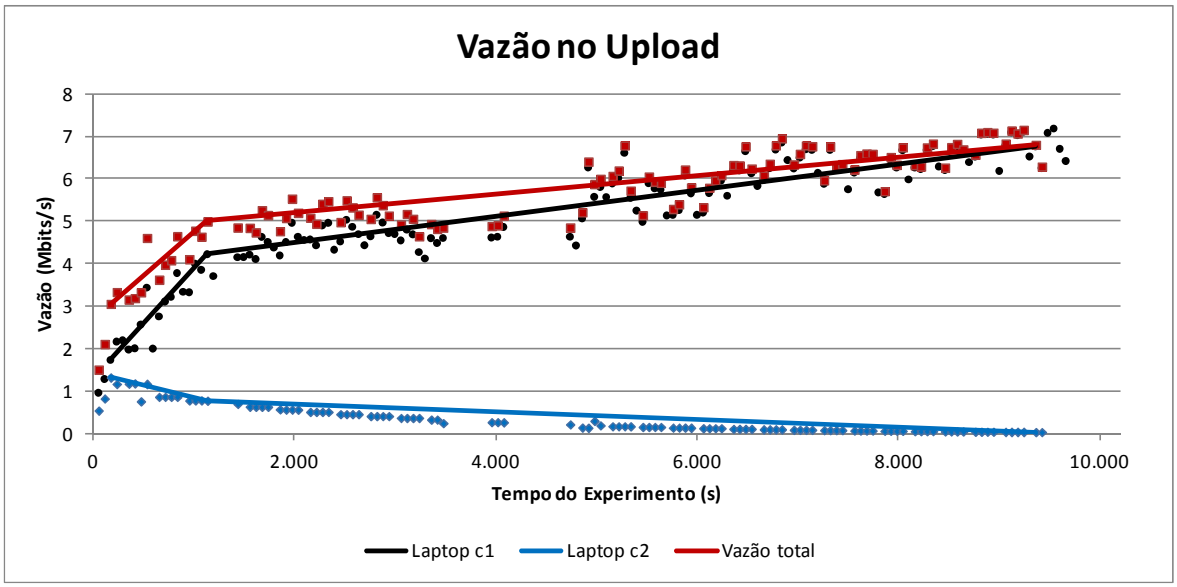


Figura 37 - Aproximação para dois segmentos de reta em em torno dos momentos de diminuição da taxa de crescimento da vazão total (upload)

Em todos os casos, o momento de diminuição da taxa de crescimento da vazão total ficou localizado quando a vazão do Laptop c2 era entre 40% e 50% da vazão inicial desse laptop.

A partir da vazão aproximada para dois segmentos de reta, foi construída a Tabela 4, na qual os dados de tráfego no início do experimento, no momento de diminuição da taxa de crescimento da vazão total e ao final do experimento são comparados.

Tabela 4 - Comparação da vazão no início e final do experimento e no momento de diminuição da taxa de crescimento da vazão total

Experimento	**Origem do dado**	**Início**	**Vazão no momento de diminuição da taxa de crescimento da vazão total**	**Final**
Experimento com controle da vazão de Download do ofensor (802.11g)	Tempo (s)	1.620	4.920	9.960
	Vazão Laptop c1 (bits/s)	1.351.976	6.462.440	7.849.504
	Vazão Laptop c2 (bits/s)	1.235.096	513.152	82.952
	Vazão total (bits/s)	2.587.072	6.975.592	7.932.456
Experimento com controle da vazão de Download do ofensor (802.11b)	Tempo (s)	3.600	5.580	10.200
	Vazão Laptop c1 (bits/s)	1.483.232	3.521.880	3.997.656
	Vazão Laptop c2 (bits/s)	1.566.320	750.416	154.392
	Vazão total (bits/s)	3.049.552	4.272.296	4.152.048
Experimento com controle de vazão de Upload do ofensor	Tempo (s)	2.762	3.722	11.942
	Vazão Laptop c1 (bits/s)	1.731.352	4.218.424	6.758.560
	Vazão Laptop c2 (bits/s)	1.325.936	779.528	41.008
	Vazão total (bits/s)	3.057.288	4.997.952	6.799.568

A partir da Tabela 4 foram calculadas as inclinações em cada segmento de reta (antes e depois do momento de diminuição da taxa de crescimento da vazão total) das retas da Figura 35, Figura 36 e Figura 37 a partir da Equação (1).

$$I_a = \frac{v_i - v_p}{t_i - t_p};\ I_d = \frac{v_p - v_f}{t_p - t_f} \tag{1}$$

Nessa equação:

- $I_a = Inclinação\ antes\ do\ ponto\ de\ inflexão$
- $I_d = Inclinação\ depois\ do\ ponto\ de\ inflexão$
- $v_i = Vazão\ no\ início\ da\ reta$
- $v_p = Vazão\ no\ ponto\ de\ inflexão;$
- $v_f = Vazão\ no\ final\ da\ reta;$

- $t_i = Tempo\ de\ sessão\ no\ início\ da\ reta$
- $t_p = Tempo\ de\ sessão\ no\ ponto\ de\ inflexão$
- $t_f = Tempo\ de\ sessão\ no\ final\ da\ reta$

Os valores calculados das inclinações antes e depois do momento de diminuição da taxa de crescimento da vazão total estão na Tabela 5.

Tabela 5 – Inclinação antes e depois do momento de diminuição da taxa de crescimento da vazão total.

Experimento	Origem do dado	Inclinação (bits/s^2)	
		Antes	Depois
Experimento com controle da vazão de Download do ofensor (802.11g)	Laptop c1	1548,63	275,21
	Laptop c2	-218,77	-85,36
	Vazão total	1329,85	189,85
Experimento com controle da vazão de Download do ofensor (802.11b)	Laptop c1	1029,62	102,98
	Laptop c2	-412,07	-129,01
	Vazão total	617,55	-26,03
Experimento com controle de vazão de Upload do ofensor	Laptop c1	2590,70	309,02
	Laptop c2	-569,18	-89,84
	Vazão total	2021,53	219,17

Este estudo, em que os dados de vazão foram aproximados para segmentos de reta e as inclinações desses segmentos foram calculadas, poderia ser realizado calculando as equações derivadas $\left(\frac{\partial(vazão)}{\partial(tempo\ do\ experimento)}\right)$ das equações das respectivas curvas de tendência. Esse estudo pode ser observado no item “10 - ANEXO C – Estudo da inclinação das curvas tendência de vazão através das derivadas dessas curvas”, porém, para este trabalho, foram consideradas as inclinações dos segmentos de reta. A razão para isso é permitir que as informações sejam obtidas por meio de cálculos simples, viabilizando a implementação computacional desses cálculos.

Para comparar as taxas de variação das vazões dos Laptops c1 e c2, considerou-se o valor absoluto das razões entre as inclinações dos respectivos segmentos de reta na Tabela 5. A esse valor deu-se o nome de sensibilidade da vazão do Laptop c1. De maneira análoga, foi definida a sensibilidade da vazão total. Os valores calculados dessas sensibilidades estão na Tabela 6.

Tabela 6 – Sensibilidade da vazão (Laptop c1 e Vazão Total) em relação à vazão do Laptop c2 antes e depois do momento de diminuição da taxa de crescimento da vazão total.

Experimento	Origem do dado	Sensibilidade	
		Antes	Depois
Experimento com controle da vazão de Download do ofensor (802.11g)	Laptop c1	7,08	3,22
	Vazão total	6,08	2,22
Experimento com controle da vazão de Download do ofensor (802.11b)	Laptop c1	2,50	0,80
	Vazão total	1,50	0,20
Experimento com controle de vazão de Upload do ofensor	Laptop c1	4,55	3,44
	Vazão total	3,55	2,44

Pela natureza da definição das sensibilidades das vazões do Laptop c1 e total, valores maiores de sensibilidade indicam que uma pequena diminuição na vazão do laptop c2 provoca um grande aumento na vazão do laptop c1 ou vazão total (dependendo de qual sensibilidade está sendo considerada).

Observando a Tabela 6, é possível perceber que, em todos os casos, a sensibilidade antes do momento de diminuição da taxa de crescimento da vazão total é maior que depois desse momento. Em outras palavras, a mitigação da anomalia da MAC foi eficiente até o ponto em que o “traffic shapping” provocou uma diminuição da vazão do ofensor 50% e 60%. A partir desse ponto a anomalia foi menos eficiente.

6. CONCLUSÃO

O procedimento de validação comprovou que a bancada de testes funcionou dentro do esperado. Isso significa que:

- O tráfego que o Laptop c1 gerou foi corretamente coletado
- O RSSI do uplink foi corretamente medido
- O BWM comandou corretamente a restrição de tráfego do Laptop c1
- O NAS restringiu o tráfego da sessão do usuário de acordo com o comandado pelo BWM
- O “Captive Portal” registrou a sessão do usuário junto ao AAA corretamente. Além disso, as mensagens de “accounting” foram corretamente enviadas para o AAA.

A bancada se mostrou eficiente em evidenciar a ocorrência da anomalia da MAC. Os dados coletados (colocados em forma de gráfico na Figura 26, Figura 27 e Figura 30) permitiram observar a anomalia da MAC nos 3 experimentos executados. Esta observação é sobre a evolução da vazão total, que sofreu uma queda acentuada a partir do momento que o ofensor iniciou a transmissão (essa queda é efeito da anomalia e indica sua ocorrência).

A anomalia da MAC foi observada tanto com o padrão IEEE 802.11g quanto com IEEE 802.11b. Não foram encontradas na literatura referências a estudos com o padrão IEEE802.11g; neste sentido, esta pode ser considerada uma contribuição dessa dissertação para a área de estudos sobre a anomalia do desempenho em redes sem fio IEEE802.11.

Os resultados obtidos mostraram a ocorrência da anomalia da MAC tanto no uplink quanto no downlink. A literatura encontrada estuda a anomalia na transmissão dos dispositivos móveis para o AP (BRANQUINHO et al., 2006) (GUIRARDELLO, 2008) (HYOGON et al., 2005) (FONTOLAN, 2010) (MOTA et al 2011), porém este trabalho observou o efeito da anomalia nos dois sentidos de

transmissão, incluindo a recepção dos dispositivos móveis (sentido este que pode ser o mais congestionado).

A análise da evolução da vazão total em relação à restrição do tráfego do ofensor (na Figura 32, Figura 33 e Figura 34) permite observar que essa restrição mitigou os efeitos da anomalia da MAC e fez com que a vazão total voltasse para níveis próximos ao da vazão total sem a anomalia.

As sensibilidades calculadas na Tabela 6 mostraram que o controle do tráfego do ofensor é bastante eficiente até o momento de diminuição da taxa de crescimento da vazão total. Após esse momento, a sensibilidade diminui, indicando a menor eficiência dessa técnica de mitigação da anomalia.

A partir deste trabalho, é possível propor um trabalho futuro no sentido de verificar a ocorrência (e mitigação) da anomalia da MAC no “uplink” do padrão IEEE 802.11b, e também no uplink e downlink do novo padrão IEEE 802.11n.

Outra evolução interessante deste trabalho é o desenvolvimento de técnicas de detecção automática de ofensores. Isso permitiria o desenvolvimento de métodos voltados para a mitigação automática da anomalia da MAC.

Finalmente, outra possibilidade de desenvolvimento futuro consiste na ampliação do AAA para desempenhar também a função de “Call Admission Control”.

A área de segurança de redes também pode se beneficiar da arquitetura proposta por este trabalho, pois a estrutura de mitigação da anomalia da MAC pode ser utilizada para mitigar ataques de usuários mal intencionados (impondo restrições ao tráfego desses usuários).

7. REFEFÊNCIAS

8950AAA. *Documentação On-line do 8950AAA,* http://www.8950aaa.com, acessado em 17/6/2011.

ANTON, B; BULLOCK, B; SHORT, J.. *Best Current Practices for Wireless Internet Service Provider (WISP) Roaming*, Wi-Fi Alliance - Wireless ISP Roaming (WISPr) Version: 1.0, 2003.

BRANQUINHO, O. C.; REGGIANI, N.; FERREIRA, D. M.. *Mitigating 802.11 Mac Anomaly Using SNR to Control Backoff Contention Window.* In: IEEE Computer Society, v. 4, p. 55-61, 2006.

COOVACHILLI. *CoovaChilli Project*, http://coova.org/CoovaChilli acessado em 17/6/2011.

DD-WRT. *Documentação On-line do DD-WRT*, http://www.dd-wrt.com/wiki, acessado em 17/6/2011.

FONTOLAN, L. F.. *Política de QOS para redes IEEE802.11 com seleção de taxa de serviço baseada em índice de justiça. 2010*. 108p. Dissertação para obtenção do grau de mestre na Pontifícia Universidade Católica de Campinas, Campinas, 2010.

FREERADIUS. *The FreeRADIUS Project*, http://freeradius.org/, acesso em 20/4/2012

GRASE. *GRASE Hotspot*, http://grasehotspot.org/, acessado em 28/07/2011

GUIRARDELLO, M.. *Política de QoS com Priorização de Acesso ao Meio para Redes IEEE 802.11. 2008*. 104f. Dissertação para obtenção do grau de mestre na Pontifícia Universidade Católica de Campinas, Campinas, 2008.

HEUSSE, M.; ROUSSEAU, F.; BERGER-SABBATEL, G.; DUDA, A.. *Performance Anomaly of 802.11b*. IEEE INFOCOM 2003, 2003.

HYOGON, K.; YUN, S.; KANG, I.; BAHK, S.. *Resolving 802.11 Performance Anomalies through QoS Differentiation. 2005.* IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 7, p655 - 657, JULY 2005

IEEE Std 802.11-2007. *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* – Jun, 2007.

IEEE Std 802.11e-2005. *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements* – Nov, 2005.

IEEE-SA. *IEEE Standards Association*, http://standards.ieee.org, acessado em 28/4/2012.

MOTA, L.; MOTA, A.; FONTOLAN, L. F.. *Quality of Sercice Policy for IEEE802.11 networks with service rate selection based on fairness index. 2011.* JOURNAL OF COMPUTER SCIENCES, VOL. 7, p600 - 604, 2011

NIYATO, D.; HOSSAIN, E.. *A Hierarchical Model for Bandwidth Management and Admission Control in Integrated IEEE 802.16_802.11 Wireless Networks*. In: WCNC 2007 - IEEE Wireless Communications and Networking Conference (WCNC), p. 3766-3770, 2007

NTP. *NTP: The Network Time Protocol*, http://www.ntp.org, acessado em 17/6/2011.

PERIS, A. J. F.; CYRIACO, F. S.; BIAZOTTO, L. H.; BRANQUINHO, O. C.; MOTA, A. A.; MOTA, L. T. M.. *Projeto De Bancada De Testes Para Estudos Em Transmissões Wi-Fi. 2010*. 40th IGIP - International Symposium on Engineering Education, 2010.

RFC 2131. *Dynamic Host Configuration Protocol*, http://tools.ietf.org/html/rfc2131, IETF, 1997, acessado em 25/4/2012

RFC 2865. *Remote Authentication Dial In User Service (RADIUS)*, http://www.ietf.org/rfc/rfc2865.txt, IETF, 2000, acesso em 20/4/2012

RFC 2866. *RADIUS "accounting"*, http://www.ietf.org/rfc/rfc2866.txt, IETF, 2000, acesso em 20/4/2012

RFC 2881. *Network Access Server Requirements Next Generation (NASREQNG) - NAS Model*, http://www.ietf.org/rfc/rfc2881.txt IETF, 2000,, acessado em 28/04/2012.

RFC 2903. *Generic AAA Architecture,* http://www.ietf.org/rfc/rfc2903.txt, IETF, 2000, acesso em 28/4/2012

RFC 5176. *Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*, http://www.ietf.org/rfc/rfc5176.txt, IETF, 2008, acessado em 17/6/2011.

RFC 5905. *Network Time Protocol Version 4: Protocol and Algorithms Specification*, http://www.ietf.org/rfc/rfc5905.txt, IETF, 2010, acessado em 25/4/2012

ROSHAN, P.; LEARY, J.. *802.11 Wireless LAN Fundamentals*, Cisco Press, ISBN: 1-58705-077-3, 2003.

TELECO. *Portal Teleco - Seção: Banda Larga*, http://www.teleco.com.br/Wi-Fi.asp, acessado em 17/1/2012.

UBUNTU. *The Ubuntu Project*, http://www.ubuntu.com, acessado em 17/6/2011.

VANHATUPA, T.. *Design of a Performance Management Model for Wireless Local Area Networks. 2008*. 116f, Thesis for the degree of Doctor of Technology in Tampere University of Technology, Tampere, Finland, 2008.

VMWARE. *VMware Vistualization Software - Seção: Products -> Free Products -> VMware Player*, http://www.vmware.com/products/desktop_virtualization/player/overview.html, acessado em 28/7/2011.

WINSCP. WinSCP - *Free SFTP, SCP and FTP client for Windows,* http://winscp.net, acessado em 27/1/2012

ANEXOS

8. ANEXO A – PROCEDIMENTOS DE INSTALAÇÃO/CONFIGURAÇÃO

8.1. NAS

Equipamento utilizado: • Computador Virtual suportado pelo software VMWare Player versão 3.1.5 (VMware, 2011) instalado num computador AMD Athlon XP, 1,5 GB RAM, HD 160GB (Sistema Operacional Windows XP SP3); 3 interfaces de rede virtuais (2 em modo bridge com interfaces físicas distintas e outra no modo host, para comunicação com o computador 2).

1. Instalar Sistema Operacional: Linux - Ubuntu Server 11.11 i386 (Ubuntu , 2011): Seguir procedimento de instalação da documentação. Durante a instalação, utilizar as seguintes opções quando perguntado:
   a. Acessa Internet;
   b. Informar usuário e senha do sistema (não criar root);
   c. Demais configuração: utilizar opção *default* da instalação.

2. Garantir que o servidor está acessando a Internet.
   Obs.: se necessário, configurar Proxy de acordo com o modelo:
   ```
   # export
      http_proxy=http://<usuário>:<senha>@172.16.0.51:3128
   ```

3. Após a instalação, executar os seguintes comandos:
   ```
   $ sudo apt-get install xinit
   $ sudo apt-get install gnome-session-bin
   $ sudo apt-get install gdm
   $ sudo apt-get install gedit
   $ sudo apt-get install synaptic
   $ sudo apt-get install gnome-terminal
   $ sudo apt-get install expect
   ```

4. Iniciar terminal gráfico:
   ```
   $ gnome-session
   ```

5. No terminal gráfico, através do gerenciador de pacotes synaptic, instalar os seguintes pacotes:
   a. Firefox
   b. Acroread
   c. sun-java-jdk
6. Fazer o download e instalar o pacote coova-chilli (CoovaChilli, 2011) via terminal texto:
   ```
   $ sudo dpkg –i coova-chilli_1.2.6_i386.deb
   ```
7. Configurar o Coova-Chilli:
   a. Arquivo /etc/defaults/chilli, acrescentar/modificar a linha:
      ```
      START_CHILLI=1
      ```
   b. No diretório /etc/chilli:
      ```
      $ cd /etc/chilli
      $ cp defaults config
      $ vi config
      ```
      Modificar as seguintes linhas:
      ```
      HS_WANIF=eth0
      HS_DNS1=10.1.0.1
      HS_UAMALLOW=10.1.0.1
      HS_RADSECRET=secret
      HS_DEFINTERIMINTERVAL=60
      HS_TCP_PORTS=”80 443 22 2812 53 3990 3128”
      ```
   c. Arquivo /etc/init.d/chilli, acrescentar/modificar as linhas:
      ```
      START_CHILLI=1
      DAEMON_ARGS=”—-coaport=3779 --coanoipcheck”
      ```
      Abaixo de start:
      ```
      -- exec $DAEMON $DAEMON_ARGS –c $CONFIG
      ```
8. Instalar o 8950AAA conforme a documentação (8950AAA, 2011). Versão utilizada: 6.6.5.
   a. Caminho de instalação: /opt/8950AAA665
   b. Escolher a opção “Build My Own PolicyFlow”
   c. Para usuário/senha escolher admin/admin

d. Gerar certificados
e. Após instalação:

```
$ sudo ln –s /opt/8950AAA665 /opt/AAA
```

f. Ver item “8.3-Gestor de Autenticação e Largura de Banda” para as configurações específicas do 8950AAA.

9. Criar arquivo de iniciação do 8950AAA (conforme padrão do Linux):
   `/etc/init.d/8950AAA`

10. Criar links de iniciação do CoovaChilli e do 8950AAA:

```
# ln –s /etc/init.d/8950AAA /etc/rc0.d/K19_8950AAA
# ln –s /etc/init.d/8950AAA /etc/rc1.d/K19_8950AAA
# ln –s /etc/init.d/8950AAA /etc/rc6.d/K19_8950AAA
# ln –s /etc/init.d/8950AAA /etc/rc2.d/S50_8950AAA
# ln –s /etc/init.d/8950AAA /etc/rc3.d/S50_8950AAA
# ln –s /etc/init.d/8950AAA /etc/rc4.d/S50_8950AAA
# ln –s /etc/init.d/8950AAA /etc/rc5.d/S50_8950AAA
# ln –s /etc/init.d/chilli  /etc/rc0.d/K20chilli
# ln –s /etc/init.d/chilli  /etc/rc1.d/K20chilli
# ln –s /etc/init.d/chilli  /etc/rc6.d/K20chilli
# ln –s /etc/init.d/chilli  /etc/rc2.d/S90chilli
# ln –s /etc/init.d/chilli  /etc/rc3.d/S90chilli
# ln –s /etc/init.d/chilli  /etc/rc4.d/S90chilli
# ln –s /etc/init.d/chilli  /etc/rc5.d/S90chilli
```

11. Acertar rede:

```
# virsh net-destroy default –undefine
# service libvirt-bin restart
```

a. Arquivo /etc/network/interfaces, acrescentar linhas:

```
auto eth1
iface eth1 inet dhcp
auto eth1:1
iface eth1:1 inet static
address 192.168.3.12
netmask 255.255.255.0
network 192.168.3.0
broadcast 192.168.3.255
```

12. Fazer o download e instalar o pacote grase-repo (GRASE, 2011) via terminal texto:

```
$ sudo dpkg –i grase-repo_1.1.5_all.deb
```

Obs.: se necessário, configurar Proxy de acordo com o modelo:

```
# export
  http_proxy=http://<usuário>:<senha>@172.16.0.51:3128
$ sudo apt-get update
$ sudo apt-get grase-www-portal grase-conf-freeRADIUS
  grase-conf-squid3 grase-conf-open-vpn
```

Obs.: Responder “Sim” (ou “Ok”) para tudo

13. Remover iniciação do squid3 e do freeRADIUS:

    `$ /etc/rc*.d:` apagar todas as ocorrências de “squid” e “freeRADIUS”.

14. Finalizar configuração do Coova-Chilli:

    `$ sudo vi /etc/chilli/ipup.sh` (deixar apenas a última linha)

```
ipt -I POSTROUTING -t nat -o $HS_WANIF -j MASQUERADE
```

Verificar o arquivo /etc/chilli/hs.conf:

```
radiusserver1"localhost"
radiusserver2"localhost"
radiussecret "secret"
radiusauthport  1812
radiusacctport  1813
uamserver    "http://10.1.0.1/grase/uam/hotspot"
radiusnasid  "nas01"

papalwaysok
uamaliasname "grase"
adminupdatefile "/etc/chilli/local.conf"
defidletimeout 600
definteriminterval 60
```

15. Verificar a instalação no PC-cliente (laptop). Exemplo para a rede da PUCC (em que é necessário proxy para acessar Internet):
    a. Configurar o Proxy no web-browser.
        i. Configurar o web-browser para não usar Proxy para 10.1.0.0/16 e 172.16.0.0/16.
    b. Iniciar a sessão no web-browser para o endereço http:// 172.16.0.51 (Deve aparece o portal de captura similar à Figura 23).

8.2. AP

Equipamento: Linksys WRT54g

Firmware utilizado: DD-WRT dd-wrt.v24_micro_generic.bin (DD-WRT,2011)

1. Tela SETUP/Basic Setup
   a. NTP -> enable, UCT-3, DST de acordo com o horário de verão corrente
   b. Local-Ip-Address=192.168.3.10/24
   c. DHCP Server Disable
   d. Assign WAN port do switch: YES
2. Tela WIRELESS/Basic Settings
   a. Wireless SSID -> mestrado2010
3. Tela WIRELESS/Wireless Security
   a. Security mode: WPA2 Personal
   b. WPA Algorit.: TKPI
   c. Shared Key: 12345678
4. Tela SECURITY/Firewall
   a. SPI-Firewall -> disable
   b. Block WAN requests -> OFF (todos)

8.3. Gestor de Autenticação e Largura de Banda

8.3.1. 8950AAA - PolicyFlow

Após a instalação do 8950AAA de acordo com o item “8.1- NAS, passo 8”, alterar os arquivos:

- root@ubuntu:/opt/AAA/run# more method_dispatch

```
RADIUS  Auth                              1                 auth
        readPolicyFlowGlobalProperties
RADIUS  Acct                              4                 acct
        readPolicyFlowGlobalProperties
RADIUS  Auth                              CoA-Request       aaa
        getUserinfo
```

```
RADIUS  Auth                                Disconnect-Request  alu_utilities
        sendRADIUSDisconnectRequest
cron    0,5,10,15,20,25,30,35,40,45,50,55  *                   *              *
        *  ”  aaa  readPolicyFlowGlobalProperties
```

- root@ubuntu:/opt/AAA/run# more auth.pf

```
# ------------------------------------------------------------------------------
# Read policy defined configuration from the external config file
# ------------------------------------------------------------------------------
readPolicyFlowGlobalProperties
        Method-Type = ReadPropertyFile
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = iterateDBServers
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadPropertyFile-CacheMap = "${cache.*} := ${file.*};"
        ReadPropertyFile-Filename = pf.properties
        ReadPropertyFile-Map = <<
             ${user.Policy.*} := ${*};
             ${user.Policy.Database-URLs} := ${Database-URLs} -> toList("LS");
        >>
        ReadPropertyFile-SkipBadProperties = FALSE
        ReadPropertyFile-NewUser = FALSE

# ------------------------------------------------------------------------------
# Loop over the configured DB servers to retrieve user information.
# ------------------------------------------------------------------------------
iterateDBServers
        Method-Type = Failover
        Method-Disabled = FALSE
        Method-On-Success = deleteWispr
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Failover-CacheName = failover
        Failover-ListVariables = user.Policy.Database-URLs
        Failover-Method = readUser
        Failover-SortOrder = Natural
        Failover-SortType = String
        Failover-Retries = 1
        Failover-ErrorThreshold = 2
        Failover-DisabledTimeout = 1m
        Failover-AuthDispScope = Global
        Failover-SuccessMap = <<
             // Map inner avps into outer.
             ${user.*} := ${inner.user.*};
             ${check.*} := ${inner.check.*};
             ${reply.*} := ${inner.reply.*};
        >>

# ------------------------------------------------------------------------------
# Lookup the user from the configured database.
# ------------------------------------------------------------------------------
readUser
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Success = readCheckTemplateFromCache
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
```

```
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ”
        Jdbc-Statement = "select password, check_group, reply_group, check_avps,
        reply_avps from aaadb.users where user_name = ? and user_realm = ?"
        Jdbc-BindMap = <<
             ${1} = ${packet.Base-User-Name};
             ${2} = ${packet.User-Realm:DEFAULT};
        >>
        Jdbc-ResultMap = <<
             ${check.Password}        := ${1};
             ${user.Check-Template}   := ${2};
             ${user.Reply-Template}   := ${3};
             ${user.check-attrs.*}              := ${4};
             ${user.reply-attrs.*}              := ${5};
        >>
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------------
# Read the check template from the cache entry.
# ------------------------------------------------------------------------------
readCheckTemplateFromCache
        Method-Type = ReadCache
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = readReplyTemplateFromCache
        Method-On-Failure = readCheckTemplateFromDB
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadCache-CacheName = checkTemplate
        ReadCache-SearchKey = ${user.Check-Template:DEFAULT}
        ReadCache-Map = <<
             ${check.*} := ${*};

             // overwrite group avps with user avps
             ${check.*} := ${user.check-attrs.*};
             delete ${user.check-attrs};
        >>
        ReadCache-NewUser = FALSE
        ReadCache-Remove = FALSE

# ------------------------------------------------------------------------------
# If check template is not found in the cache, read it from the DB server
# ------------------------------------------------------------------------------
readCheckTemplateFromDB
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Success = writeCheckTemplateToCache
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
```

```
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ”
        Jdbc-Statement = "SELECT avps FROM aaadb.group_avps WHERE name = ? AND
        type = 0"
        Jdbc-BindMap = "${1} = ${user.Check-Template:DEFAULT};"
        Jdbc-ResultMap = <<
             ${user.check.*}      := ${1};
             ${check.*}           := ${1};
             ${user.Connection-Limit} := ${check.Connection-Limit};
             delete ${check.Connection-Limit};

             // overwrite group avps with user avps
             ${check.*} := ${user.check-attrs.*};
             delete ${user.check-attrs};
        >>
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------------
# Write the check group AVPs into cache for faster retrieval.
# ------------------------------------------------------------------------------
writeCheckTemplateToCache
        Method-Type = WriteCache
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = readReplyTemplateFromCache
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        WriteCache-CacheName = checkTemplate
        WriteCache-Map = <<
             ${*} := ${user.check.*};

             // delete after storing in cache.
             delete ${user.check};
        >>
        WriteCache-SearchKey = ${user.Check-Template:DEFAULT}
        WriteCache-EntryTimeout = 1h
        WriteCache-IdleTimeout = 0s
        WriteCache-Replace = TRUE
        WriteCache-NewEntry = TRUE

# ------------------------------------------------------------------------------
# Read the reply template from the cache.
# ------------------------------------------------------------------------------
readReplyTemplateFromCache
        Method-Type = ReadCache
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Failure = readReplyTemplateFromDB
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadCache-CacheName = replyTemplate
```

```
        ReadCache-SearchKey = ${user.Reply-Template:DEFAULT}
        ReadCache-Map = <<
             ${reply.*} := ${*};

             // overwrite group avps with user avps
             ${reply.*} := ${user.reply-attrs.*};
             delete ${user.reply-attrs};
        >>
        ReadCache-NewUser = FALSE
        ReadCache-Remove = FALSE

# ------------------------------------------------------------------------------
# If reply template is not found in the cache, read it from the DB.
# ------------------------------------------------------------------------------
readReplyTemplateFromDB
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Success = writeReplyTemplateToCache
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ‘’
        Jdbc-Statement = "SELECT avps FROM aaadb.group_avps WHERE name = ? AND
        type = 1"
        Jdbc-BindMap = "${1} = ${user.Reply-Template:DEFAULT};"
        Jdbc-ResultMap = <<
             ${user.reply.*}      := ${1};
             ${reply.*}           := ${1};

             // overwrite group avps with user avps
             ${reply.*} := ${user.reply-attrs.*};
             delete ${user.reply-attrs};
        >>
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------------
# Write the check group AVPs into cache for faster retrieval.
# ------------------------------------------------------------------------------
writeReplyTemplateToCache
        Method-Type = WriteCache
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        WriteCache-CacheName = replyTemplate
        WriteCache-Map = <<
             ${*} := ${user.reply.*};

             // delete after storing
```

```
             delete ${user.reply};
        >>
        WriteCache-SearchKey = ${user.Reply-Template:DEFAULT}
        WriteCache-EntryTimeout = 1h
        WriteCache-IdleTimeout = 0s
        WriteCache-Replace = TRUE
        WriteCache-NewEntry = TRUE

deleteWispr
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = checkPassword
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = <<
             ${user.dummy}:=“;
             #delete ${reply.WISPr-Bandwidth-Max-Up};
             #delete ${reply.WISPr-Bandwidth-Max-Down};
        >>
        ReadWrite-NewUser = FALSE

#-------------------------------------------------------------------------
# Check The User's Password
#-------------------------------------------------------------------------
checkPassword
        Method-Type = AuthLocal
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = checkVerifications
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        AuthLocal-UserName = ${request.User-Name}
        AuthLocal-StripMsDomain = TRUE

# ------------------------------------------------------------------------
# Check Other Verification Items
# ------------------------------------------------------------------------
# MethodOnSuccess=CheckuserLimits retirado
checkVerifications
        Method-Type = CheckItems
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = checkUserLimits
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        CheckItems-CheckAll = FALSE

# ------------------------------------------------------------------------
# Check the user's Port-Limit
# ------------------------------------------------------------------------
checkUserLimits
        Method-Type = StateServer
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = CheckUserName
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        StateServer-RequestMap = <<
             ${uss.User-Name} = "${packet.Base-User-Name}@${packet.User-
        Realm:DEFAULT}";
             ${uss.Calling-Station-Id}:=${request.Calling-Station-Id[toList(-
        "),fromList(":")]};
```

```
              ${uss.WISPr-Bandwidth-Max-Up} = ${reply.WISPr-Bandwidth-Max-Up};
              ${uss.WISPr-Bandwidth-Max-Down} = ${reply.WISPr-Bandwidth-Max-Down};
              ${limit.User-Name} = ${user.Connection-Limit:-1};
              ${uss.NAS-Port}:=${request.NAS-Port};
        >>
        StateServer-Event = Auto
        StateServer-KeyAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}+${packet.Normalized-NAS-Port:request.NAS-
        Port:request.NAS-Port-Id}
        StateServer-NasAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}
        StateServer-UserAttribute = ${packet.Base-User-Name}
        StateServer-SessionIdAttribute = ${request.Session-Id:request.Acct-
        Session-Id}
        StateServer-EventTimeAttribute = ${packet.Receipt-
        Time[FormatLocalTimestampWithMillis]}

CheckUserName
        Method-Type = Compare
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = setPortVariaBanda
        Method-On-Failure = StartWispr
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Compare-Input1 = ${request.User-Name}
        Compare-Input2 = ${user.Policy.User-Varia-Banda}
        Compare-Type = Unknown
        Compare-Operator = "=="

setPortVariaBanda
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = StartWispr
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = "${user.policy.Port-Varia-Banda}:=${request.Nas-Port};"
        ReadWrite-NewUser = FALSE

StartWispr
        Method-Type = WriteCache
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        WriteCache-CacheName = Wispr
        WriteCache-Map = <<
              ${wispr}:="54000000";
              ${Port-Varia-Banda}:=${user.policy.Port-Varia-Banda};
        >>
        WriteCache-SearchKey = Wispr
        WriteCache-EntryTimeout = 0s
        WriteCache-IdleTimeout = 0s
        WriteCache-Replace = TRUE
        WriteCache-NewEntry = TRUE
```

- root@ubuntu:/opt/AAA/run# more acct.pf

```
# ------------------------------------------------------------------------
# Read the policy defined configuration from the external config file.
# ------------------------------------------------------------------------
```

```
readPolicyFlowGlobalProperties
        Method-Type = ReadPropertyFile
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = updateUserLimits
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadPropertyFile-CacheMap = "${cache.*} := ${file.*};"
        ReadPropertyFile-Filename = pf.properties
        ReadPropertyFile-Map = <<
                ${user.Policy.*} := ${*};
                ${user.Policy.Database-URLs} := ${Database-URLs} -> toList("LS");
        >>
        ReadPropertyFile-SkipBadProperties = FALSE
        ReadPropertyFile-NewUser = FALSE

# ------------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------------
updateUserLimits
        Method-Type = StateServer
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = CheckMaxBandwdth
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        StateServer-RequestMap = <<
                ${uss.User-Name} := "${packet.Base-User-Name}@${packet.User-
        Realm:DEFAULT}";
                ${uss.NAS-Port}:=${request.NAS-Port};
                ${user.Acct-Input-Octets} := ${uss.Acct-Input-Octets};
                ${user.Acct-Output-Octets} := ${uss.Acct-Output-Octets};
                ${user.Acct-Input-Gigawords} := ${uss.Acct-Input-Gigawords};
                ${user.Acct-Output-Gigawords} := ${uss.Acct-Output-Gigawords};
                ${user.Acct-Session-Time} := ${uss.Acct-Session-Time};
                ${user.RssiAcumulado} := ${uss.RssiAcumulado};
                ${user.NoiseAcumulado} := ${uss.NoiseAcumulado};
                ${user.leiturasOK} := ${uss.leiturasOK};
                ${uss.Acct-Input-Octets} := ${request.Acct-Input-Octets};
                ${uss.Acct-Output-Octets} := ${request.Acct-Output-Octets};
                ${uss.Acct-Input-Gigawords} := ${request.Acct-Input-Gigawords};
                ${uss.Acct-Output-Gigawords} := ${request.Acct-Output-Gigawords};
                ${uss.Acct-Session-Time} := ${request.Acct-Session-Time};
                ${user.WISPr-Bandwidth-Max-Up} := ${uss.WISPr-Bandwidth-Max-Up};
                ${user.WISPr-Bandwidth-Max-Down} := ${uss.WISPr-Bandwidth-Max-
        Down};
                ${uss.RssiAcumulado} := "0";
                ${uss.NoiseAcumulado} := "0";
                ${uss.leiturasOK} := "0";
                ${uss.Calling-Station-Id}:=${request.Calling-Station-Id[toList(-
        "),fromList(":")]};
                ${uss.Acct-Session-Id}:=${request.Session-Id:request.Acct-
        Session-Id};
        >>
        StateServer-Event = Auto
        StateServer-KeyAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}+${packet.Normalized-NAS-Port:request.NAS-
        Port:request.NAS-Port-Id}
        StateServer-NasAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}
        StateServer-UserAttribute = ${packet.Base-User-Name}
        StateServer-SessionIdAttribute = ${request.Session-Id:request.Acct-
        Session-Id}
        StateServer-EventTimeAttribute = ${packet.Receipt-
        Time[FormatLocalTimestampWithMillis]}
```

```
# ------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------
CheckMaxBandwdth
        Method-Type = If
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = calculateBand
        Method-On-Failure = getMaxBandwidth
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        If-Condition = "${user.WISPr-Bandwidth-Max-
        Down[convert(0,false,true)]:false}"
        If-Mode = AND
        If-Reverse = FALSE

# ------------------------------------------------------------------------
# Loop over the configured DB servers to retrieve user information.
# ------------------------------------------------------------------------
getMaxBandwidth
        Method-Type = Failover
        Method-Disabled = FALSE
        Method-On-Success = SetMaxBandwdth
        Method-On-Failure = SetMaxBandwdth
        Method-On-Error = SetMaxBandwdth
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Failover-CacheName = failover
        Failover-ListVariables = user.Policy.Database-URLs
        Failover-Method = auth:readUser
        Failover-SortOrder = Natural
        Failover-SortType = String
        Failover-Retries = 1
        Failover-ErrorThreshold = 2
        Failover-DisabledTimeout = 1m
        Failover-AuthDispScope = Global
        Failover-SuccessMap = <<
                // Map inner avps into outer.
                ${user.WISPr-Bandwidth-Max-Up} := ${inner.reply.WISPr-Bandwidth-
        Max-Up};
                ${user.WISPr-Bandwidth-Max-Down} := ${inner.reply.WISPr-
        Bandwidth-Max-Down};
        >>

# ------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------
SetMaxBandwdth
        Method-Type = RADIUS
        Method-Disabled = FALSE
        Method-On-Success = updateBandwidth
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        RADIUS-ServerAddress = 127.0.0.1:3779
        RADIUS-Secret = secret
        RADIUS-Dictionary = "#default"
        RADIUS-Timeout = 2s
        RADIUS-Retries = 0
        RADIUS-RequestMap = <<
                ${user.WISPr-Bandwidth-Max-Up} := ${user.WISPr-Bandwidth-Max-
        Up:500000};
                ${user.WISPr-Bandwidth-Max-Down} := ${user.WISPr-Bandwidth-Max-
        Down:500000};
```

```
                ${WISPr-Bandwidth-Max-Up} := ${user.WISPr-Bandwidth-Max-Up};
                ${WISPr-Bandwidth-Max-Down} := ${user.WISPr-Bandwidth-Max-Down};
                ${User-Name} := ${request.User-Name};
                ${NAS-Port}:=${request.NAS-Port};
        >>
        RADIUS-SuccessMap = "${reply.*} ?= ${*};"
        RADIUS-FailureMap = <<
                delete ${reply.*};
                ${reply.*} := ${*};
        >>
        RADIUS-ChallengeMap = <<
                delete ${reply.*};
                ${reply.*} := ${*};
        >>
        RADIUS-PacketType = CoA-Request
        RADIUS-ClientAddress = ${server.Local-Address}
        RADIUS-CharSet = 8859_1
        RADIUS-InauthenticFailure = FALSE
        RADIUS-CheckAuthenticator = TRUE
        RADIUS-RemoveTrailingNul = TRUE
        RADIUS-AppendTrailingNul = FALSE
        RADIUS-StrictEncoding = FALSE
        RADIUS-CopyMode = TRUE
        RADIUS-Mib = AUTO
        RADIUS-RecvBufferSize = 262144
        RADIUS-SendBufferSize = 262144
        RADIUS-MessageAuthenticator = FALSE

# ------------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------------
updateBandwidth
        Method-Type = StateServer
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = calculateBand
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        StateServer-RequestMap = <<
                ${uss.WISPr-Bandwidth-Max-Up} := ${user.WISPr-Bandwidth-Max-
        Up:500000};
                ${uss.WISPr-Bandwidth-Max-Down} := ${user.WISPr-Bandwidth-Max-
        Down:500000};
        >>
        StateServer-Event = Auto
        StateServer-KeyAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}+${packet.Normalized-NAS-Port:request.NAS-
        Port:request.NAS-Port-Id}
        StateServer-NasAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}
        StateServer-UserAttribute = ${packet.Base-User-Name}
        StateServer-SessionIdAttribute = ${request.Session-Id:request.Acct-
        Session-Id}
        StateServer-EventTimeAttribute = ${packet.Receipt-
        Time[FormatLocalTimestampWithMillis]}

# ------------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------------
calculateBand
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = calculateBand2
        Method-On-Error = iterateDBServers
        Level-On-Success = Info
```

```
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = <<
                ${request.Acct-Input-Gigawords} = "0";
                ${request.Acct-Output-Gigawords} = "0";
                ${request.Acct-Input-Octets} = "0";
                ${request.Acct-Output-Octets} = "0";

                ${user.Acct-Input-Gigawords} = "0";
                ${user.Acct-Output-Gigawords} = "0";
                ${user.Acct-Input-Octets} = "0";
                ${user.Acct-Output-Octets} = "0";

                ${user.Input-Atual} = "${request.Acct-Input-
        Gigawords[fromUnsigned32]}${request.Acct-Input-Octets[fromUnsigned32]}";
                ${user.Input-Anterior} = "${user.Acct-Input-
        Gigawords[fromUnsigned32]}${user.Acct-Input-Octets[fromUnsigned32]}";
                ${user.Input-Atual} := ${user.Input-Atual[toUnsigned64]};
                ${user.Input-Anterior} := ${user.Input-Anterior[toUnsigned64]};

                ${user.Output-Atual} = "${request.Acct-Output-
        Gigawords[fromUnsigned32]}${request.Acct-Output-Octets[fromUnsigned32]}";
                ${user.Output-Anterior} = "${user.Acct-Output-
        Gigawords[fromUnsigned32]}${user.Acct-Output-Octets[fromUnsigned32]}";
                ${user.Output-Atual} := ${user.Output-Atual[toUnsigned64]};
                ${user.Output-Anterior} := ${user.Output-Anterior[toUnsigned64]};

                ${user.leiturasOK} :=
        ${user.leiturasOK[convert(0,1)]:user.leiturasOK:"1"};
        >>
        ReadWrite-NewUser = FALSE

# ------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------
calculateBand2
        Method-Type = Calculate
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = calculateBand3
        Level-On-Success = Info
        Level-On-Error = Info
        Calculate-Expression = <<
                ${user.Session-Time} = ${request.Acct-Session-Time:0} -
        ${user.Acct-Session-Time:0}
                ${user.Band-Input} := ${user.Input-Atual} - ${user.Input-
        Anterior}
                ${user.Band-Output} := ${user.Output-Atual} - ${user.Output-
        Anterior}
        >>

# ------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------
calculateBand3
        Method-Type = Calculate
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = CalculaMediamW
        Level-On-Success = Info
        Level-On-Error = Info
        Calculate-Expression = <<
                ${user.Band-Input} := ${user.Band-Input} / ${user.Session-
        Time[convert(0,1)]:user.Session-Time:1}
                ${user.Band-Output} := ${user.Band-Output} / ${user.Session-
        Time[convert(0,1)]:user.Session-Time:1}
                ${user.Band-Input} := ${user.Band-Input} * 8
```

```
                ${user.Band-Output} := ${user.Band-Output} * 8
        >>

CalculaMediamW
        Method-Type = Exec
        Method-Disabled = FALSE
        Method-On-Success = iterateDBServers
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Exec-Command = "./CalculaMediamW.sh ${user.RssiAcumulado:0}
        ${user.leiturasOK} ${user.NoiseAcumulado:0} ${user.leiturasOK}"
        Exec-Timeout = 2s
        Exec-ProcessLimit = 10
        Exec-RedirectErrors = TRUE
        Exec-Map = <<
                ${user.exec-status} := ${exit};
                ${user.RssiAcumulado} := "${stdout[toList(LS),get(1)]}";
                ${user.NoiseAcumulado} :=
        "${user.RssiAcumulado[toList(","),get(2)]}";
                ${user.RssiAcumulado} :=
        "${user.RssiAcumulado[toList(","),get(1)]}";
                ${user.NoiseAcumulado} :=
        "${user.NoiseAcumulado[padleft(7,0),nLeft(7)]}";
                ${user.RssiAcumulado} :=
        "${user.RssiAcumulado[padleft(7,0),nLeft(7)]}";
        >>
        Exec-FailOnExit = FALSE
        Exec-NewUser = FALSE

# ------------------------------------------------------------------------------
# Iterate through the configured DB servers to update “accounting” records.
# ------------------------------------------------------------------------------
iterateDBServers
        Method-Type = Failover
        Method-Disabled = FALSE
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Failover-CacheName = failover
        Failover-ListVariables = user.Policy.Database-URLs
        Failover-Method = acct:checkStatus
        Failover-SortOrder = Natural
        Failover-SortType = String
        Failover-Retries = 1
        Failover-ErrorThreshold = 2
        Failover-DisabledTimeout = 1m
        Failover-AuthDispScope = Global

# ------------------------------------------------------------------------------
# Figure out what type of “accounting” packet this is and goto the correct
# Method
# ------------------------------------------------------------------------------
checkStatus
        Method-Type = Branch
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Level-On-Failure = Info
        Level-On-Error = Info
        Branch-Case = <<
                *Start          writeStartRecord
                *Stop           deleteActiveRecord
                *Reject         deleteActiveRecord
                Failed          deleteActiveRecord
                Interim-Update  updateRecord
                accounting-On   errorStarts
                accounting-Off  errorStarts
```

```
                *               writeDetailFile
        >>
        Branch-SelectMode = WILDCARD
        Branch-SearchKey = ${request.Acct-Status-Type}
        Branch-IgnoreCase = TRUE

# ------------------------------------------------------------------------------
# write active records into the active database.  If we fail write to a detail
# file
# ------------------------------------------------------------------------------
writeStartRecord
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Success = RegistraAcctCSV
        Method-On-Failure = writeDetailFile
        Method-On-Error = writeDetailFile
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ‟
        Jdbc-Statement = @Jdbc.acct_insert_active.sql
        Jdbc-BindMap = @Jdbc.acct_insert.map
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------------
# Update Existing Records
# If we fail to update an existing record try to write it as a new record.
# If we encounter an error the write it to a detail file
# ------------------------------------------------------------------------------
updateRecord
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Success = RegistraAcctCSV
        Method-On-Failure = writeStartRecord
        Method-On-Error = writeDetailFile
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ‟
        Jdbc-Statement = <<
                UPDATE aaadb.active SET
                User_Name = '${packet.Base-User-Name}',
                User_Realm = '${packet.User-Realm:DEFAULT}',
                Service_Type = '${request.Service-Type}',
                Framed_Protocol = '${request.Framed-Protocol}',
```

```
                Framed_IP_Address = '${request.Framed-IP-Address}',
                Login_IP_Host = '${request.Login-IP-Host}',
                Login_Service = '${request.Login-Service}',
                Login_TCP_Port = ${request.Login-TCP-Port:NULL},
                Framed_IPX_Network = '${request.Framed-IPX-Network}',
                Class = '${request.Class}',
                Vendor_Specific = '${request.Vendor-Specific}',
                Called_Station_Id = '${request.Called-Station-Id}',
                Calling_Station_Id = '${request.Calling-Station-Id}',
                Acct_Status_Type = '${request.Acct-Status-Type}',
                Acct_Delay_Time = ${request.Acct-Delay-Time:NULL},
                Acct_Input_Octets = ${request.Acct-Input-Octets:NULL},
                Acct_Input_Packets = ${request.Acct-Input-Packets:NULL},
                Acct_Output_Octets = ${request.Acct-Output-Octets:NULL},
                Acct_Output_Packets = ${request.Acct-Output-Packets:NULL},
                Acct_Authentic = '${request.Acct-Authentic}',
                Acct_Session_Time = ${request.Acct-Session-Time:NULL},
                Acct_Terminate_Cause = '${request.Acct-Terminate-Cause}',
                Acct_Multi_Session_Id = '${request.Acct-Multi-Session-Id}',
                Acct_Link_Count = ${request.Acct-Link-Count:NULL},
                Acct_Input_Gigawords = ${request.Acct-Input-Gigawords:NULL},
                Acct_Output_Gigawords = ${request.Acct-Output-Gigawords:NULL},
                NAS_Port_Type = '${request.NAS-Port-Type}',
                Tunnel_Type = '${request.Tunnel-Type}',
                Tunnel_Medium_Type = '${request.Tunnel-Medium-Type}',
                Tunnel_Client_Endpoint = '${request.Tunnel-Client-Endpoint}',
                Tunnel_Server_Endpoint = '${request.Tunnel-Server-Endpoint}',
                Connect_Info = '${request.Connect-Info}',
                LE_Terminate_Detail = '${request.LE-Terminate-Detail}',
                LE_Advice_of_Charge = '${request.LE-Advice-of-Charge}',
                LE_Connect_Detail = '${request.LE-Connect-Detail}',
                LE_IP_Pool = '${request.LE-IP-Pool}',
                Ascend_Dial_Number = '${request.Ascend-Dial-Number}',
                Ascend_Home_Agent_IP_Addr = '${request.Ascend-Home-Agent-IP-
        Addr}',
                Ascend_Home_Agent_UDP_Port = '${request.Ascend-Home-Agent-UDP-
        Port}',
                Ascend_Home_Network_Name = '${request.Ascend-Home-Network-Name}',
                Ascend_Modem_PortNo = '${request.Ascend-Modem-PortNo}',
                Ascend_Modem_SlotNo = '${request.Ascend-Modem-SlotNo}',
                Ascend_Session_Svr_Key = '${request.Ascend-Session-Svr-Key}',
                Ascend_User_Acct_Base = '${request.Ascend-User-Acct-Base}',
                Ascend_User_Acct_Host = '${request.Ascend-User-Acct-Host}',
                Ascend_User_Acct_Key = '${request.Ascend-User-Acct-Key}',
                Ascend_User_Acct_Port = '${request.Ascend-User-Acct-Port}',
                Ascend_User_Acct_Time = '${request.Ascend-User-Acct-Time}',
                Ascend_User_Acct_Type = '${request.Ascend-User-Acct-Type}',
                Ascend_Connect_Progress = '${request.Ascend-Connect-Progress}',
                Ascend_Data_Rate = '${request.Ascend-Data-Rate}',
                Ascend_Disconnect_Cause = '${request.Ascend-Disconnect-Cause}',
                Ascend_Event_Type = '${request.Ascend-Event-Type}',
                Ascend_First_Dest = '${request.Ascend-First-Dest}',
                Ascend_Multilink_ID = '${request.Ascend-Multilink-ID}',
                Ascend_Num_In_Multilink = ${request.Ascend-Num-In-
        Multilink:NULL},
                Ascend_Number_Sessions = ${request.Ascend-Number-Sessions:NULL},
                Ascend_Pre_Input_Octets = ${request.Ascend-Pre-Input-
        Octets:NULL},
                Ascend_Pre_Input_packets = ${request.Ascend-Pre-Input-
        packets:NULL},
                Ascend_Pre_Output_Octets = ${request.Ascend-Pre-Output-
        Octets:NULL},
                Ascend_Pre_Output_packets = ${request.Ascend-Pre-Output-
        packets:NULL},
                Ascend_PreSession_Time = ${request.Ascend-PreSession-Time:NULL},
                Acct_Start_Time = '${packet.Acct-Start-
        Time[toList("/"),fromLIst("-")]}',
```

```
                Acct_Stop_Time = '${packet.Acct-Stop-Time[toList("/"),fromLIst("-
        ")]}'
                WHERE NAS_IP_Address = '${request.NAS-IP-Address:request.NAS-
        Identifier:-1}' AND NAS_Port = '${request.Redback-NAS-Real-
        Port:packet.Normalized-NAS-Port:request.NAS-Port:request.NAS-Port-Id:-1}'
        AND Acct_Session_Id = '${request.Acct-Session-Id}'
        >>
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------------
# The Session has stopped.  Delete the active Session
# ------------------------------------------------------------------------------
deleteActiveRecord
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Success = writeStopRecord
        Method-On-Failure = writeStopRecord
        Method-On-Error = writeStopRecord
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ”
        Jdbc-Statement = "DELETE FROM aaadb.active WHERE NAS_IP_Address = ? AND
        NAS_Port = ? AND ACCT_Session_Id = ?"
        Jdbc-BindMap = <<
                ${1} := ${request.NAS-IP-Address:request.NAS-Identifier};
                ${2} := ${request.NAS-Port:request.NAS-Port-Id:-1};
                ${3} := ${request.Acct-Session-Id};
        >>
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------------
# write inactive records into the “accounting” database. If we fail write to a
# detail file
# ------------------------------------------------------------------------------
writeStopRecord
        Method-Type = Jdbc
        Method-Disabled = FALSE
```

```
        Method-On-Success = RegistraAcctCSV
        Method-On-Failure = writeDetailFile
        Method-On-Error = writeDetailFile
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ”
        Jdbc-Statement = @Jdbc.acct_insert.sql
        Jdbc-BindMap = @Jdbc.acct_insert.map
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------------
# A NAS has rebooted.  Move all open start records to the closed table
# marked as incomplete/error
# ------------------------------------------------------------------------------
errorStarts
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Success = deleteStarts
        Method-On-Failure = deleteStarts
        Method-On-Error = deleteStarts
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ”
        Jdbc-Statement = <<
                INSERT INTO aaadb.”accounting”
                SELECT
                User_Name,
                User_Realm,
                NAS_IP_Address,
                NAS_Port,
                Service_Type,
                Framed_Protocol,
                Framed_IP_Address,
                Login_IP_Host,
                Login_Service,
                Login_TCP_Port,
                Framed_IPX_Network,
                Class,
                Vendor_Specific,
                Called_Station_Id,
                Calling_Station_Id,
                'Error',
                Acct_Delay_Time,
```

```
                Acct_Input_Octets,
                Acct_Input_Packets,
                Acct_Output_Octets,
                Acct_Output_Packets,
                Acct_Session_Id,
                Acct_Authentic,
                Acct_Session_Time,
                Acct_Terminate_Cause,
                Acct_Multi_Session_Id,
                Acct_Link_Count,
                Acct_Input_Gigawords,
                Acct_Output_Gigawords,
                NAS_Port_Type,
                Tunnel_Type,
                Tunnel_Medium_Type,
                Tunnel_Client_Endpoint,
                Tunnel_Server_Endpoint,
                Connect_Info,
                LE_Terminate_Detail,
                LE_Advice_of_Charge,
                LE_Connect_Detail,
                LE_IP_Pool,
                Ascend_Dial_Number,
                Ascend_Home_Agent_IP_Addr,
                Ascend_Home_Agent_UDP_Port,
                Ascend_Home_Network_Name,
                Ascend_Modem_PortNo,
                Ascend_Modem_SlotNo,
                Ascend_Session_Svr_Key,
                Ascend_User_Acct_Base,
                Ascend_User_Acct_Host,
                Ascend_User_Acct_Key,
                Ascend_User_Acct_Port,
                Ascend_User_Acct_Time,
                Ascend_User_Acct_Type,
                Ascend_Connect_Progress,
                Ascend_Data_Rate,
                Ascend_Disconnect_Cause,
                Ascend_Event_Type,
                Ascend_First_Dest,
                Ascend_Multilink_ID,
                Ascend_Num_In_Multilink,
                Ascend_Number_Sessions,
                Ascend_Pre_Input_Octets,
                Ascend_Pre_Input_packets,
                Ascend_Pre_Output_Octets,
                Ascend_Pre_Output_packets,
                Ascend_PreSession_Time,
                Acct_Start_Time,
                Acct_Stop_Time
                FROM active
                WHERE NAS_IP_Address = ?
                AND Acct_Start_Time < ?
        >>
        Jdbc-BindMap = <<
                ${1}            := ${request.NAS-IP-Address:request.NAS-
        Identifier};
                ${2[TIMESTAMP]} := ${packet.Acct-Start-Time:packet.Receipt-Time};
        >>
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
```

```
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------
# Delete the old start records
# ------------------------------------------------------------------------
deleteStarts
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Success = writeBootRecord
        Method-On-Failure = writeBootRecord
        Method-On-Error = writeBootRecord
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ‟
        Jdbc-Statement = "DELETE FROM aaadb.active WHERE NAS_IP_Address = ? AND
        Acct_Start_Time < ?"
        Jdbc-BindMap = <<
                ${1}            := ${request.NAS-IP-Address:request.NAS-
        Identifier};
                ${2[TIMESTAMP]} := ${packet.Acct-Start-Time:packet.Receipt-Time};
        >>
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------
# Put the boot record in the boot table
# ------------------------------------------------------------------------
writeBootRecord
        Method-Type = Jdbc
        Method-Disabled = FALSE
        Method-On-Failure = writeDetailBoot
        Method-On-Error = writeDetailBoot
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Jdbc-Driver = ${user.Policy.Database-Driver}
        Jdbc-Url = ${user.Policy.Database-URLs}
        Jdbc-User = ${user.Policy.Database-User}
        Jdbc-Password = ${user.Policy.Database-
        Password:security.database_login[getPlainTextPassword]}
        Jdbc-ExtraConnectionProperties = ‟
        Jdbc-Statement = "INSERT INTO aaadb.boot VALUES ( ?, ?, ?, ?, ?, ?)"
        Jdbc-BindMap = <<
                ${1[TIMESTAMP]} := ${packet.Acct-Start-Time:packet.Receipt-Time};
                ${2}            := ${request.NAS-IP-Address:request.NAS-
        Identifier};
```

```
                ${3}            := ${request.Acct-Status-Type};
                ${4}            := ${request.Acct-Delay-Time};
                ${5}            := ${request.Acct-Session-Id};
                ${6}            := ${request.Acct-Terminate-Cause};
        >>
        Jdbc-CacheConnections = TRUE
        Jdbc-ConnectionsPerUrl = 1
        Jdbc-ConnectionMaxAge = 0ms
        Jdbc-ConnectionMaxAgeSkew = 0ms
        Jdbc-ConnectingLimit = 1
        Jdbc-ConnectionTimeout = 10s
        Jdbc-StatementTimeout = 10s
        Jdbc-ReuseOnTimeout = FALSE
        Jdbc-TestResult = 1
        Jdbc-TestAllResults = FALSE
        Jdbc-TestOutParameter = 0
        Jdbc-NewUser = FALSE
        Jdbc-BatchMode = FALSE

# ------------------------------------------------------------------------------
# Record any “accounting” reocrds we did not know how to process or that had an
# error while processing
# ------------------------------------------------------------------------------
writeDetailFile
        Method-Type = WriteDetailFile
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = RegistraAcctCSV
        Level-On-Success = Info
        Level-On-Error = Info
        WriteDetailFile-Filename = radaccts/detail
        WriteDetailFile-FilenameExtension = “
        WriteDetailFile-RolloverMode = None
        WriteDetailFile-FormatTimeZone = FALSE
        WriteDetailFile-LogInterim = FALSE
        WriteDetailFile-AutoFlush = TRUE
        WriteDetailFile-AutoSync = FALSE
        WriteDetailFile-IdleTime = 0
        WriteDetailFile-Rename = “
        WriteDetailFile-RevealHiddenAttributes =
        ${server.reveal_hidden_attributes}

RegistraAcctCSV
        Method-Type = If
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = RegistraAcctCSV1
        Method-On-Failure = RegistraAcctCSVHeader
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        If-Condition = <<
                ${user.RssiAcumulado[convert(0000000,false,true)]}
                ${user.NoiseAcumulado[convert(0000000,false,true)]}
        >>
        If-Mode = AND
        If-Reverse = FALSE

RegistraAcctCSV1
        Method-Type = WriteDelimitedFile
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = MonitoraRSSI:ObtemInfoRSSI
        Level-On-Success = Info
        Level-On-Error = Info
        WriteDelimitedFile-Filename = Acct
        WriteDelimitedFile-CSVMode = FALSE
```

```
        WriteDelimitedFile-DelimitChar = "\t"
        WriteDelimitedFile-Map = <<
                ${1} = ${system.Timestamp};
                ${2} = ${request.Calling-Station-Id};
                ${3} =
        ${user.RssiAcumulado[convert(0,#)]:user.RssiAcumulado:"#"};
                ${4} =
        ${user.NoiseAcumulado[convert(0,#)]:user.NoiseAcumulado:"#"};
                ${5} = ${request.Acct-Session-Time:0};
                ${6} = ${request.Acct-Session-Id};
                ${7} = ${user.Session-Time[convert(1,0)]:user.Session-Time};
                ${8} = ${user.Input-Atual};
                ${9} = ${user.Input-Anterior};
                ${10} = ${user.Output-Atual};
                ${11} = ${user.Output-Anterior};
                ${12} = ${user.Band-Input};
                ${13} = ${user.Band-Output};
                ${14} = ${user.WISPr-Bandwidth-Max-Up:"#"};
                ${15} = ${user.WISPr-Bandwidth-Max-Down:"#"};
        >>
        WriteDelimitedFile-FilenameExtension = .log
        WriteDelimitedFile-RolloverMode = Daily
        WriteDelimitedFile-IdleTime = 0
        WriteDelimitedFile-AutoFlush = TRUE
        WriteDelimitedFile-AutoSync = FALSE

RegistraAcctCSVHeader
        Method-Type = WriteDelimitedFile
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = MonitoraRSSI:ObtemInfoRSSI
        Level-On-Success = Info
        Level-On-Error = Info
        WriteDelimitedFile-Filename = Acct
        WriteDelimitedFile-CSVMode = FALSE
        WriteDelimitedFile-DelimitChar = "\t"
        WriteDelimitedFile-Map = <<
                ${1} = "Timestamp";
                ${2} = "MAC";
                ${3} = "Rssi_Average";
                ${4} = "Noise_Average";
                ${5} = "Session_Time";
                ${6} = "Session_Id";
                ${7} = "Elapsed_Session_Time";
                ${8} = "Current_Input";
                ${9} = "Last_Input";
                ${10} = "Current_Output";
                ${11} = "Last_Output";
                ${12} = "Input_Bandwidth";
                ${13} = "Output_Bandwidth";
                ${14} = "Bandwidth-Max-Up";
                ${15} = "Bandwidth-Max-Down";
        >>
        WriteDelimitedFile-FilenameExtension = .log
        WriteDelimitedFile-RolloverMode = Daily
        WriteDelimitedFile-IdleTime = 0
        WriteDelimitedFile-AutoFlush = TRUE
        WriteDelimitedFile-AutoSync = FALSE

# ------------------------------------------------------------------------
# Record any “accounting” reocrds we did not know how to process or that had an
# error while processing
# ------------------------------------------------------------------------
writeDetailBoot
        Method-Type = WriteDetailFile
        Method-Timeout = 0ms
        Method-Disabled = FALSE
```

```
        Level-On-Success = Info
        Level-On-Error = Info
        WriteDetailFile-Filename = radaccts/detail
        WriteDetailFile-FilenameExtension = “
        WriteDetailFile-RolloverMode = None
        WriteDetailFile-FormatTimeZone = FALSE
        WriteDetailFile-LogInterim = FALSE
        WriteDetailFile-AutoFlush = TRUE
        WriteDetailFile-AutoSync = FALSE
        WriteDetailFile-IdleTime = 0
        WriteDetailFile-Rename = “
        WriteDetailFile-RevealHiddenAttributes =
        ${server.reveal_hidden_attributes}
```

- root@ubuntu:/opt/AAA/run# more aaa.pf

```
acceptRadiusAuth
        Method-Type = Return
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Return-Disposition = ACCEPT
        Return-LogLevel = Info
        Return-LogMessage = "Accepting RADIUS Auth Request"

ObtemInfoAcct
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = MonitoraRSSI:ObtemInfoRSSI
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = "${user.Vez}:=-1;"
        ReadWrite-NewUser = FALSE

# ------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------
getUserinfo
        Method-Type = StateServer
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = updateUserLimits
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        StateServer-RequestMap = <<
             ${user.Acct-Session-Id}:=${uss.Acct-Session-Id};
             ${user.Acct-Input-Octets} := ${uss.Acct-Input-Octets};
             ${user.Acct-Output-Octets} := ${uss.Acct-Output-Octets};
             ${user.Acct-Input-Gigawords} := ${uss.Acct-Input-Gigawords};
             ${user.Acct-Output-Gigawords} := ${uss.Acct-Output-Gigawords};
             ${user.Acct-Session-Time} := ${uss.Acct-Session-Time};
             ${user.Calling-Station-Id}:=${uss.Calling-Station-Id};
        >>
        StateServer-Event = None
        StateServer-KeyAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}+${packet.Normalized-NAS-Port:request.NAS-
        Port:request.NAS-Port-Id}
        StateServer-NasAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}
        StateServer-UserAttribute = ${packet.Base-User-Name}
```

```
        StateServer-SessionIdAttribute = ${request.Session-Id:request.Acct-
        Session-Id}
        StateServer-EventTimeAttribute = ${packet.Receipt-
        Time[FormatLocalTimestampWithMillis]}

# ------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------
updateUserLimits
        Method-Type = StateServer
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = preparaCoA
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        StateServer-RequestMap = <<
             ${uss.User-Name} := "${packet.Base-User-Name}@${packet.User-
        Realm:DEFAULT}";
             ${uss.WISPr-Bandwidth-Max-Up} := ${request.WISPr-Bandwidth-Max-Up};
             ${uss.WISPr-Bandwidth-Max-Down} := ${request.WISPr-Bandwidth-Max-
        Down};
             ${uss.Acct-Session-Id}:=${user.Acct-Session-Id};
             ${uss.Acct-Input-Octets} := ${user.Acct-Input-Octets};
             ${uss.Acct-Output-Octets} := ${user.Acct-Output-Octets};
             ${uss.Acct-Input-Gigawords} := ${user.Acct-Input-Gigawords};
             ${uss.Acct-Output-Gigawords} := ${user.Acct-Output-Gigawords};
             ${uss.Acct-Session-Time} := ${user.Acct-Session-Time};
             ${uss.RssiAcumulado} := "0";
             ${uss.NoiseAcumulado} := "0";
             ${uss.leiturasOK} := "0";
             ${uss.Calling-Station-Id}:=${user.Calling-Station-Id};
        >>
        StateServer-Event = Update
        StateServer-KeyAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}+${packet.Normalized-NAS-Port:request.NAS-
        Port:request.NAS-Port-Id}
        StateServer-NasAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}
        StateServer-UserAttribute = ${packet.Base-User-Name}
        StateServer-SessionIdAttribute = ${user.Acct-Session-Id}
        StateServer-EventTimeAttribute = ${packet.Receipt-
        Time[FormatLocalTimestampWithMillis]}

preparaCoA
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = alu_utilities:proxyRadiusDynamicAuthRequest
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = "${request.Acct-Session-Id}:=${user.Acct-Session-Id};"
        ReadWrite-NewUser = FALSE

# ------------------------------------------------------------------------
# Read the policy defined configuration from the external config file.
# ------------------------------------------------------------------------
readPolicyFlowGlobalProperties
        Method-Type = ReadPropertyFile
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = setBanda
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadPropertyFile-CacheMap = "${cache.*} := ${file.*};"
```

```
        ReadPropertyFile-Filename = pf.properties
        ReadPropertyFile-Map = <<
             ${user.Policy.*} := ${*};
             ${user.Policy.Database-URLs} := ${Database-URLs} -> toList("LS");
        >>
        ReadPropertyFile-SkipBadProperties = FALSE
        ReadPropertyFile-NewUser = FALSE

setBanda
        Method-Type = ReadCache
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = checkBanda
        Method-On-Failure = checkBanda
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadCache-CacheName = Wispr
        ReadCache-SearchKey = Wispr
        ReadCache-Map = <<
             ${user.Nova-Banda}:=${Wispr};
             ${user.policy.Port-Varia-Banda}:=${Port-Varia-Banda};
        >>
        ReadCache-NewUser = FALSE
        ReadCache-Remove = FALSE

checkBanda
        Method-Type = Compare
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = initBanda
        Method-On-Failure = nextBanda
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Compare-Input1 = ${user.Nova-Banda}
        Compare-Input2 = 54000000
        Compare-Type = Unknown
        Compare-Operator = "=="

initBanda
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = nextBanda
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = "${user.Nova-Banda}:=${user.policy.Max-Varia-
        Banda:5000000};"
        ReadWrite-NewUser = FALSE

nextBanda
        Method-Type = Calculate
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = setNextBanda
        Level-On-Success = Info
        Level-On-Error = Info
        Calculate-Expression = <<
             ${user.WisprNext} := ${user.Nova-Banda} / 10
             ${user.WisprNext} := ${user.WisprNext} * 9
        >>

setNextBanda
        Method-Type = WriteCache
        Method-Timeout = 0ms
```

```
        Method-Disabled = FALSE
        Method-On-Success = envCoa
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        WriteCache-CacheName = Wispr
        WriteCache-Map = <<
             ${Wispr}:=${user.WisprNext};
             ${Port-Varia-Banda}:=${user.policy.Port-Varia-Banda};
        >>
        WriteCache-SearchKey = Wispr
        WriteCache-EntryTimeout = 0s
        WriteCache-IdleTimeout = 0s
        WriteCache-Replace = TRUE
        WriteCache-NewEntry = TRUE

setBanda1
        Method-Type = RandomNumber
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = envCoa
        Level-On-Success = Info
        Level-On-Error = Info
        RandomNumber-Min = ${user.policy.Min-Varia-Banda}
        RandomNumber-Max = ${user.policy.Max-Varia-Banda}
        RandomNumber-Output = ${user.Nova-Banda}
        RandomNumber-Pad = FALSE

envCoa
        Method-Type = Exec
        Method-Disabled = FALSE
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Exec-Command = "/opt/AAA/bin/setBanda.sh 10.1.0.1 ${user.policy.Port-
        Varia-Banda} ${user.policy.User-Varia-Banda} ${user.Nova-Banda}
        ${user.policy.Intervalo-Varia-Banda}"
        Exec-Timeout = 20s
        Exec-ProcessLimit = 10
        Exec-RedirectErrors = FALSE
        Exec-Directory = /opt/AAA/run
        Exec-FailOnExit = FALSE
        Exec-NewUser = FALSE
```

- root@ubuntu:/opt/AAA/run# more MonitoraRSSI.pf

```
ObtemInfoRSSI
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = IniciaContagem
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = <<
             ${user.N-Vezes}:=20;
             ${user.ajusteIntervalo}:=400;
        >>
        ReadWrite-NewUser = FALSE

IniciaContagem
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = CalculaIntervalo
```

```
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = <<
             ${user.Vez}:=0;
             ${user.RssiAcumulado}:=0;
             ${user.NoiseAcumulado}:=0;
             ${user.leiturasOK}:=0;
        >>
        ReadWrite-NewUser = FALSE

CalculaIntervalo
        Method-Type = Calculate
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = IndicaProximaVez
        Level-On-Success = Info
        Level-On-Error = Info
        Calculate-Expression = <<
             ${user.Intervalo} := 60 / ${user.N-Vezes}
             ${user.UltimoPeriodo}  :=
        ${system.Timestamp[fromLocalTimestamp,toUnsigned32,increment(60)]} -
        ${user.Intervalo}
             ${user.Intervalo} := ${user.Intervalo} * 1000
             ${user.Intervalo} := ${user.Intervalo} - ${user.ajusteIntervalo}
        >>

IndicaProximaVez
        Method-Type = Call
        Method-Disabled = FALSE
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Call-Method = LeRSSItelnet
        Call-WriteMap = <<
             ${user.Vez}:=${user.Vez} + 1;
             ${user.rssi}:=0;
        >>
        Call-ChallengeMap = <<
             delete ${reply.*};
             ${reply.*} := ${reply.*};
        >>
        Call-CopyMode = TRUE

LeRSSItelnet
        Method-Type = Exec
        Method-Disabled = FALSE
        Method-On-Success = ObtemRSSI
        Method-On-Failure = RegistraAcctCSV
        Method-On-Error = RegistraAcctCSV
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Exec-Command = "./ObtemRSSI.sh 192.168.3.10 root admin ${request.Calling-
        Station-Id[toList(\"-\"),fromList(\":\")]}"
        Exec-Timeout = 2s
        Exec-ProcessLimit = 10
        Exec-EnvMap = <<
             ${IP}:="192.168.3.10";
             ${MAC}:=${request.Calling-Station-Id};
        >>
        Exec-RedirectErrors = TRUE
        Exec-Map = <<
             ${user.exec-status} := ${exit};
             ${user.rssi} := ${stdout};
        >>
        Exec-FailOnExit = FALSE
```

```
        Exec-NewUser = FALSE

ObtemRSSI
        Method-Type = PatternMatch
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = CalculamW
        Method-On-Failure = RegistraAcctCSV
        Method-On-Error = RegistraAcctCSV
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        PatternMatch-SearchKey = ${user.rssi[escape]}
        PatternMatch-Mode = REGEX
        PatternMatch-Operation = MATCHES
        PatternMatch-Case = @padraorssi.pat
        PatternMatch-IgnoreCase = TRUE
        PatternMatch-DotAll = FALSE
        PatternMatch-MultiLine = FALSE
        PatternMatch-CanonEquals = FALSE
        PatternMatch-Comments = FALSE
        PatternMatch-Literal = FALSE
        PatternMatch-UnicodeCase = FALSE
        PatternMatch-UnixLines = FALSE

CalculamW
        Method-Type = Exec
        Method-Disabled = FALSE
        Method-On-Success = AcumulaRSSI
        Method-On-Failure = RegistraAcctCSV
        Method-On-Error = RegistraAcctCSV
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        Exec-Command = "./CalculamW.sh ${user.rssi} ${user.RssiAcumulado}
        ${user.noise} ${user.NoiseAcumulado}"
        Exec-Timeout = 2s
        Exec-ProcessLimit = 10
        Exec-RedirectErrors = TRUE
        Exec-Map = <<
             ${user.exec-status} := ${exit};
             ${user.RssiAcumulado} := "${stdout[toList(LS),get(1)]}";
             ${user.NoiseAcumulado}:=
        "${user.RssiAcumulado[toList(","),get(2)]}";
             ${user.RssiAcumulado} :=
        "${user.RssiAcumulado[toList(","),get(1)]}";
        >>
        Exec-FailOnExit = FALSE
        Exec-NewUser = FALSE

AcumulaRSSI
        Method-Type = Calculate
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = updateUserLimits1
        Method-On-Error = RegistraAcctCSV
        Level-On-Success = Info
        Level-On-Error = Info
        Calculate-Expression = "${user.leiturasOK}    := ${user.leiturasOK} + 1"

AcumulaRSSI2
        Method-Type = Calculate
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = updateUserLimits1
        Method-On-Error = AcumulaRSSI3
        Level-On-Success = Info
```

```
        Level-On-Error = Info
        Calculate-Expression = "${user.RssiAcumulado} := ${user.RssiAcumulado} +
        ${user.mW}"

AcumulaRSSI3
        Method-Type = Calculate
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = RegistraAcctCSV
        Level-On-Success = Info
        Level-On-Error = Info
        Calculate-Expression = "${user.leiturasOK}    := ${user.leiturasOK} - 1"

# ------------------------------------------------------------------------------
# Update the USS with “accounting” state information.
# ------------------------------------------------------------------------------
updateUserLimits1
        Method-Type = StateServer
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = VerificaNovoAcct
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        StateServer-RequestMap = <<
             ${uss.RssiAcumulado} := ${user.RssiAcumulado};
             ${uss.NoiseAcumulado} := ${user.NoiseAcumulado};
             ${uss.leiturasOK} := ${user.leiturasOK};
        >>
        StateServer-Event = Auto
        StateServer-KeyAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}+${packet.Normalized-NAS-Port:request.NAS-
        Port:request.NAS-Port-Id}
        StateServer-NasAttribute = ${request.Origin-Host:request.NAS-IP-
        Address:request.NAS-Identifier}
        StateServer-UserAttribute = ${packet.Base-User-Name}
        StateServer-SessionIdAttribute = ${request.Session-Id:request.Acct-
        Session-Id}
        StateServer-EventTimeAttribute = ${packet.Receipt-
        Time[FormatLocalTimestampWithMillis]}

VerificaNovoAcct
        Method-Type = If
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Failure = RegistraAcctCSV
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        If-Condition = "${packet.Last-Disposition-Message[contains(Event not
        handled)]}"
        If-Mode = AND
        If-Reverse = FALSE

RegistraAcctCSV
        Method-Type = Nop
        Method-Disabled = FALSE
        Method-On-Success = VerificaUltimaVez
        Level-On-Success = Info

RegistraAcctCSV1
        Method-Type = WriteDelimitedFile
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = VerificaUltimaVez
        Level-On-Success = Info
        Level-On-Error = Info
```

```
        WriteDelimitedFile-Filename = Acct
        WriteDelimitedFile-CSVMode = FALSE
        WriteDelimitedFile-DelimitChar = "\t"
        WriteDelimitedFile-Map = <<
             ${1} = ${system.Timestamp};
             ${2} = ${request.Calling-Station-Id};
             ${3} = "${user.RssiAcumulado:#}-${user.leiturasOK}-${user.Vez}";
             ${4} = ${request.Acct-Session-Time:0};
             ${5} = ${request.Acct-Session-Id};
             ${6} = ${user.Session-Time[convert(1,0)]:user.Session-Time};
             ${7} = ${user.Input-Atual};
             ${8} = ${user.Input-Anterior};
             ${9} = ${user.Output-Atual};
             ${10} = ${user.Output-Anterior};
             ${11} = ${user.Band-Input};
             ${12} = ${user.Band-Output};
             ${13} = ${user.WISPr-Bandwidth-Max-Up:"#"};
             ${14} = ${user.WISPr-Bandwidth-Max-Down:"#"};
        >>
        WriteDelimitedFile-FilenameExtension = .log
        WriteDelimitedFile-RolloverMode = Daily
        WriteDelimitedFile-IdleTime = 0
        WriteDelimitedFile-AutoFlush = TRUE
        WriteDelimitedFile-AutoSync = FALSE

VerificaUltimaVez
        Method-Type = ReadWrite
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = VerificaUltimaVez2
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        ReadWrite-Map = <<
             if ${user.N-Vezes} > ${user.Vez}
             then
             ${user.Flag-Ultima-Vez}:="FALSE";
             else
             ${user.Flag-Ultima-Vez}:="TRUE";
             endif;

             if ${user.Vez} == 1
             then
             ${user.Flag-1a-Vez}:="TRUE";
             else
             ${user.Flag-1a-Vez}:="FALSE";
             endif;

             if  ${user.UltimoPeriodo} > ${system.Timestamp}-
        >fromLocalTimestamp()->toUnsigned32()
             then
             ${user.Flag-Ultimo-Periodo}:="FALSE";
             else
             ${user.Flag-Ultimo-Periodo}:="TRUE";
             endif;
        >>
        ReadWrite-NewUser = FALSE

VerificaUltimaVez2
        Method-Type = If
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Failure = EsperaProxima
        Level-On-Success = Info
        Level-On-Failure = Info
        Level-On-Error = Info
        If-Condition = <<
```

```
              ${user.Flag-Ultima-Vez}
              ${user.Flag-Ultimo-Periodo}
        >>
        If-Mode = OR
        If-Reverse = FALSE

EsperaProxima
        Method-Type = Delay
        Method-Timeout = 0ms
        Method-Disabled = FALSE
        Method-On-Success = IndicaProximaVez
        Level-On-Success = Info
        Level-On-Error = Info
        Delay-Time = ${user.Intervalo}ms
        Delay-Mode = SUSPEND
```

- root@ubuntu:/opt/AAA/run# more Jdbc.acct_insert.map

```
${1}=${packet.Base-User-Name};
${2}=${packet.User-Realm:DEFAULT};
${3}=${request.NAS-IP-Address:request.NAS-Identifier:-1};
${4[VARCHAR]}=${request.Redback-NAS-Real-Port:packet.Normalized-NAS-
        Port:request.NAS-Port:request.NAS-Port-Id:-1};
${5}=${request.Service-Type};
${6}=${request.Framed-Protocol};
${7}=${request.Framed-IP-Address};
${8}=${request.Login-IP-Host};
${9}=${request.Login-Service};
${10[INTEGER]}=${request.Login-TCP-Port};
${11}=${request.Framed-IPX-Network};
${12}=${request.Class};
${13}=${request.Vendor-Specific};
${14}=${request.Called-Station-Id};
${15}=${request.Calling-Station-Id};
${16}=${request.Acct-Status-Type};
${17[INTEGER]}=${request.Acct-Delay-Time};
${18[NUMERIC]}=${request.Acct-Input-Octets};
${19[NUMERIC]}=${request.Acct-Input-Packets};
${20[NUMERIC]}=${request.Acct-Output-Octets};
${21[NUMERIC]}=${request.Acct-Output-Packets};
${22}=${request.Acct-Session-Id};
${23}=${request.Acct-Authentic};
${24[INTEGER]}=${request.Acct-Session-Time};
${25}=${request.Acct-Terminate-Cause};
${26}=${request.Acct-Multi-Session-Id};
${27[INTEGER]}=${request.Acct-Link-Count};
${28[INTEGER]}=${request.Acct-Input-Gigawords};
${29[INTEGER]}=${request.Acct-Output-Gigawords};
${30}=${request.NAS-Port-Type};
${31}=${request.Tunnel-Type};
${32}=${request.Tunnel-Medium-Type};
${33}=${request.Tunnel-Client-Endpoint};
${34}=${request.Tunnel-Server-Endpoint};
${35}=${request.Connect-Info};
${36}=${request.LE-Terminate-Detail};
${37}=${request.LE-Advice-of-Charge};
${38}=${request.LE-Connect-Detail};
${39}=${request.LE-IP-Pool};
${40}=${request.Ascend-Dial-Number};
${41}=${request.Ascend-Home-Agent-IP-Addr};
${42}=${request.Ascend-Home-Agent-UDP-Port};
${43}=${request.Ascend-Home-Network-Name};
${44}=${request.Ascend-Modem-PortNo};
${45}=${request.Ascend-Modem-SlotNo};
${46}=${request.Ascend-Session-Svr-Key};
```

```
${47}=${request.Ascend-User-Acct-Base};
${48}=${request.Ascend-User-Acct-Host};
${49}=${request.Ascend-User-Acct-Key};
${50}=${request.Ascend-User-Acct-Port};
${51}=${request.Ascend-User-Acct-Time};
${52}=${request.Ascend-User-Acct-Type};
${53}=${request.Ascend-Connect-Progress};
${54}=${request.Ascend-Data-Rate};
${55}=${request.Ascend-Disconnect-Cause};
${56}=${request.Ascend-Event-Type};
${57}=${request.Ascend-First-Dest};
${58}=${request.Ascend-Multilink-ID};
${59[INTEGER]}=${request.Ascend-Num-In-Multilink};
${60[INTEGER]}=${request.Ascend-Number-Sessions};
${61[INTEGER]}=${request.Ascend-Pre-Input-Octets};
${62[INTEGER]}=${request.Ascend-Pre-Input-packets};
${63[INTEGER]}=${request.Ascend-Pre-Output-Octets};
${64[INTEGER]}=${request.Ascend-Pre-Output-packets};
${65[INTEGER]}=${request.Ascend-PreSession-Time};
${66[TIMESTAMP]}=${packet.Acct-Start-Time};
${67[TIMESTAMP]}=${packet.Acct-Stop-Time};
```

- root@ubuntu:/opt/AAA/run# more Jdbc.acct_insert.sql

```
insert into aaadb.”accounting”
(
    User_Name,
    User_Realm,
    NAS_IP_Address,
    NAS_Port,
    Service_Type,
    Framed_Protocol,
    Framed_IP_Address,
    Login_IP_Host,
    Login_Service,
    Login_TCP_Port,
    Framed_IPX_Network,
    Class,
    Vendor_Specific,
    Called_Station_Id,
    Calling_Station_Id,
    Acct_Status_Type,
    Acct_Delay_Time,
    Acct_Input_Octets,
    Acct_Input_Packets,
    Acct_Output_Octets,
    Acct_Output_Packets,
    Acct_Session_Id,
    Acct_Authentic,
    Acct_Session_Time,
    Acct_Terminate_Cause,
    Acct_Multi_Session_Id,
    Acct_Link_Count,
    Acct_Input_Gigawords,
    Acct_Output_Gigawords,
    NAS_Port_Type,
    Tunnel_Type,
    Tunnel_Medium_Type,
    Tunnel_Client_Endpoint,
    Tunnel_Server_Endpoint,
    Connect_Info,
    LE_Terminate_Detail,
    LE_Advice_of_Charge,
    LE_Connect_Detail,
    LE_IP_Pool,
```

```
    Ascend_Dial_Number,
    Ascend_Home_Agent_IP_Addr,
    Ascend_Home_Agent_UDP_Port,
    Ascend_Home_Network_Name,
    Ascend_Modem_PortNo,
    Ascend_Modem_SlotNo,
    Ascend_Session_Svr_Key,
    Ascend_User_Acct_Base,
    Ascend_User_Acct_Host,
    Ascend_User_Acct_Key,
    Ascend_User_Acct_Port,
    Ascend_User_Acct_Time,
    Ascend_User_Acct_Type,
    Ascend_Connect_Progress,
    Ascend_Data_Rate,
    Ascend_Disconnect_Cause,
    Ascend_Event_Type,
    Ascend_First_Dest,
    Ascend_Multilink_ID,
    Ascend_Num_In_Multilink,
    Ascend_Number_Sessions,
    Ascend_Pre_Input_Octets,
    Ascend_Pre_Input_packets,
    Ascend_Pre_Output_Octets,
    Ascend_Pre_Output_packets,
    Ascend_PreSession_Time,
    Acct_Start_Time,
    Acct_Stop_Time
)
values
(
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?
)
```

- root@ubuntu:/opt/AAA/run# more Jdbc.acct_insert_active.sql

```
insert into  aaadb.active
(
    User_Name,
    User_Realm,
    NAS_IP_Address,
    NAS_Port,
    Service_Type,
    Framed_Protocol,
    Framed_IP_Address,
    Login_IP_Host,
    Login_Service,
    Login_TCP_Port,
    Framed_IPX_Network,
    Class,
    Vendor_Specific,
    Called_Station_Id,
    Calling_Station_Id,
    Acct_Status_Type,
    Acct_Delay_Time,
    Acct_Input_Octets,
    Acct_Input_Packets,
    Acct_Output_Octets,
    Acct_Output_Packets,
```

```
    Acct_Session_Id,
    Acct_Authentic,
    Acct_Session_Time,
    Acct_Terminate_Cause,
    Acct_Multi_Session_Id,
    Acct_Link_Count,
    Acct_Input_Gigawords,
    Acct_Output_Gigawords,
    NAS_Port_Type,
    Tunnel_Type,
    Tunnel_Medium_Type,
    Tunnel_Client_Endpoint,
    Tunnel_Server_Endpoint,
    Connect_Info,
    LE_Terminate_Detail,
    LE_Advice_of_Charge,
    LE_Connect_Detail,
    LE_IP_Pool,
    Ascend_Dial_Number,
    Ascend_Home_Agent_IP_Addr,
    Ascend_Home_Agent_UDP_Port,
    Ascend_Home_Network_Name,
    Ascend_Modem_PortNo,
    Ascend_Modem_SlotNo,
    Ascend_Session_Svr_Key,
    Ascend_User_Acct_Base,
    Ascend_User_Acct_Host,
    Ascend_User_Acct_Key,
    Ascend_User_Acct_Port,
    Ascend_User_Acct_Time,
    Ascend_User_Acct_Type,
    Ascend_Connect_Progress,
    Ascend_Data_Rate,
    Ascend_Disconnect_Cause,
    Ascend_Event_Type,
    Ascend_First_Dest,
    Ascend_Multilink_ID,
    Ascend_Num_In_Multilink,
    Ascend_Number_Sessions,
    Ascend_Pre_Input_Octets,
    Ascend_Pre_Input_packets,
    Ascend_Pre_Output_Octets,
    Ascend_Pre_Output_packets,
    Ascend_PreSession_Time,
    Acct_Start_Time,
    Acct_Stop_Time
)
values
(
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
    ?, ?, ?, ?, ?, ?, ?
)
```

- root@ubuntu:/opt/AAA/run# more custom.dict

```
<?xml version="1.0" standalone="no"?>
<!-- Custom Dictionary - this file will not be discarded during upgrades -->
<!DOCTYPE dictionary SYSTEM "dictionary.dtd">

<dictionary codec="#default" overwrite="true">
```

```
    <avp name="WISPr-Location-ID" vendor="WISPr" code="1" type="string"/>
    <avp name="WISPr-Location-Name" vendor="WISPr" code="2" type="string"/>
    <avp name="WISPr-Logoff-URL" vendor="WISPr" code="3" type="string"/>
    <avp name="WISPr-Redirection-URL" vendor="WISPr" code="4" type="string"/>
    <avp name="WISPr-Bandwidth-Min-Up" vendor="WISPr" code="5"
        type="Unsigned32"/>
    <avp name="WISPr-Bandwidth-Min-Down" vendor="WISPr" code="6"
        type="Unsigned32"/>
    <avp name="WISPr-Bandwidth-Max-Up" vendor="WISPr" code="7"
        type="Unsigned32"/>
    <avp name="WISPr-Bandwidth-Max-Down" vendor="WISPr" code="8"
        type="Unsigned32"/>
    <avp name="WISPr-Session-Terminate-Time" vendor="WISPr" code="9"
        type="string"/>
    <avp name="WISPr-Session-Terminate-End-Of-Day" vendor="WISPr" code="10"
        type="string"/>
    <avp name="WISPr-Billing-Class-Of-Service" vendor="WISPr" code="11"
        type="string"/>
</dictionary>
```

- root@ubuntu:/opt/AAA/run# more padraorssi.pat

```
(.*)rssi='(.*)'(.*)noise='(.*)'(.*) ${user.rssi}:=${2};${user.noise}:=${4};
```

- root@ubuntu:/opt/AAA/run# more pf.properties

```
Database-Urls = <<
        jdbc:derby://localhost:1527/provision
>>
Database-Driver = "org.apache.derby.jdbc.ClientDriver"
Database-User = "aaadb"

Acct-Interim-Interval = 60

Intervalo-Varia-Banda = 300 #deve ser igual ao valor definido em cron no
        method_dispach
Max-Varia-Banda = 5000000
Min-Varia-Banda = 50000
Port-Varia-Banda = 2
User-Varia-Banda = "teste2"
```

8.3.2. Scripts utilizados

- root@ubuntu:/opt/AAA/run# more ../bin/setBanda.sh

```
/opt/AAA/bin/aaa-rt -Code "CoA-Request" -Request "User-Name=$3,NAS-Port=$2,NAS-
        Identifier=\"$1\",WISPr-Bandwidth-Max-Up=$4,WISPr-Bandwidth-Max-Down=$4"
```

- root@ubuntu:/opt/AAA/run# more CalculamW.sh

```
#!/bin/bash
rssi=$(echo "$2+e(l(10)*($1/10))"|bc -l)
noise=$(echo "$4+e(l(10)*($3/10))"|bc -l)
echo $rssi,$noise
```

- root@ubuntu:/opt/AAA/run# more ObtemRSSI.sh

```
#!/usr/bin/expect
#Where the script should be run from.
```

```
set timeout 1
#If it all goes pear shaped the script will timeout after 0.5 seconds.

set name [lindex $argv 0]
#First argument is assigned to the variable name

set user [lindex $argv 1]
#Second argument is assigned to the variable user

set password [lindex $argv 2]
#Third argument is assigned to the variable password

set mac [lindex $argv 3]
#Fourth argument is assigned to the variable mac


spawn telnet $name
#This spawns the telnet program and connects it to the variable name


expect "DD-WRT login:"
#The script expects login
send "$user\n"
#The script sends the user variable

expect "Password:"
#The script expects Password
send "$password\n"
#The script sends the user variable

expect "$ "
#The script expects prompt
send "echo rssi=\\'`wl rssi $mac`\\'; echo noise=\\'`wl noise`\\'\n"
#The script sends the command to get rssi

expect "$ "
#The script expects prompt
send "exit\n"
#The script sends the command to exit
```

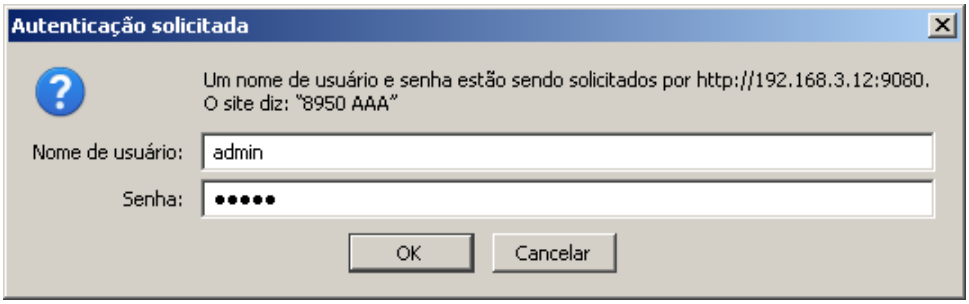
- root@ubuntu:/opt/AAA/run# more CalculaMediamW.sh

```
#!/bin/bash
if [ $1 = "0" ]; then
  rssi=0
else
  rssim=$(echo "$1/$2"|bc -l)
  rssi=$(echo "10*(l($rssim)/l(10))"|bc -l)
fi
if [ $3 = "0" ]; then
  noise=0
else
  noisem=$(echo "$3/$2"|bc -l)
  noise=$(echo "10*(l($noisem)/l(10))"|bc -l)
fi
echo $rssi,$noise
```

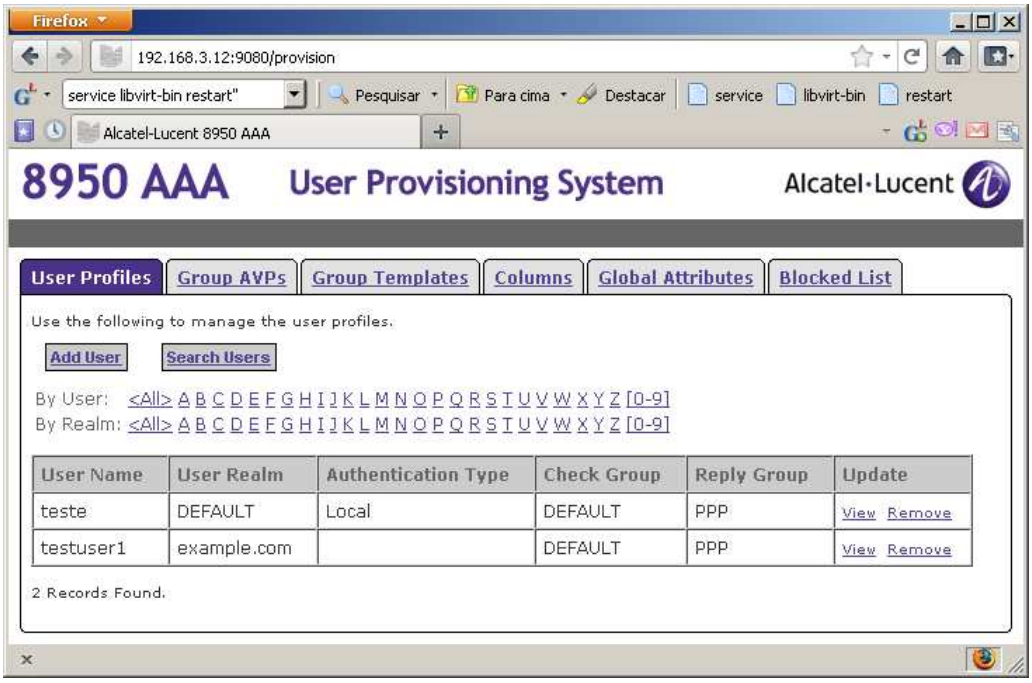
8.3.3. 8950AAA - Aprovisionamento do usuário “teste”

Obs.: O usuário “teste2” deve ser aprovisionado de forma similar.

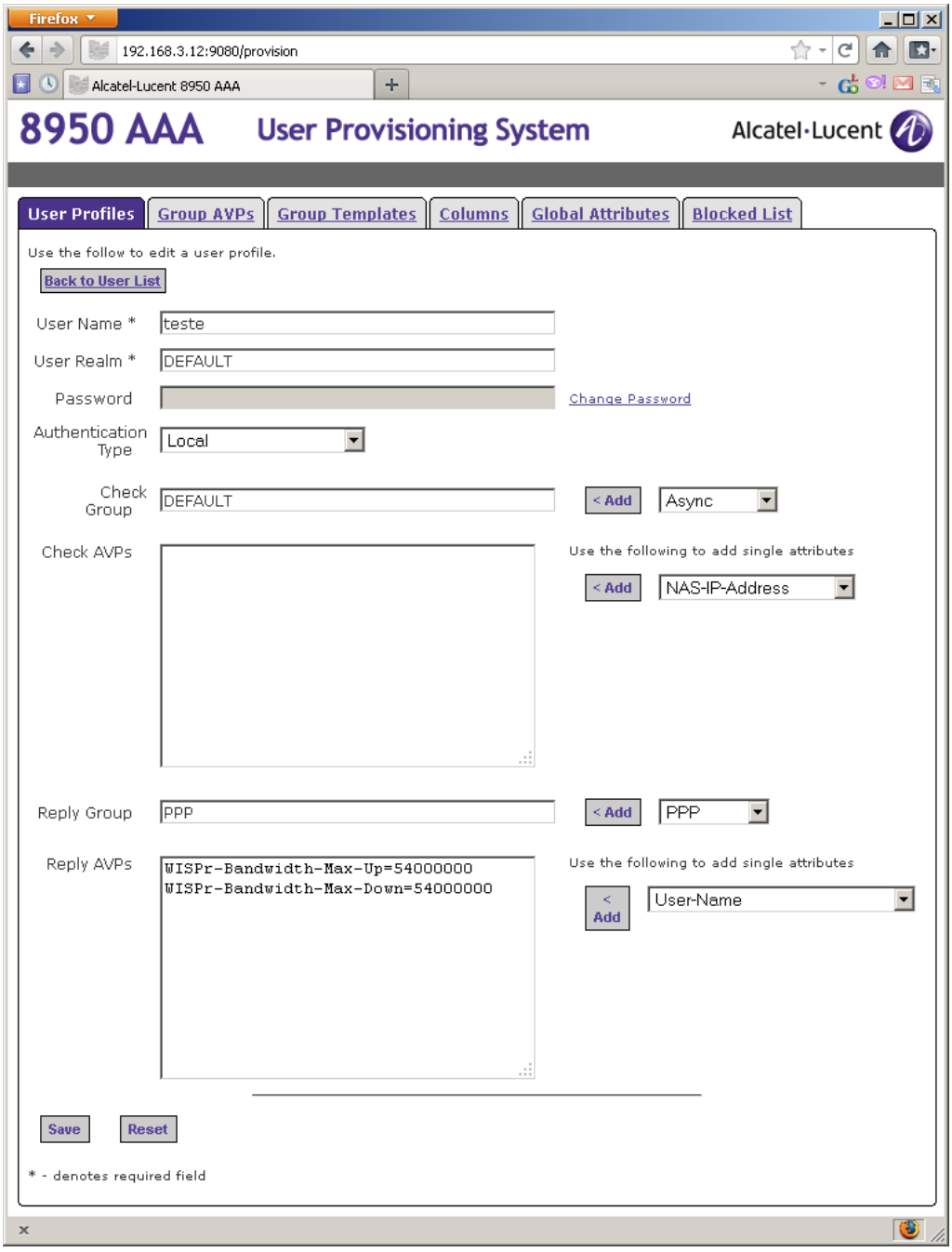
1) No web browser, ir para o endereço . http://192.168.3.12:9080/provision
2) Entrar com usuário e senha do 8950AAA



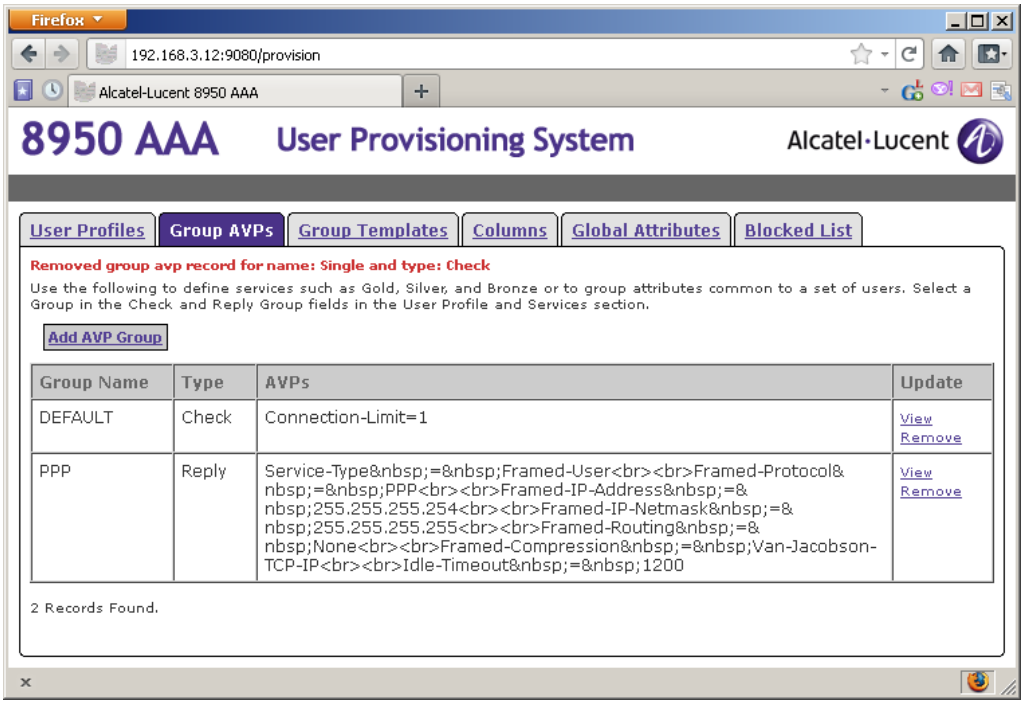
3) Selecionar “User Profiles” e depois “Add User”



4) Digitar dados do usuário “teste” conforme abaixo.



5) Verificar os grupos (em “Group AVPs”) “DEFAULT” e “PPP” conforme abaixo



9. ANEXO B - DETERMINAÇÃO DA TENDÊNCIA DOS PONTOS COLETADOS

Em relação aos pontos coletados nos experimentos conduzidos neste trabalho, a tendência do comportamento dos três conjuntos de pontos foi associada a uma curva polinominal, com a finalidade de facilitar a visualização da trajetória da evolução da vazão (em MBits/s) em razão do tempo decorrido no experimento (em segundos). A curva polinomial selecionada foi aquela que apresentou, para os três conjuntos de pontos (vazão total, vazão na estação c1 e vazão na estação c2), o maior índice R2 para determinar a confiabilidade da tendência e a precisão da previsão.

O índice R2 corresponde ao Coeficiente de Determinação e corresponde a o quadrado do valor da Correlação entre os valores medidos e a curva polinomial. O Coeficiente de Determinação tem valores variando entre 0 e 1. Quanto mais próximo de 1 o valor do índice R2, melhor é a representação da tendência.

As figuras a seguir apresentam a evolução das curvas de tendência para polinômios de ajuste com grau variando de 1 (linear) a 6.

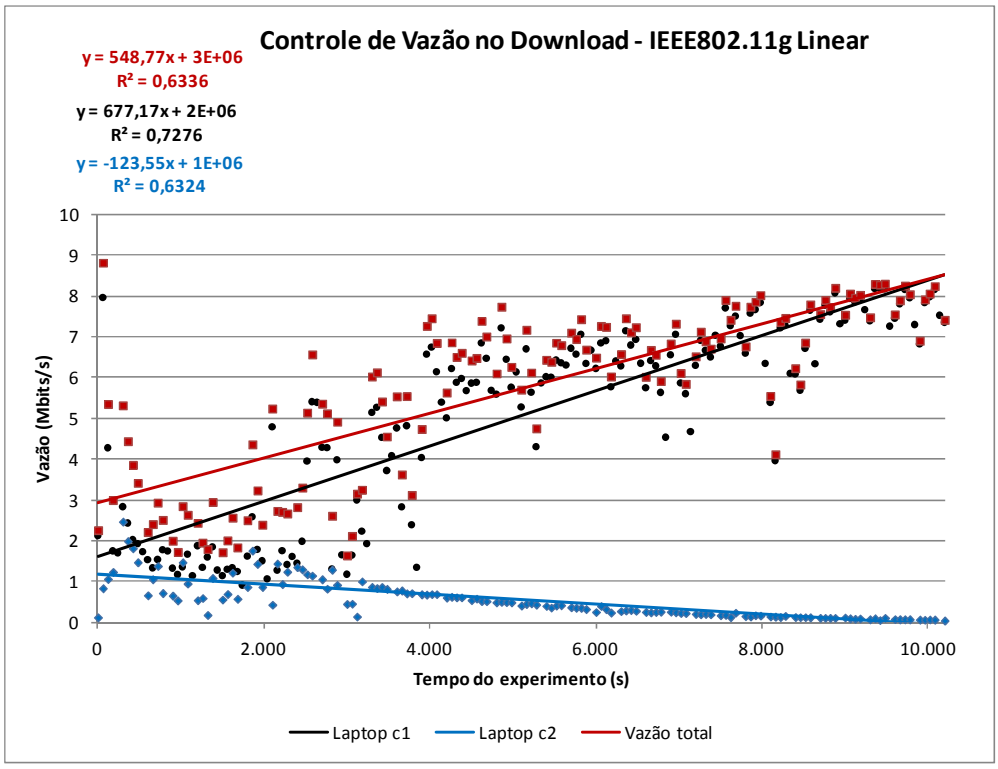


Figura B-1 - Curva de tendência para polinômio de ajuste com grau 1 (linear)

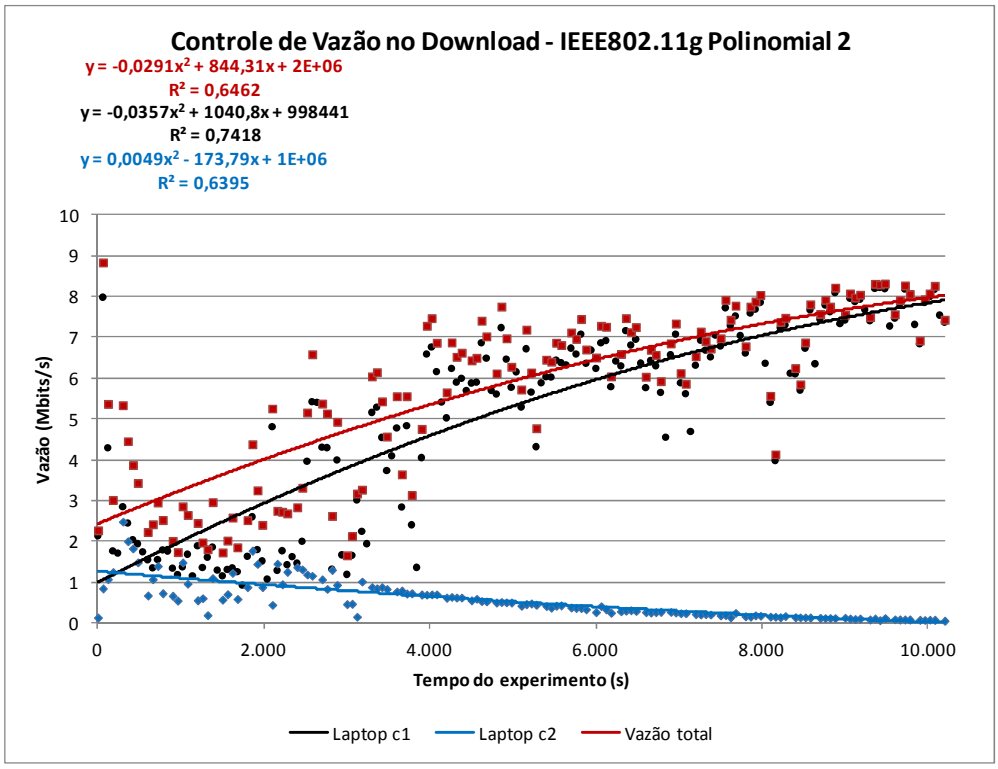


Figura B-2 - Curva de tendência para polinômio de ajuste com grau 2

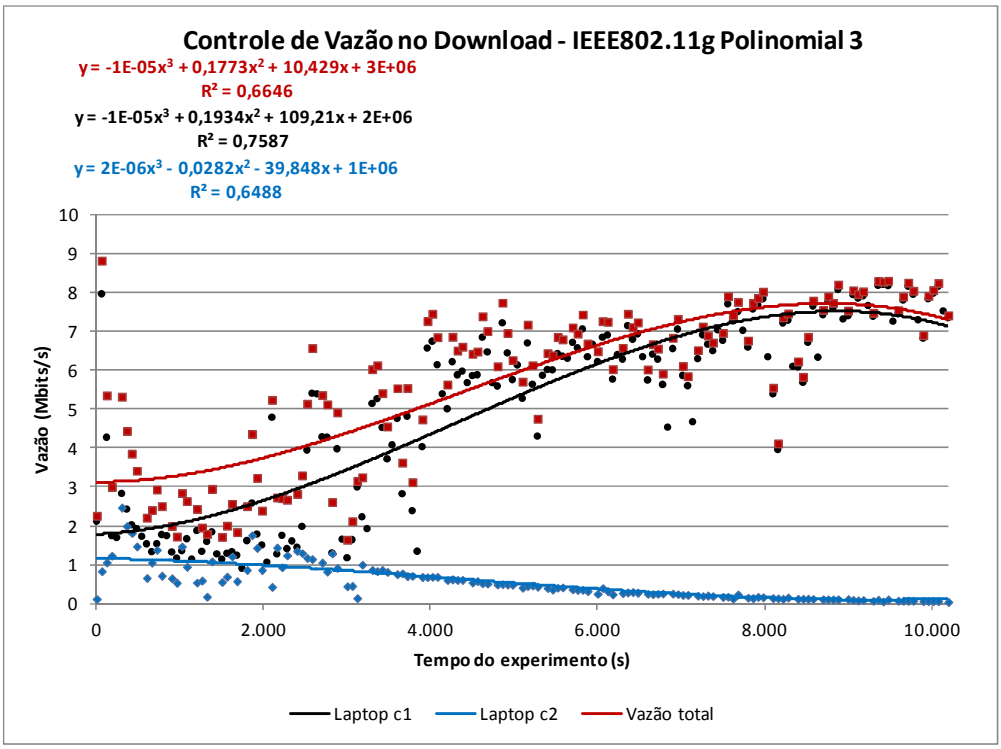


Figura B-3 - Curva de tendência para polinômio de ajuste com grau 3

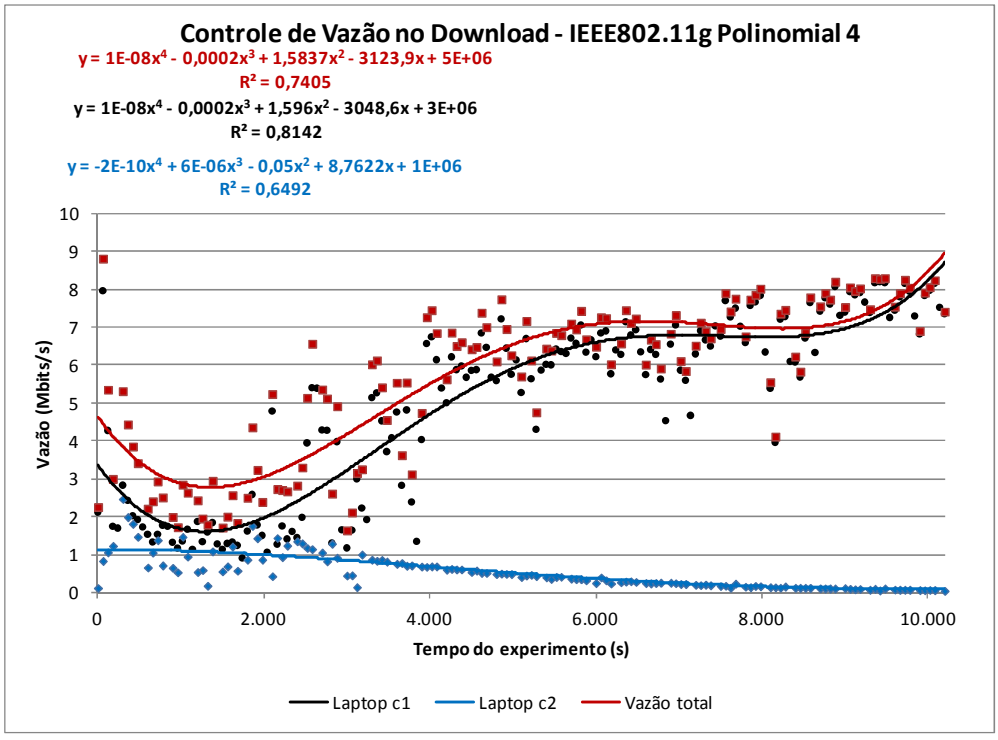


Figura B-4 - Curva de tendência para polinômio de ajuste com grau 4

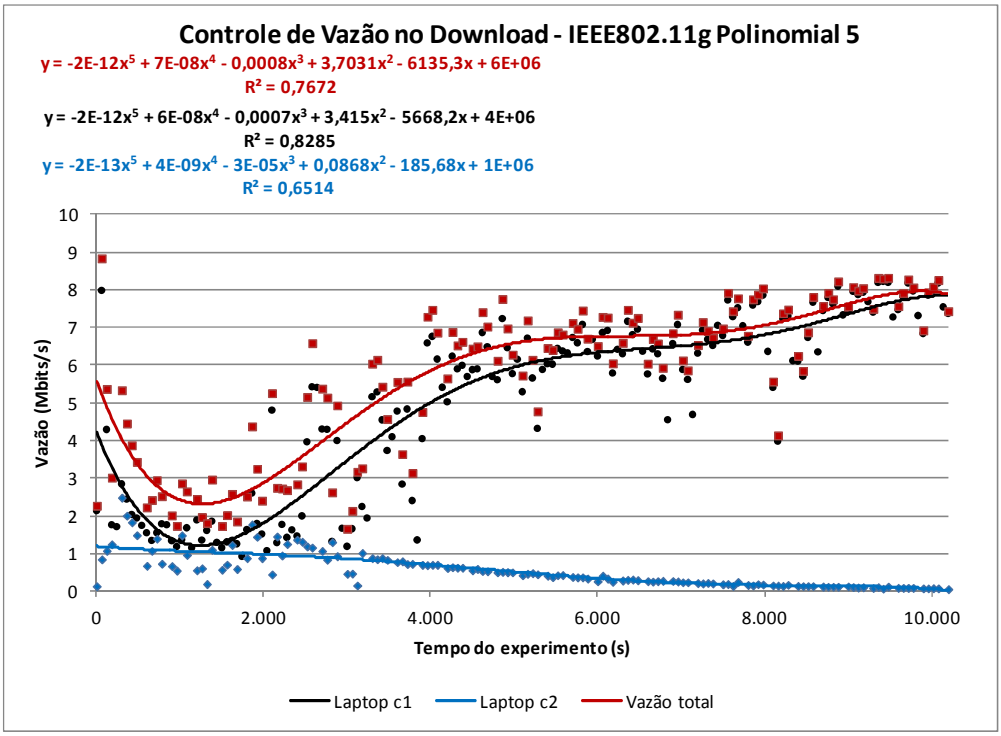


Figura B-5 - Curva de tendência para polinômio de ajuste com grau 5

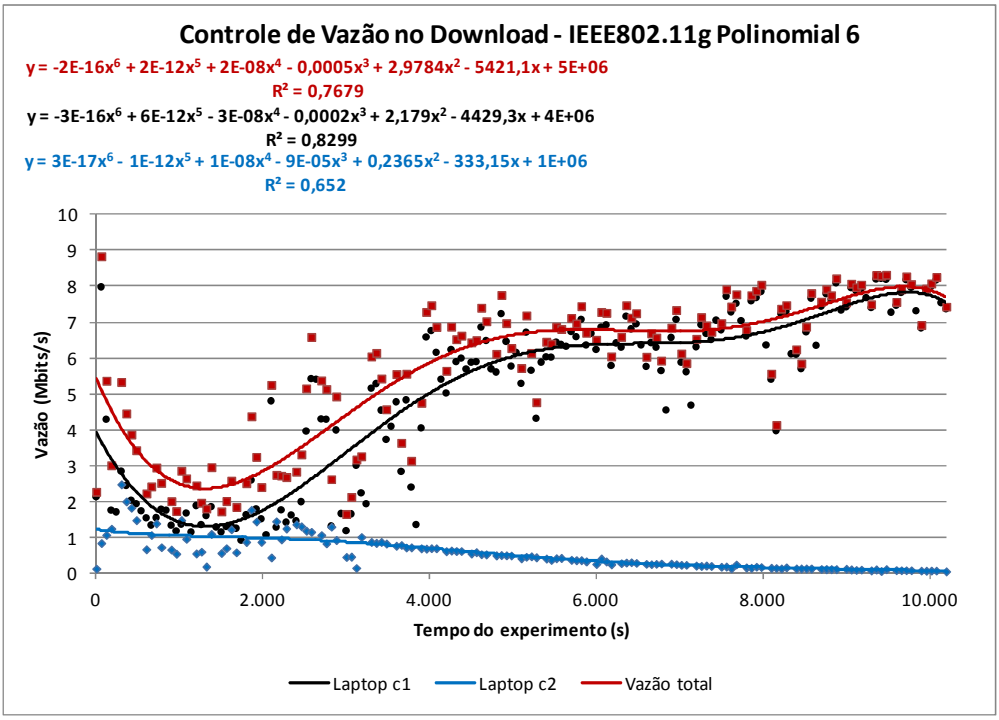


Figura B-6 - Curva de tendência para polinômio de ajuste com grau 6

10. ANEXO C – ESTUDO DA INCLINAÇÃO DAS CURVAS TENDÊNCIA DE VAZÃO ATRAVÉS DAS DERIVADAS DESSAS CURVAS

A partir dos gráficos recortados e com o objetivo de avaliar o comportamento das curvas de tendência, foram calculadas as equações derivadas $\left(\frac{\partial(vazão)}{\partial(tempo\ do\ experimento)}\right)$ das equações das curvas. Essas equações representam as tendências das variações das vazões dos experimentos e são mostradas na Tabela 7 - Equações (1) a (9).

Tabela 7 – Equações da tendência da variação da vazão

Experimento com controle da vazão de Download do ofensor (802.11g)		
Laptop c1	$y = -(5{,}24079134171648\mathrm{E}-15)x^5 + (1{,}36725466530414\mathrm{E}-10)x^4 - (1{,}29903842953338\mathrm{E}-06)x^3 + (5{,}4899653908053\mathrm{E}-03)x^2 - 10{,}2819121195846x + 8270{,}09122970481$	(1)
Laptop c2	$y = -(1{,}67363174442463\mathrm{E}-15)x^5 + (5{,}18657115008902\mathrm{E}-11)x^4 - (6{,}17095228385943\mathrm{E}-07)x^3 + (3{,}4987845371882\mathrm{E}-03)x^2 - 9{,}3380690888887x + 9059{,}99034390482$	(2)
Vazão Total	$y = -(5{,}63849435138991\mathrm{E}-15)x^5 + (1{,}50979375887436\mathrm{E}-10)x^4 - (1{,}49825105250935\mathrm{E}-06)x^3 + (6{,}8292558960704\mathrm{E}-03)x^2 - 14{,}5114080531814x + 12977{,}74478065300$	(3)
Experimento com controle da vazão de Download do ofensor (802.11b)		
Laptop c1	$y = -(7{,}24186177421930\mathrm{E}-16)x^5 - (2{,}11111035205549\mathrm{E}-12)x^4 + (4{,}08220481481973\mathrm{E}-07)x^3 - (5{,}1849810168536\mathrm{E}-03)x^2 + 24{,}0478952069326x - 37091{,}58457319030$	(4)

Laptop c2	$y = (1{,}28607132522758\mathrm{E}-15)x^5 - (4{,}57088280574023\mathrm{E}-11)x^4 + (6{,}38386050734064\mathrm{E}-07)x^3 - (4{,}3841788237162\mathrm{E}-03)x^2 + 14{,}8892312921024x - 20404{,}16743156250$	(5)
Vazão Total	$y = (6{,}96852822705164\mathrm{E}-16)x^5 - (5{,}24661134914860\mathrm{E}-11)x^4 + (1{,}10825183863363\mathrm{E}-06)x^3 - (9{,}9608914896930\mathrm{E}-03)x^2 + 40{,}1166798125605x - 58813{,}25864218340$	(6)
Experimento com controle de vazão de Upload do ofensor		
Laptop c1	$y = -(6{,}05989885222433\mathrm{E}-16)x^5 + (2{,}26015406100501\mathrm{E}-11)x^4 - (3{,}04987955495284\mathrm{E}-07)x^3 + (1{,}8493330354566\mathrm{E}-03)x^2 - 4{,}9572133160845x + 4925{,}15581555140$	(7)
Laptop c2	$y = -(3{,}80952292249889\mathrm{E}-16)x^5 + (9{,}90005896653474\mathrm{E}-12)x^4 - (9{,}46940264930184\mathrm{E}-08)x^3 + (4{,}0016624659803\mathrm{E}-04)x^2 - 0{,}6600326887120x + 79{,}07324976535$	(8)
Vazão Total	$y = -(1{,}47572355948513\mathrm{E}-15)x^5 + (4{,}45868148174484\mathrm{E}-11)x^4 - (5{,}09073642001403\mathrm{E}-07)x^3 + (2{,}6884786503645\mathrm{E}-03)x^2 - 6{,}3572195541432x + 5386{,}78355036924$	(9)

Nessas equações:

- $x = tempo\ do\ experimento$
- $y =$ $Tendência\ da\ variação\ da\ vazão\ (Laptop\ c1, c2\ ou\ total$

Neste trabalho o foco é aumentar a utilização dos recursos do AP. Com isso, os gráficos das tendências das variações das vazões do Laptop c2 e Total - equações (2), (3), (5), (6), (8) e (9) - foram plotados e mostrados na Figura 38, na Figura 39 e na Figura 40.

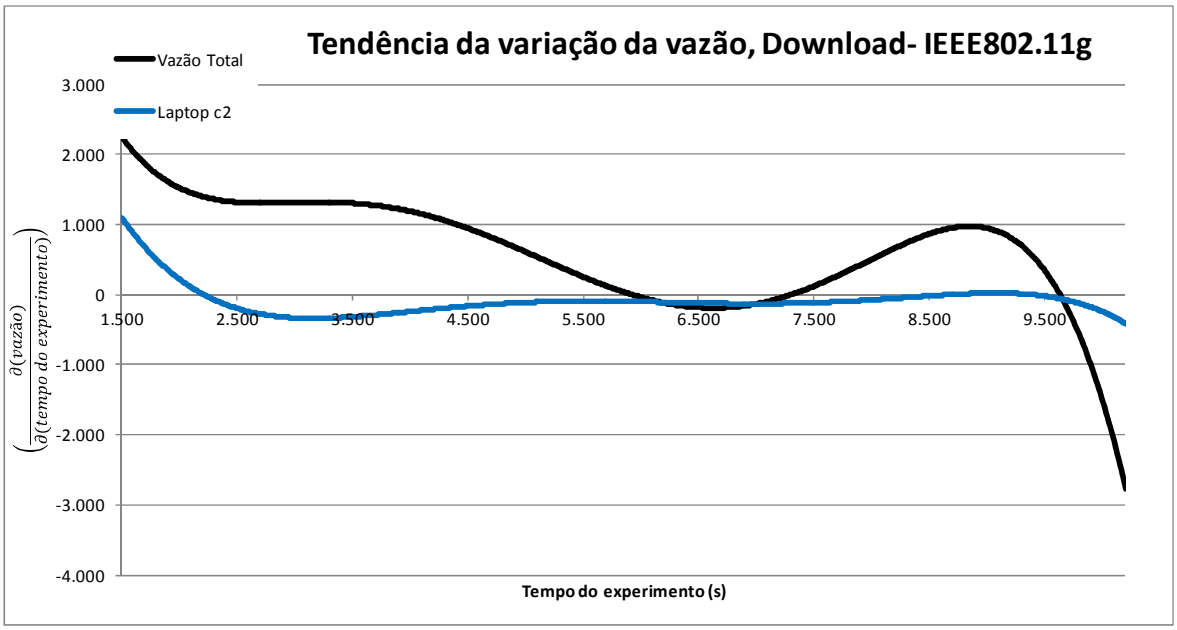


Figura 38 – Tendência da variação da vazão - download, IEEE 802.11g.

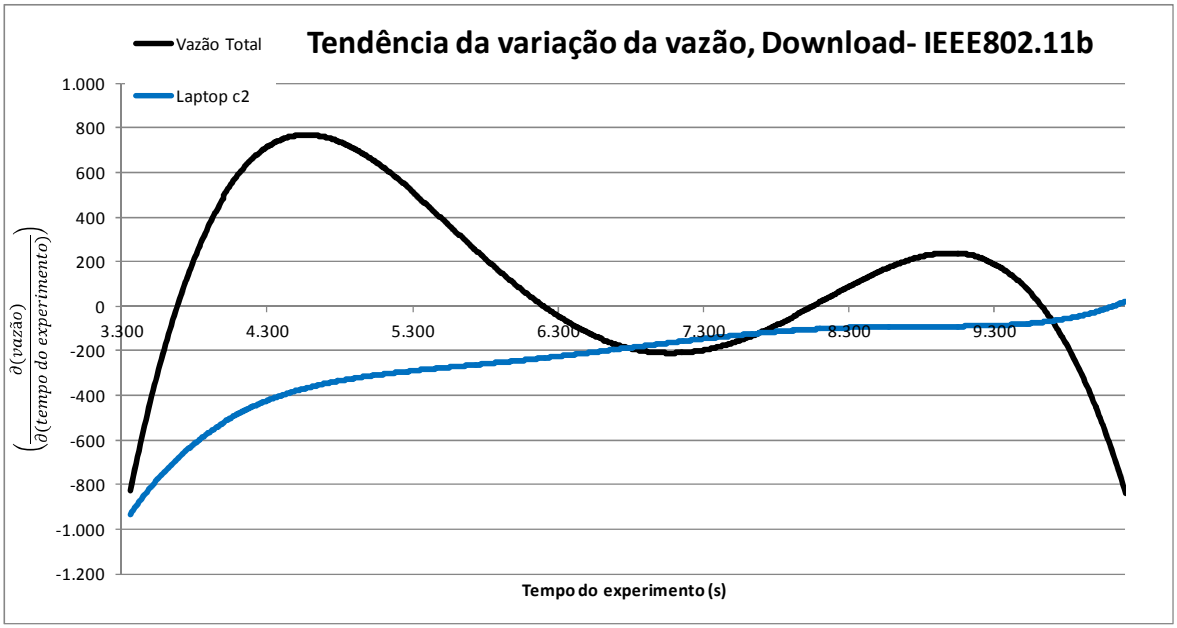


Figura 39 – Tendência da variação da vazão - download, IEEE 802.11b.

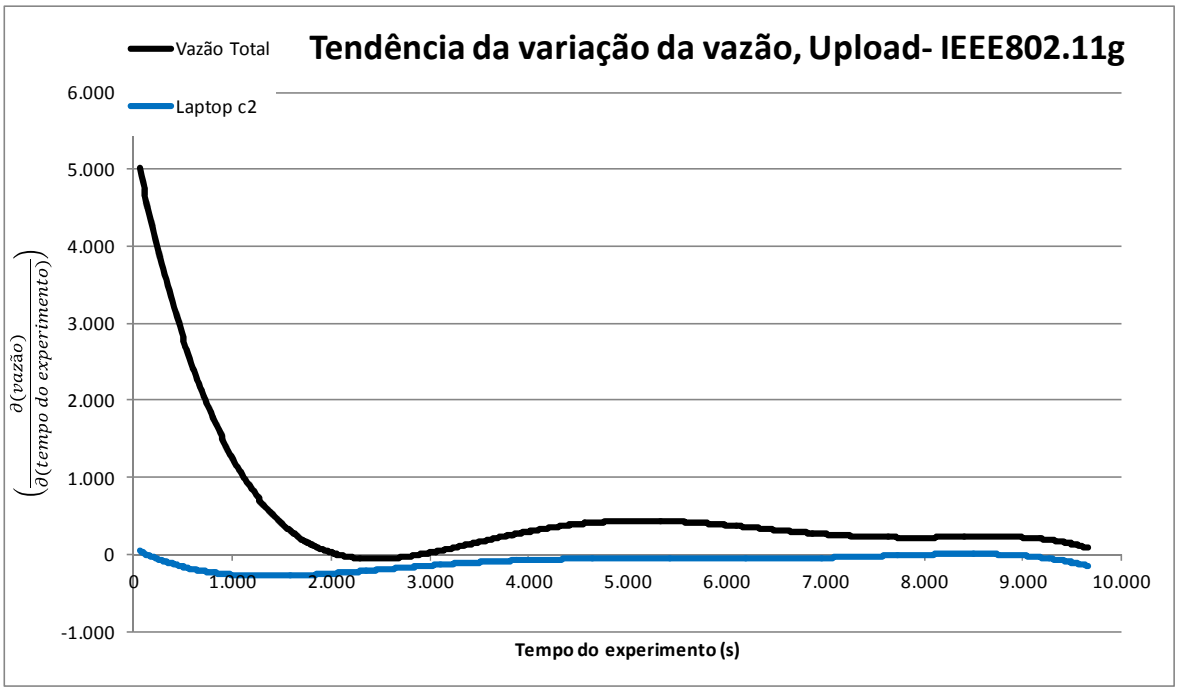


Figura 40 – Tendência da variação da vazão - upload, IEEE 802.11g.

Neste trabalho, define-se como índice de eficiência na mitigação da anomalia da MAC o valor absoluto da relação entre a tendência da variação Vazão Total e a tendência da variação da vazão do Laptop c2. Com base nas equações (1) a (9), estes índices podem ser calculados pelas equações da Tabela 8 - Equações (10) a (12).

Tabela 8 – Indice de eficiência na mitigação da anomalia da MAC

Experimento com controle da vazão de Download do ofensor (802.11g)
$Eficiência = ABS\left(\frac{\begin{matrix}-(5{,}63849435138991\mathrm{E}-15)x^5 \\ +(1{,}50979375887436\mathrm{E}-10)x^4 \\ -(1{,}49825105250935\mathrm{E}-06)x^3 \\ +(6{,}8292558960704\mathrm{E}-03)x^2 \\ -14{,}5114080531814x \\ +12977{,}74478065300\end{matrix}}{\begin{matrix}-(1{,}67363174442463\mathrm{E}-15)x^5 \\ +(5{,}18657115008902\mathrm{E}-11)x^4 \\ -(6{,}17095228385943\mathrm{E}-07)x^3 \\ +(3{,}4987845371882\mathrm{E}-03)x^2 \\ -9{,}3380690888887x \\ +9059{,}99034390482\end{matrix}}\right)$ (10)
Experimento com controle da vazão de Download do ofensor (802.11b)

$$Eficiência = ABS\left(\frac{\begin{matrix}(6{,}96852822705164\mathrm{E}-16)x^5\\-(5{,}24661134914860\mathrm{E}-11)x^4\\+(1{,}10825183863363\mathrm{E}-06)x^3\\-(9{,}9608914896930\mathrm{E}-03)x^2\\+40{,}1166798125605x\\-58813{,}25864218340\end{matrix}}{\begin{matrix}(1{,}28607132522758\mathrm{E}-15)x^5\\-(4{,}57088280574023\mathrm{E}-11)x^4\\+(6{,}38386050734064\mathrm{E}-07)x^3\\-(4{,}3841788237162\mathrm{E}-03)x^2\\+14{,}8892312921024x\\-20404{,}16743156250\end{matrix}}\right) \quad (11)$$

Experimento com controle de vazão de Upload do ofensor

$$Eficiência = ABS\left(\frac{\begin{matrix}-(1{,}47572355948513\mathrm{E}-15)x^5\\+(4{,}45868148174484\mathrm{E}-11)x^4\\-(5{,}09073642001403\mathrm{E}-07)x^3\\+(2{,}6884786503645\mathrm{E}-03)x^2\\-6{,}3572195541432x\\+5386{,}78355036924\end{matrix}}{\begin{matrix}-(3{,}80952292249889\mathrm{E}-16)x^5\\+(9{,}90005896653474\mathrm{E}-12)x^4\\-(9{,}46940264930184\mathrm{E}-08)x^3\\+(4{,}0016624659803\mathrm{E}-04)x^2\\-0{,}6600326887120x\\+79{,}07324976535\end{matrix}}\right) \quad (12)$$

Nessas equações:

- $x = tempo\ do\ experimento$
- $ABS = valor\ absoluto$
- $Eficiência =$ $índice\ de\ eficiência\ conforme\ definido\ neste\ trabalho$

Para melhor visualização do comportamento dos índices de eficiência das equações (10) a (12), estes foram plotados nos gráficos da Figura 41, da Figura 42 e da Figura 43.

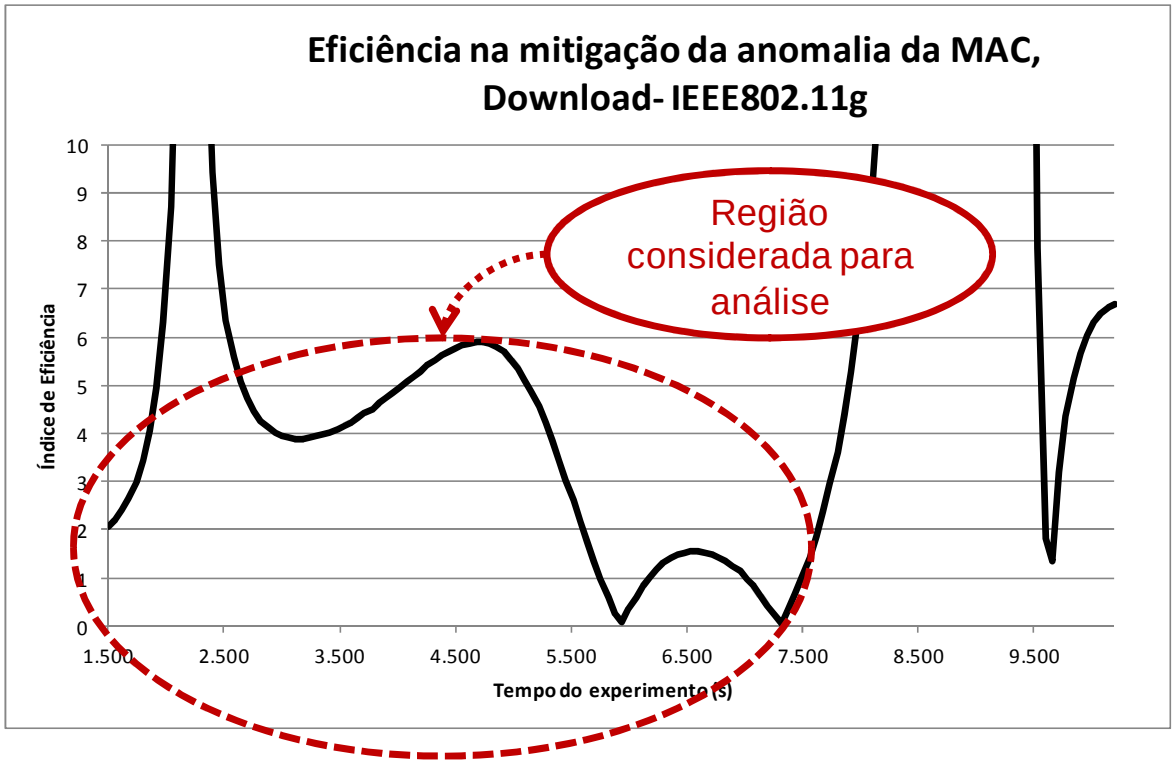


Figura 41 – Índice de eficiência na mitigação da anomalia da MAC, Download-IEEE802.11g.

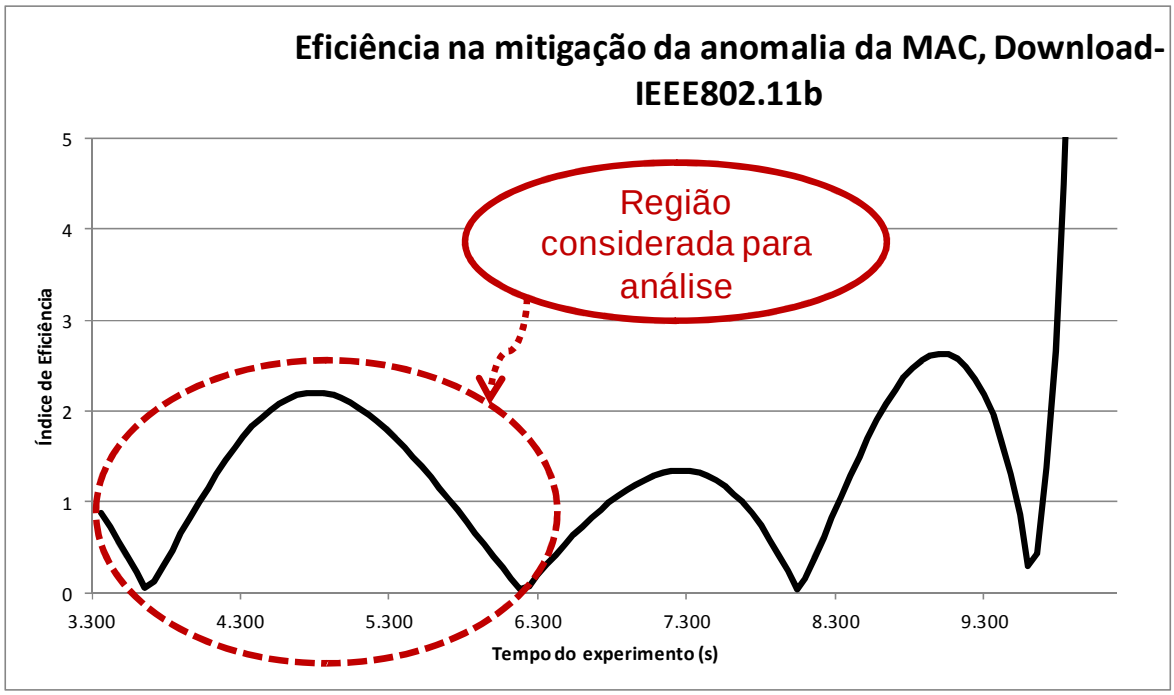


Figura 42 – Índice de eficiência na mitigação da anomalia da MAC, Download-IEEE802.11b.

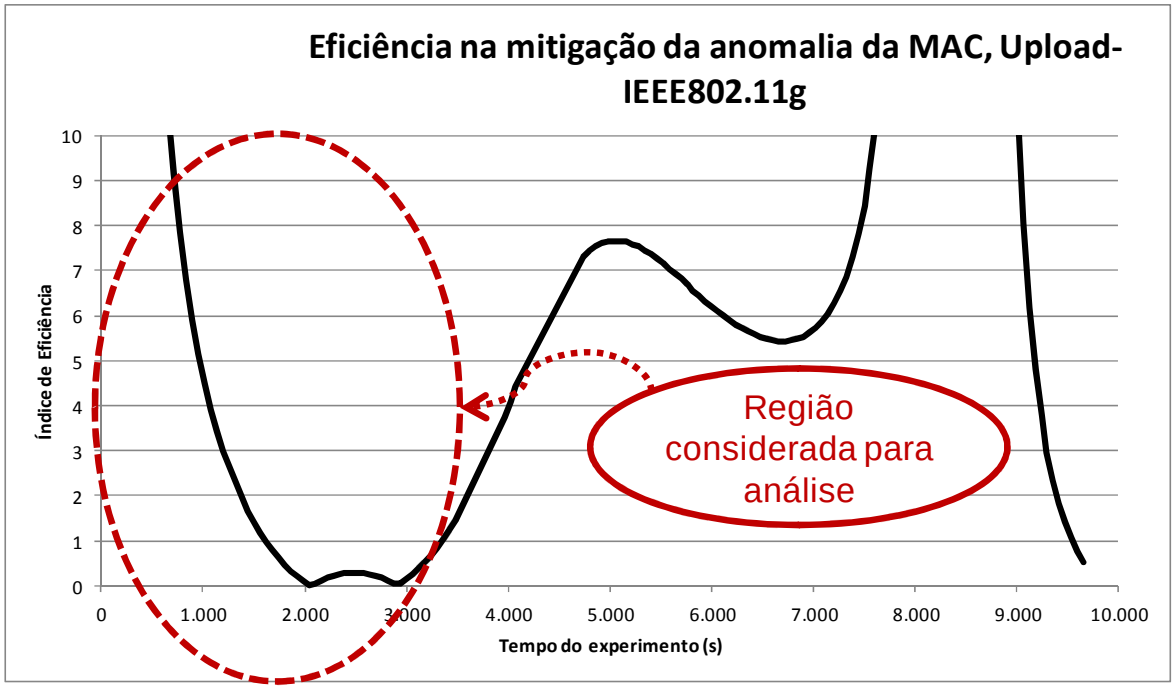


Figura 43 - Índice de eficiência na mitigação da anomalia da MAC, Upload-IEEE802.11g

Em todos os casos a variação do índice tornou-se irregular no final do experimento. Isso acontece por conta de que os valores envolvidos na variação da vazão dos laptops são pequenos algum tempo de experimento. Por isso, a análise do índice de eficiência considerou valores antes dessas irregularidades.

Índices de eficiência maiores são desejados, já que indicam que o aumento da vazão total é maior que a diminuição da vazão de c2. Índices menores que 1 indicam que o ganho da vazão total é menor que a diminuição da vazão de c2; isso que a anomalia da MAC não é mitigada eficazmente.

Analisando os gráficos de eficiência, determinaram-se os instantes em que os valores das eficiências ficaram abaixo de 1. Através dos dados das experiências realizadas, determinou-se, para cada um desses instantes, qual era a vazão de c2 e qual a porcentagem de restrição (em relação ao tráfego máximo – Tabela 2) o NAS estava impondo ao tráfego desse laptop pela equação (13). Esses dados estão na Tabela 9.

$$Restrição_{exp} = 100 * \left(1 - \frac{v_{1s}}{v_{max}}\right) \quad (13)$$

Nessa equação:

- $Restrição_{exp} = Restrição\ imposta\ pelo\ NAS\ no\ experimento$
- $v_{1s} = Vazão\ de\ c2\ no\ instante\ em\ que\ a\ eficiância$ é 1
- $v_{max} = Vazão\ máxima\ de\ c2\ no\ experimento$

Tabela 9 – Valores de vazão e restrição para o instante em que a eficiência ficou abaixo de 1.

Experimento	**Instante de eficiência 1 (s)**	**Valor da vazão (bits/s)**	**Vazão máxima (bits/s)**	**Restrição ao tráfego de c2 (%)**
Experimento com controle da vazão de Download do ofensor (802.11g)	5700	404.955	1.777.664	77%
Experimento com controle da vazão de Download do ofensor (802.11b)	5460	815.412	1.944.928	58%
Experimento com controle de vazão de Upload do ofensor	1500	706.668	1.325.936	47%