UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ALEXANDRE BORTOLIN ARGENTON

Uma Proposta para Gerenciamento de QoS em Redes IEEE802.16

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Prof. Dr. Lisandro Zambenedetti Granville
Orientador

Porto Alegre, março de 2008.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Argenton, Alexandre Bortolin

Uma Proposta para Gerenciamento de QoS em Redes IEEE802.16 / Alexandre Bortolin Argenton – Porto Alegre: Programa de Pós-Graduação em Computação, 2008.

3 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2008. Orientador: Lisandro Zambenedetti Granville.

1.IEEE802.16. 2.QoS. 3.Gerenciamento de redes. I. Granville, Lisandro Zambenedetti. II. Título.

AGRADECIMENTOS

Agradeço a Deus por todas as coisas boas que aconteceram e acontecem em minha vida, e também por todas as pessoas boas que me rodeiam. Se tenho vencido muitos obstáculos e alcançado muitas conquistas, certamente é porque Ele sempre esteve a meu lado.

Agradeço também a meus pais por tudo que representam em minha vida. Deus não poderia ter me dado melhores pais. Agradeço o amor e apoio que tenho deles desde que vim para este mundo e gostaria de dizer que os amo muito.

Agradeço a meus irmãos pela força que sempre me passaram. Minha vida não teria as mesmas alegrias sem eles, sem suas brincadeiras e seu companheirismo. Tenho certeza que estaremos sempre juntos, mesmo que a vida, às vezes, nos mantenha longe.

Agradeço a meus avôs e avós por serem uma extensão de meus pais. Sei que, de onde estão, sempre rezaram, torceram e fizeram tudo a seu alcance para que eu fosse sempre a pessoa feliz que sou.

Agradeço a minha namorada por ser uma pessoa tão especial e que eu amo muito. Ela preenche minha vida de alegria sempre que estamos juntos. Quando vejo seu sorriso, sinto que tudo está iluminado ao redor. Te amo muito, minha flor amada!

Agradeço a meu orientador pela confiança que depositou em mim. Agradeço também pela sugestão do tema que deu origem a este trabalho e pelo apoio que tive durante a realização do mesmo. Tudo flui melhor quando trabalhamos com pessoas competentes.

Durante o mestrado, tive de me mudar para o Rio de Janeiro. Agradeço minha prima Luciana e seu marido, Marcelo, por me receberem tão calorosamente em sua casa durante o período em que eu ainda não dispunha de moradia própria. Sem eles, tudo teria sido muito mais difícil.

Agradeço também à proprietária do imóvel onde resido por todo o auxílio durante a montagem de minha casa no Rio de Janeiro e por se mostrar uma vizinha tão amável e querida.

Agradeço à empresa onde trabalhei anteriormente pela experiência adquirida, que foi fundamental para permitir a realização deste trabalho. Agradeço aos amigos que fiz, em especial aqueles que cursaram o mestrado comigo e que torço para que concluam.

Por fim, agradeço aos demais parentes e amigos por toda sua torcida e apoio, bem como a todos que contribuíram de alguma forma com a realização deste trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
BE	Best Effort
BS	Base Station
BSID	Base Station Identifier
BWA	Broadband Wireless Access
CAC	Connection Admission Control
CBR	Constant Bit Rate
CID	Connection Identifier
COPS	Common Open Policy Service
CPS	Common Part Sublayer
CS	Convergence Sublayer
DHCP	Dynamic Host Configuration Protocol
DMTF	Distributed Management Task Force
DSA	Dynamic Service Addition
DSC	Dynamic Service Change
DSCP	Differentiated Services Code Point
DSD	Dynamic Service Deletion
ex	Exemplo
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplexing
HTTP	Hypertext Transfer Protocol
ID	Identifier
IEEE	Institute of Eletrical and Eletronic Engineer
IETF	Internet Engineering task Force
IP	Internet Protocol
IPC	Interprocess Communication
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MAC	Media Access Control
MAN	Metropolitan Area Network
MDHO	Macro-diversity Handover
MIB	Management Information Base
ms	Milisegundo
MS	Mobile Station
NAT	Network Address Translation
OFDM	Orthogonal Frequency Division Multiplexing

OFDMA	Orthogonal Frequency Division Multiple Access
PCIM	Policy Core Information Model
PCIME	Policy Core Information Model Extensions
PDP	Policy Decision Point
PDU	Protocol Data Unit
PEP	Policy Enforcement Point
PHS	Payload Header Suppression
PMP	Point-to-Multipoint
PoP	Policy of Policies
QAME	QoS-Aware Management Environment
QoS	Quality of Service
RFC	Request For Comments
RSVP	Resource Reservation Protocol
rtPS	Real-time Polling Service
RTT	Round-trip Time
nrtPS	Non-real-time Polling Service
s	Segundo
SAID	Security Association Identifier
SF	Service Flow
SFID	Service Flow Identifier
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SS	Subscriber Station
TDD	Time Division Duplexing
TFTP	Trivial File Transfer Protocol
UGS	Unsolicited Grant Service
VoD	Video on demand
VoIP	Voice Over IP

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

RESUMO

O padrão IEEE802.16 define uma tecnologia para acesso sem fio em banda larga que deve tornar-se bastante popular nos próximos anos. A tecnologia pode resolver o problema de comunicação em áreas de difícil penetração e prover conectividade com Qualidade de Serviço (QoS), mais agilidade e menor custo em muitas outras situações.

Não basta, entretanto, a tecnologia de rede oferecer recursos para Qualidade de Serviço. É necessário que se possa gerenciá-los de maneira simples e eficaz. Sem um estudo que permita desenvolver sistemas de gerenciamento adequados para uma dada tecnologia, o gerenciamento acaba ocorrendo através de ferramentas rudimentares.

Este trabalho propõe e avalia uma forma de gerenciar QoS em redes IEEE802.16. Através do estudo destas redes e de modelos de gerenciamento existentes, foram levantados requisitos relevantes ao gerenciamento deste tipo de rede, e foi proposto o gerenciamento baseado em políticas como possível solução.

A solução proposta engloba recomendações quanto ao que se deveria poder expressar através das políticas, uma reflexão sobre o mapeamento destas políticas na rede e a escolha da arquitetura de gerenciamento baseado em políticas do IETF (Internet Engineering Task Force) para implementação. Questões práticas, como o uso do processo de *ranging* para disparar a configuração das políticas, são abordadas e tratadas no momento em que a arquitetura é apresentada, viabilizando a implementação da solução.

Tendo em vista a escolha do protocolo SNMP para a configuração dos dispositivos, uma MIB de gerenciamento foi proposta para viabilizar o acesso aos parâmetros de configuração necessários. A MIB foi construída sobre outra MIB já existente, adicionando-lhe os objetos e capacidades faltantes. A forma de operação da MIB é apresentada com maiores detalhes para facilitar o entendimento.

Visando validar a solução proposta, o sistema de gerenciamento baseado em políticas QAME foi estendido para permitir o gerenciamento de redes IEEE802.16 dentro da abordagem proposta. A extensão incluiu a implementação de um novo PDP, específico para redes IEEE802.16.

Através da emulação de uma rede IEEE802.16 e do uso do sistema de gerenciamento QAME adaptado, a solução proposta foi avaliada. Pôde-se obter conclusões quanto ao uso de gerenciamento baseado em políticas no gerenciamento de QoS de redes IEEE802.16 e também quanto aos limites de aplicabilidade da arquitetura de gerenciamento proposta.

Palavras-Chave: IEEE802.16, QoS, Gerenciamento de redes

A proposal for QoS management in IEEE802.16 networks

ABSTRACT

The IEEE802.16 standard defines a broadband wireless access technology that should become very popular in the next years. The technology may solve the communication problem in hard access areas and provide connectivity with Quality of Service (QoS), greater agility and lower costs in many other situations.

Offering QoS capabilities is not sufficient to a given network technology. A simple and effective way to manage its capabilities is also required. Without a study that allows the development of adequate management systems, only rudimentary tools will be used for the management task.

This work proposes and evaluates a way to manage QoS on IEEE802.16 networks. By studying IEEE802.16 networks and existing management models, the relevant requirements to manage this kind of network have been gathered and the Policy-Based Network Management (PBNM) model has been proposed as a possible solution.

The proposed solution includes recommendations on what should be possible to express with policies, a thought on how to map the policies on the network, and the choice of the IETF (Internet Engineering Task Force) PBNM architecture. Some practical matters, such as the use of the ranging process to trigger the policy configuration, are considered when the architecture is presented.

Since SNMP has been chosen to configure the devices, a MIB has been proposed to allow access to the required configuration parameters. The MIB has been constructed over a previously existing one by adding the lacking objects and capabilities. The MIB operation is also presented in more details.

In order to validate the proposed solution, the QAME PBNM system has been extended to allow IEEE802.16 network management. The extension included a new PDP, developed specifically to manage IEEE802.16 networks.

By means of an emulated IEEE802.16 network and the use of the adapted QAME management system, the proposed solution has been evaluated. Conclusions have been taken on the use of PBNM to manage QoS for IEEE802.16 networks and on the applicability limits of the proposed architecture.

Keywords: IEEE802.16, QoS, Network management.

1 INTRODUÇÃO

Hoje em dia as redes de computadores estão convergindo para um modelo em que diferentes tipos de serviços são trafegados sobre uma mesma estrutura física. Além dos serviços de dados existentes desde o início, serviços multimídia, como voz e vídeo, têm representado parcelas significativas dos dados transmitidos. Observando-se a popularidade de serviços como Skype e vídeo sob demanda, pode-se dizer que atualmente a Internet já é uma rede de serviços integrados.

Os serviços multimídia introduzem novos requisitos em redes cujos protocolos foram originalmente projetados para o tráfego de dados. Serviços multimídia interativos freqüentemente exigem atrasos pequenos e não admitem muitas perdas. A satisfação de um usuário destes serviços é altamente dependente da QoS (Qualidade de Serviço) entregue pela rede ao fluxo de informações (CISCO SYSTEMS, 2001).

Cientes dos requisitos dos novos fluxos, projetistas de novas tecnologias de rede têm incluído suporte a QoS no nível de enlace, possibilitando que a rede ofereça tratamento diferenciado a fluxos que o necessitarem. Um exemplo de tecnologia que foi inicialmente projetada com suporte a QoS é o padrão IEEE802.16 (IEEE, 2006), cujo estudo será alvo desta dissertação. Também tem havido um esforço em adicionar suporte a QoS em tecnologias que não previram este suporte inicialmente (IEEE, 2005a).

O padrão IEEE802.16 define uma tecnologia para acesso sem fio em banda larga que deve tornar-se bastante popular nos próximos anos. O grande apelo desta tecnologia está na possibilidade de se levar acesso em banda larga a lugares onde o acesso através de redes cabeadas é muito difícil e caro de ser implementado. O padrão permite que a rede opere tanto de forma coordenada por um ponto de acesso central (modo PMP) quanto na forma de uma rede *ad-hoc* (modo *mesh*). O suporte a QoS em nível de enlace foi previsto apenas para a operação no modo PMP.

Para que o suporte a QoS das novas tecnologias seja verdadeiramente aproveitado, é necessário alguma forma de gerenciá-lo. Ao longo dos anos, diversos estudos sobre formas de gerenciar QoS em vários ambientes foram apresentados (MEER et al., 2000; GRANVILLE, 2001; PONNAPPAN et al., 2002; SAMAAN e KARMOUCH, 2003). Entretanto, não se pode saber de antemão a aplicabilidade destas formas de gerenciamento a novas tecnologias, visto que estas podem apresentar requisitos de gerenciamento diversos daqueles considerados durante a realização destes estudos. Exemplos de tecnologias que exigiram modelos de gerenciamento distintos em virtude de suas características são redes *ad-hoc* (FESTOR, 2005a), redes *peer-to-peer* (FESTOR, 2005b) e redes altamente distribuídas (GRANVILLE, 2005). A ausência de

estudos que permitam definir a forma de gerenciamento para uma tecnologia específica freqüentemente leva a utilização de tecnologias rudimentares de gerenciamento, como o uso de terminais remotos para configuração de cada dispositivo individualmente.

Um estudo que proponha a forma adequada de gerenciar a QoS em redes IEEE802.16 é algo que está ausente na literatura. Inicialmente cabe ressaltar que, atualmente, há três versões para o padrão IEEE802.16. O primeiro padrão para operação fixa (IEEE, 2002) foi aprovado em 2001, mas foi revogado em favor do padrão equivalente publicado em 2004 (IEEE, 2004). O padrão publicado em 2006 (IEEE, 2006), uma emenda a seu antecessor (IEEE, 2004), apresenta as especificações necessárias para mobilidade, sendo referenciado como padrão móvel.

O único estudo sobre gerenciamento de redes publicado que trata de IEEE802.16 sugere a utilização do protocolo SNMP para o gerenciamento (HWANG e KIM, 2003). Trata-se de um artigo que analisa genericamente redes de acesso sem fio em banda larga (BWA) e que utiliza a primeira versão do padrão IEEE802.16 (IEEE, 2002) como um estudo de caso. O estudo se restringe a apresentar uma arquitetura de gerenciamento clássica em que a estação gerente interage com o sistema gerenciado via SNMP e apresenta informações através de uma interface gráfica, na qual também é possível alterar-se configurações. Não é feita nenhuma análise a respeito de desempenho ou aplicabilidade da arquitetura frente as particularidades da rede, ela é apenas proposta.

A grande maioria dos estudos que tratam de QoS em redes IEEE802.16 concentram-se na escolha dos algoritmos de escalonamento de pacotes (CICCONETTI et al., 2006; CHEN, JIAO e WANG, 2005; LIU et al., 2005; WONGTHAVARAWAT e GANZ, 2003). Alguns vão um pouco além e apresentam uma arquitetura interna para os elementos da rede (ALAVI, MOJDEH e YAZDANI, 2005; CHU, WANG e MEI, 2002), mas não chegam a tocar no assunto de como gerenciar a QoS. Também há estudos sobre escalonamento de pacotes para redes operando no modo *mesh* (CHEN, CHI e GUO, 2005; WEI et al., 2005), embora este modo não apresente suporte a QoS em nível de enlace. O artigo (GOSH et al., 2005) apresenta uma visão geral do padrão fixo (IEEE, 2004) e faz uma avaliação de sua performance.

Alguns artigos apresentam ainda mecanismos para controle de admissão (WANG, HE e AGRAWAL, 2006; WANG, LI e AGRAWAL, 2005). Estes artigos, no entanto, restringem-se a analisar a capacidade da rede em prover os recursos solicitados, não entrando no mérito de se o solicitante tem ou não o direito a tais recursos, que seria uma questão de gerenciamento.

O artigo mais completo encontrado a respeito de controle de QoS é (CHEN, JIAO e GUO, 2005). Ele apresenta uma arquitetura completa de controle que permite integrar redes IEEE802.16 com as arquiteturas clássicas de provisionamento de QoS da Internet – DiffServ (BLAKE et al., 1998) e IntServ (BRADEN, CLARK e SHENKER, 1994). O foco é dado na integração com a arquitetura IntServ, pois até o mecanismo de sinalização desta arquitetura é integrado ao mecanismo de sinalização das redes IEEE802.16. No caso da arquitetura DiffServ, a integração fica restrita ao mapeamento das classes de serviço DiffServ em seus correspondentes nas redes IEEE802.16. O artigo não apresenta nenhum mecanismo para gerenciamento da QoS, que envolveria indicar a forma como o administrador da rede define a QoS a ser entregue para os fluxos da rede e como o sistema de gerenciamento coloca estas definições em operação.

Algo que convém que se note é que todos os artigos mencionados até o momento tratam, no máximo, do padrão fixo de 2004 (IEEE, 2004). O padrão móvel (IEEE, 2006) é bastante recente, não havendo muitos trabalhos publicados sobre o mesmo até o presente momento. No contexto de QoS, o enfoque sobre este padrão tem sido dado em como manter a QoS dos fluxos em uma situação de *handorver* (LEE, KYAMAKYA e UMONDI, 2006; CHOI et al., 2005). Não há trabalhos que proponham gerenciamento de QoS sobre este padrão.

Detectada a necessidade de estudos que enfoquem em como gerenciar QoS em redes IEEE802.16, a proposta desta dissertação estará em sugerir uma forma de realizar este gerenciamento para o padrão móvel (IEEE, 2006), levando em consideração as particularidades de uma rede operando conforme este padrão. Inicialmente, o capítulo 2 apresentará uma revisão bibliográfica sobre formas de gerenciamento de redes em geral e sobre a tecnologia IEEE802.16 em si. O capítulo 3 irá abordar particularidades das redes IEEE802.16 que tornam esta rede distinta de outras redes existentes, citando em especial requisitos para seu gerenciamento. O capítulo 4 foi reservado para apresentar a solução proposta para a realização do gerenciamento, seguido de capítulos que entram em detalhes sobre uma implementação da proposta (capítulos 5 e 6) e resultados obtidos através desta implementação (capítulo 7). Ao final, o capítulo 8 apresenta as conclusões obtidas durante o desenvolvimento deste trabalho.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo irá apresentar uma revisão sobre modelos de gerenciamento de redes existentes e uma visão geral sobre o funcionamento de redes IEEE802.16. O objetivo principal é familiarizar o leitor com os conceitos que serão utilizados no restante da dissertação.

2.1 Modelos de Gerenciamento

Tendo-se o objetivo de propor uma alternativa para gerenciamento de QoS em redes IEEE802.16, torna-se importante realizar um estudo prévio a respeito das alternativas utilizadas atualmente para gerenciamento de redes. Isto permite que se tenha uma avaliação mais criteriosa sobre a possibilidade de reutilizar um modelo já existente ou sobre a necessidade de definição de um modelo de gerenciamento específico para este tipo de rede. Esta seção apresentará uma breve revisão sobre arquiteturas e modelos de gerenciamento existentes, entrando em maior detalhe na definição do modelo de gerenciamento que se mostrou mais promissor.

Conforme a distribuição das responsabilidades de gerenciamento, um sistema de gerenciamento pode apresentar uma arquitetura centralizada, hierárquica ou distribuída (LEINWAND, 1996). Em uma arquitetura centralizada, toda a responsabilidade de gerenciamento fica centralizada em uma única estação de gerenciamento. Em uma arquitetura hierárquica, parte desta responsabilidade é delegada a estações de gerenciamento secundárias, mais próximas à entidade sendo gerenciada e controladas pela estação de gerenciamento imediatamente superior na hierarquia. Por fim, em uma arquitetura de gerenciamento distribuída, a responsabilidade de gerenciamento é dividida entre um conjunto de estações de gerenciamento, não havendo nenhum tipo de hierarquia entre as mesmas. A Figura 2.1 ilustra as arquiteturas de gerenciamento existentes.

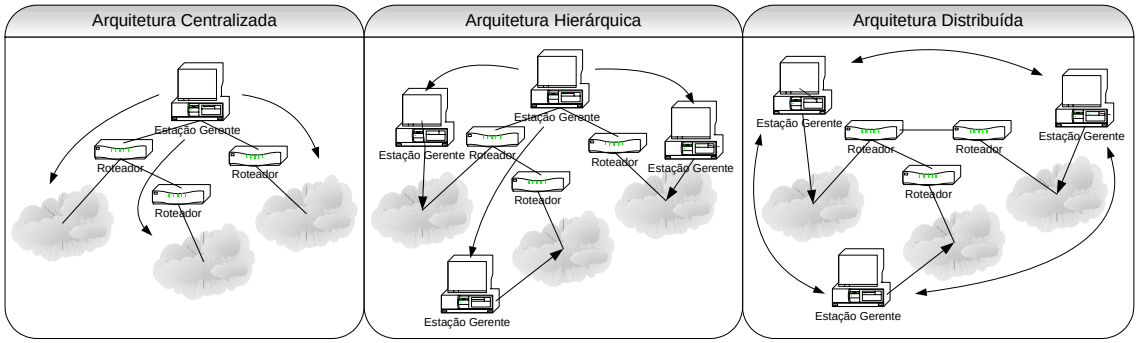


Figura 2.1: Arquiteturas de gerenciamento

Em função da centralização, a arquitetura centralizada é mais simples de ser implementada e permite correlacionar mais facilmente problemas de rede. A desvantagem da arquitetura centralizada é concentrar tráfego nas proximidades da estação gerente, o que a torna uma arquitetura não escalável. Além disto, esta estação gerente também acaba representando um ponto único de falha no sistema de gerenciamento. Por estes motivos, a utilização de arquiteturas hierárquicas ou distribuídas tende a ser mais interessante.

Pode-se encontrar na literatura uma série de modelos de gerenciamento de redes, cada qual criado tendo-se em mente a solução de alguma situação específica. Saindo do modelo de gerenciamento clássico, em que a configuração dos elementos de rede é feita diretamente pelos gerentes através do sistema de gerenciamento, foram criados, entre outros, modelos de gerenciamento baseado em políticas (STRASSNER, 2003), gerenciamento por exceção (LABARRE, 1991), gerenciamento por delegação (GOLDSZMIDT e YEMINI, 1995) e gerenciamento com a utilização de agentes móveis (BIESZCZAD, PAGUREK e WHITE, 1998).

O gerenciamento baseado em políticas (STRASSNER, 2003) foi pensado como uma forma de separar o como expressar aquilo que se deseja que ocorra na rede do como configurar os dispositivos para que este comportamento realmente ocorra. Neste modelo de gerenciamento, o gerente define políticas de alto nível para a rede e o sistema de gerenciamento se encarrega de converter estas políticas nas configurações específicas para cada tipo de dispositivo que deve ser configurado. Este modelo de gerenciamento torna o gerenciamento mais simples, pois permite que o gerente tenha uma visão única da rede, independente do tipo dos dispositivos presentes nela.

O gerenciamento por exceção (LABARRE, 1991) foi desenvolvido para tratar situações em que o volume de informações de gerenciamento que deveriam ser repassadas a uma estação gerente é muito grande. Neste modelo de gerenciamento, ao invés de serem repassadas informações cruas a respeito do estado da rede, apenas situações não habituais (chamadas de exceções) são reportadas, evitando sobrecarregar o sistema de gerenciamento. Ao perceber estas situações não habituais, o sistema de gerenciamento pode tomar alguma atitude.

O gerenciamento por delegação (GOLDSZMIDT e YEMINI, 1995) foi proposto ao se observar que, freqüentemente, muitas informações de gerenciamento eram enviadas a estações gerente pelo simples fato de que apenas nestas estações havia a inteligência para se analisar os dados e tomar as atitudes apropriadas. A proposta deste modelo de gerenciamento é que, ao invés de se levar todas as informações à estação gerente, a tomada de decisão seja delegada a algum componente próximo aos dados, evitando que os mesmos tenham de ser transmitidos e aproveitando a capacidade de processamento local. A tomada de decisão é delegada através do repasse, pela estação gerente, para o elemento gerenciado, de um programa especial capaz de analisar os dados onde eles são gerados e tomar a atitude correta.

O gerenciamento com a utilização de agentes móveis (BIESZCZAD, PAGUREK e WHITE, 1998) apresenta um senso semelhante ao do gerenciamento por delegação. Os agentes móveis, por se moverem através dos nós da rede, podem trabalhar mais próximos aos dados, evitando que os mesmos tenham de ser transmitidos. Pode haver cooperação entre agentes móveis e, ao migrar de um nó a outro, o agente móvel leva consigo conhecimento a respeito do estado do nó anterior. Este conhecimento pode ser

utilizado para a identificação de um estado global da rede. Os agentes móveis também podem ser utilizados para a configuração de um conjunto de dispositivos em seqüência, migrando de um dispositivo para o outro. Um exemplo desta utilização seria configurar QoS na rota por onde um fluxo passa (MEER et al., 2000).

Em função do grande interesse no gerenciamento baseado em políticas, o IETF definiu uma arquitetura de gerenciamento hierárquica padrão (YAVATKAR, PENDARAKIS e GUERIN, 2000) para este modelo de gerenciamento. A arquitetura foi definida focando basicamente na possibilidade de se realizar controle de admissão baseado em políticas. Há quatro componentes definidos para esta arquitetura: estação gerente, repositório de políticas, PDP (*Policy Decision Point*) e PEP (*Policy Enforcement Point*). A arquitetura do IETF está ilustrada na Figura 2.2.

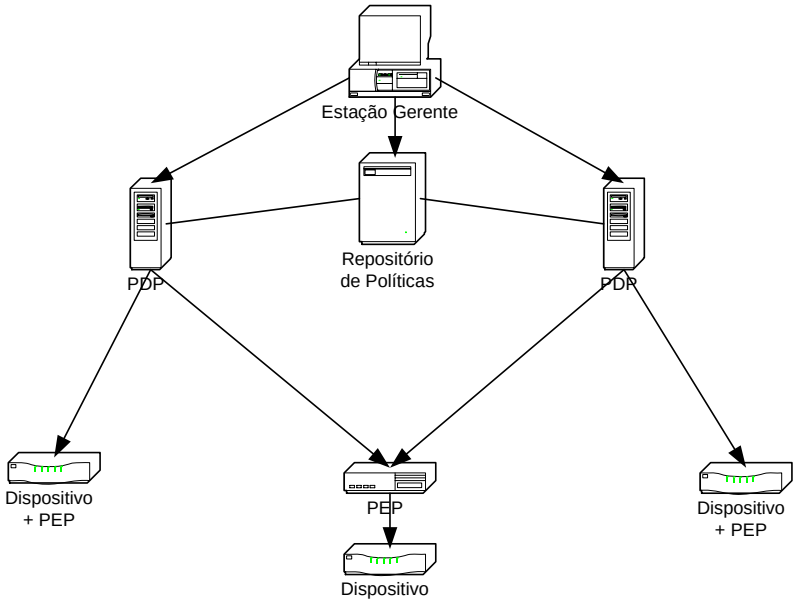


Figura 2.2: Arquitetura de gerenciamento baseado em políticas do IETF

A função da estação gerente é permitir que o gerente da rede defina políticas da alto nível sobre o comportamento que a rede deve apresentar. Estas políticas ficam, então, armazenadas em um repositório de políticas para que sejam ativadas em um momento oportuno.

Quando o gerente da rede deseja que uma política passe a funcionar em sua rede, ele ativa esta política repassando a definição da mesma a um ou mais PDPs. Pode-se tanto enviar a política a um PDP, quanto instruí-lo para que baixe a mesma a partir do repositório de políticas. O PDP possui a função de identificar quais dispositivos sob sua responsabilidade devem ser afetados pela política e tomar decisões de modo a garantir que a política seja efetiva. Para aplicar as políticas de rede, o PDP comunica-se com o PEP, que é uma entidade próxima (ou interna) aos dispositivos sendo gerenciados e que possui controle sobre os mesmos. O protocolo COPS (DURHAM et al., 2000) foi definido pelo IETF para a comunicação entre PDP e PEP.

A arquitetura do IETF pode ser utilizada em dois modos de operação: *outsourcing* e *provisioning*. Pela definição da arquitetura (YAVATKAR, PENDARAKIS e GUERIN, 2000), fica claro que a idéia inicial foi trabalhar no modo *outsourcing*. Neste modo, o PEP solicita uma decisão ao PDP a respeito de algum evento importante que esteja ocorrendo na rede, e, com base na resposta do PDP, o PEP efetua a ação que

corresponde à política avaliada pelo PDP. A primeira extensão (HERZOG et al., 2000) ao protocolo COPS definiu a sua utilização com o protocolo RSVP (BRADEN et al., 1997). Conforme esta extensão, para cada solicitação de recursos feita à rede via RSVP o PEP solicitava ao PDP uma decisão. Com base na política da rede, o PDP instruía o PEP sobre se a solicitação deveria ou não ser aceita.

O modo *provisioning* (CHAN et al., 2001) foi proposto posteriormente no protocolo COPS para permitir que configurações fossem provisionadas diretamente nos dispositivos, ao invés de fazer os mesmos delegarem (via *outsourcing*) a decisão sobre os eventos da rede. No modo *provisioning*, o PDP instrui o PEP sobre qual a configuração que deve ser efetuada no dispositivo para que ele atenda à política da rede e o PEP simplesmente efetua a configuração.

O modo *provisioning* surgiu em função da falta de escalabilidade apresentada pelo RSVP em conjunto com o modelo *outsourcing*. O artigo (PONNAPPAN et al., 2002) faz uma comparação bastante interessante a respeito da eficiência dos dois modelos, embora não considere especificamente a questão da escalabilidade. Apesar de a arquitetura do IETF ser uma referência em gerenciamento baseado em políticas, o protocolo COPS não tem sido muito utilizado em função de poder ser substituído facilmente pelo SNMP em uma operação no modo *provisioning*.

2.2 Visão Geral de IEEE802.16

O padrão IEEE802.16 define uma tecnologia para acesso sem fio em banda larga que deve tornar-se bastante popular nos próximos anos. Com a possibilidade de disponibilização a um custo menor do que outras tecnologias e com abrangência muito superior ao das tecnologias cabeadas (visto que não requer o enterramento de cabos), esta tecnologia tem o potencial de tornar o acesso à Internet realmente universal.

Devido a suas características técnicas, redes IEEE802.16 permitem a instalação de verdadeiras MANs sem fio. O alcance real atingido por uma rede IEEE802.16, operando em freqüências na faixa de 2,5Ghz, pode ficar entre 18Km e 20Km havendo linha de visada e entre 9Km e 10Km sem depender de visibilidade direta (PRADO, 2006). Este amplo alcance é crucial em áreas de difícil penetração. Em condições favoráveis, estima-se que a taxa de transmissão possa atingir 70Mbps.

O padrão IEEE802.16 é promovido majoritariamente pelo consórcio WiMAX (WIMAX FORUM, s.d.), que possui como integrantes empresas como Intel, Nokia, Motorola e At&T. Implementações físicas deste padrão estão sendo testadas já há algum tempo em diversas partes do mundo, sendo comum referir-se ao mesmo simplesmente por WiMAX – da mesma forma que o padrão IEEE802.11 acabou também conhecido por WiFi.

De fato, a denominação IEEE802.16 refere-se a um conjunto de padrões do IEEE. O primeiro padrão publicado foi aprovado em dezembro de 2001 (IEEE, 2002), mas apenas em 2004 foi aprovado um padrão para operação fixo alinhado com definições da ETSI (WIKIPEDIA, s.d.). Este padrão ficou conhecido como IEEE802.16d ou IEEE802.16-2004 (IEEE, 2004).

O padrão IEEE802.16d define duas entidades participantes do enlace sem fio: estações base (*base stations*) e estações cliente (*subscriber stations*). As estações base

provêem conectividade às estações clientes e são mantidas fixas em torres espalhadas de forma a otimizar a área de cobertura da rede. Tipicamente as estações base são conectadas entre si por uma rede denominada *backhaul*, que, estando conectada à Internet, permitiria que as estações cliente também obtivessem acesso a esta rede global.

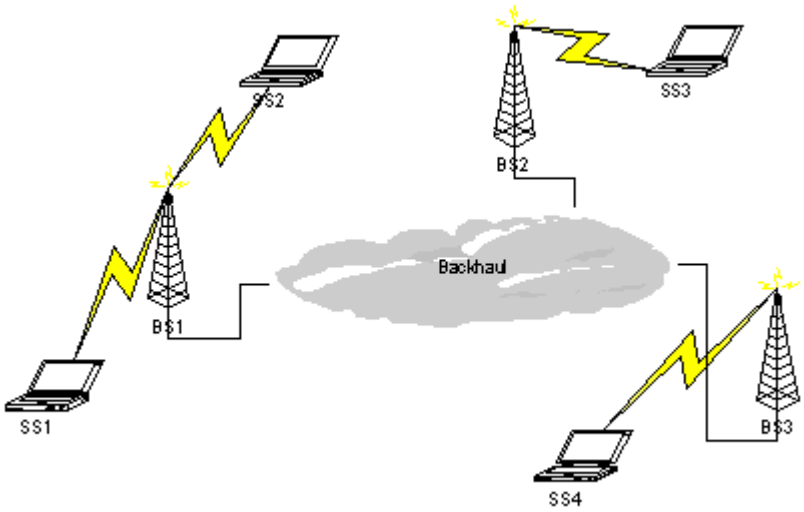


Figura 2.3: Rede IEEE802.16 com estações base (BS) e estações cliente (SS)

Embora seja referenciado como padrão fixo, o padrão IEEE802.16d permite que as estações cliente apresentem alguma mobilidade em baixas velocidades. A funcionalidade que falta a este padrão e que justifica sua denominação como fixo é a possibilidade de realização de *handover*, que permite a uma estação cliente trocar de estação base sem perder a conectividade. Neste caso, costuma denominar-se a estação cliente como estação móvel.

A funcionalidade de *handover* somente foi incluída no conjunto de padrões IEEE802.16 no início de 2006, com a publicação do padrão IEEE802.16e (IEEE, 2006), que automaticamente recebeu a denominação de padrão IEEE802.16 móvel. Embora o padrão IEEE802.16e seja, de fato, uma emenda ao padrão IEEE802.16d, ele não é compatível com seu antecessor por alterações na camada física de transmissão (PRADO, 2007; GINEVAN, 2008; JONES, 2005; LYMAN, 2005). O cenário mais provável para futuro é que apenas o padrão móvel seja instalado em larga escala. Dada esta visão, fica clara a escolha desta dissertação em trabalhar sobre o padrão móvel.

Quatro alternativas para a sinalização de nível físico foram estabelecidas pelo padrão. Há duas alternativas *Single Carrier*, sendo uma para freqüências de 10Ghz a 66Ghz (WirelessMAN-SC) e outra para freqüências abaixo de 11Ghz (WirelessMAN-SCa), uma alternativa de utilização de OFDM (WirelessMAN-OFDM) e outra alternativa de utilização de OFDMA (WirelessMAN-OFDMA), ambas estas últimas operando em freqüências abaixo de 11Ghz. Uma quinta alternativa foi construída com base nas três últimas adicionando-se a capacidade de seleção dinâmica de freqüência. Conforme o padrão IEEE802.16e, apenas as alternativas de freqüência inferior a 11Ghz sem seleção dinâmica de freqüência podem ser utilizadas fornecendo mobilidade.

Com relação ao protocolo de enlace definido para IEEE802.16, foram definidos dois modos de operação: PMP (*Point-to-Multipoint*) e *Mesh*. No modo de operação PMP, todo o acesso à interface aérea é controlado pela estação base, e toda a comunicação entre nodos da rede IEEE802.16 deve ser feito via estação base – somente podem

ocorrer transmissões de estação base para estação cliente (*downlink*) ou vice-versa (*uplink*). Já no modo *Mesh*, as transmissões podem ocorrer entre quaisquer nodos vizinhos. Neste modo, os nós são organizados logicamente na forma de árvores em que uma estação conectada ao *backhaul* é denominada estação base da rede *Mesh* e constitui a raiz de uma árvore. Em cada árvore, o controle do acesso ao meio físico é feito tanto de forma centralizada, através de um cronograma distribuído pela estação base de rede *Mesh*, como distribuída, caso em que nós não raiz negociam o acesso nos momentos em que o cronograma permite. A mobilidade especificada no padrão IEEE802.16e foi definida apenas para o modo PMP.

O protocolo de enlace IEEE802.16 é totalmente orientado a conexão, sendo que cada conexão possui um CID (*Connection Identifier*) de 16 bits que a identifica. Há dois tipos de conexões nas redes IEEE802.16: conexões de gerenciamento e conexões de transporte. As conexões de gerenciamento são utilizadas para a transmissão de mensagens de gerenciamento e controle da rede enquanto as conexões de transporte são utilizadas para transmissão de dados de usuário. Cabe notar que não há transmissão de dados de usuário em conexões de gerenciamento e também não há transmissão de mensagens de gerenciamento em conexões de transporte. Alguns CIDs (ou faixas de CIDs) foram pré-alocados para finalidades específicas, como o CID de gerenciamento em broadcast (0xFFFF) e a faixa de CIDs para conexões de transporte multicast em *downlink* (0xFEA0-0xFEFE).

Há três conexões de gerenciamento principais que foram definidas pelo padrão e possuem seus CIDs determinados no momento em que uma estação entra na rede: a conexão básica, a conexão de gerenciamento primária e a conexão de gerenciamento secundária. As duas primeiras são obrigatórias para todas as estações clientes. A diferença entre elas é que a conexão básica é utilizada para mensagens mais curtas e urgentes, enquanto a conexão de gerenciamento primária é utilizada para mensagens mais longas e mais tolerantes a atraso. A conexão de gerenciamento secundária é criada apenas para estações cliente gerenciáveis e sua função é a transmissão de protocolos de gerenciamento padronizados de alto nível para as estações (SNMP, DHCP, TFTP).

Em termos de estrutura, o nível de enlace do padrão IEEE802.16 foi dividido em três sub-camadas: *Convergence Sublayer* (CS), *Common Part Sublayer* (CPS) e *Security Sublayer*. A idéia é que a camada CS adapte protocolos de níveis superiores à transmissão via rede IEEE802.16. Exemplos de implementação para a camada CS são a *Packet CS* – que prevê o encapsulamento para pacotes IP, quadros IEEE802.3 (Ethernet), quadros IEEE802.1Q (VLAN) – e a ATM CS – que prevê o encapsulamento para células ATM. A camada CPS, por outro lado, provê a parte comum do protocolo de enlace IEEE802.16, como o gerenciamento de conexões e a definição de mensagens de controle e cabeçalhos padrão para os quadros IEEE802.16. A *Security Sublayer* é responsável pela segurança da rede, provendo mecanismos de autenticação e confidencialidade. A Figura 2.4 ilustra a estrutura de camadas do padrão IEEE802.16 através de seu modelo de referência.

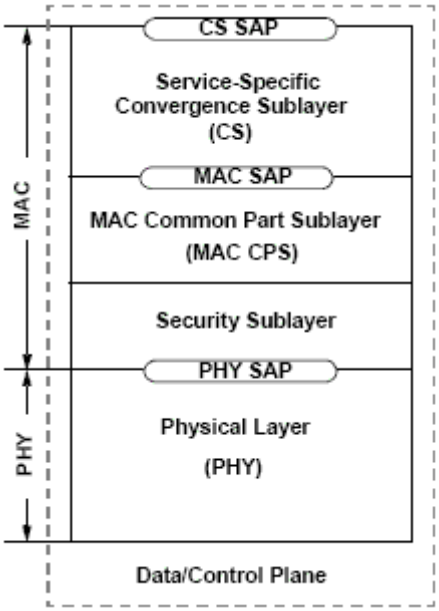


Figura 2.4: Modelo de referência para os planos de dados e controle (IEEE, 2004)

Entrando um pouco mais na questão de QoS, toda a definição de parâmetros de QoS é feita com base no conceito de *Service Flow*. Toda conexão de transporte em uma rede IEEE802.16 possui um *Service Flow* associado que especifica os parâmetros de QoS para tal conexão. De fato, o melhor seria dizer que é o *Service Flow* que possui uma conexão de transporte, visto que ele pode existir independentemente da existência desta e, quando se torna ativo, um CID é criado para representar a conexão associada ao mesmo. Os *Service Flows* são definidos como um fluxo unidirecional de pacotes para os quais é provida determinada QoS. Eles são identificados univocamente no contexto entre uma estação base e uma estação cliente através de um identificador único de 32 bits: o SFID (*Service Flow Identifier*). O modelo de dados que indica a relação entre uma conexão de transporte (*Transport Connection*) e um *Service Flow* pode ser visto na Figura 2.5.

A Figura 2.5 também apresenta outros elementos relacionados ao enlace das redes IEEE802.16. Cada pacote a ser transmitido na rede (*MAC PDU*) é classificado em algum *Service Flow* através de um classificador (*Classifier Rule*). Este classificador pode ou não ter associado a si uma regra de PHS (*PHS Rule*), que indica a supressão de campos de cabeçalho repetitivos (uma medida para aumento de eficiência de transmissão). Múltiplos classificadores podem classificar pacotes distintos para o mesmo *Service Flow*. Por fim, um *Service Flow* pode ou não possuir uma conexão associada (dependendo se está ou não ativo) e pode ou não estar vinculado a uma classe de serviço (*Service Class*). Uma classe de serviço mantém um conjunto de parâmetros de QoS comuns, que podem ser reutilizados em diversos *Service Flows*. Cabe notar que um *Service Flow* pode personalizar parâmetros de QoS mesmo quando está vinculado a uma classe de serviço.

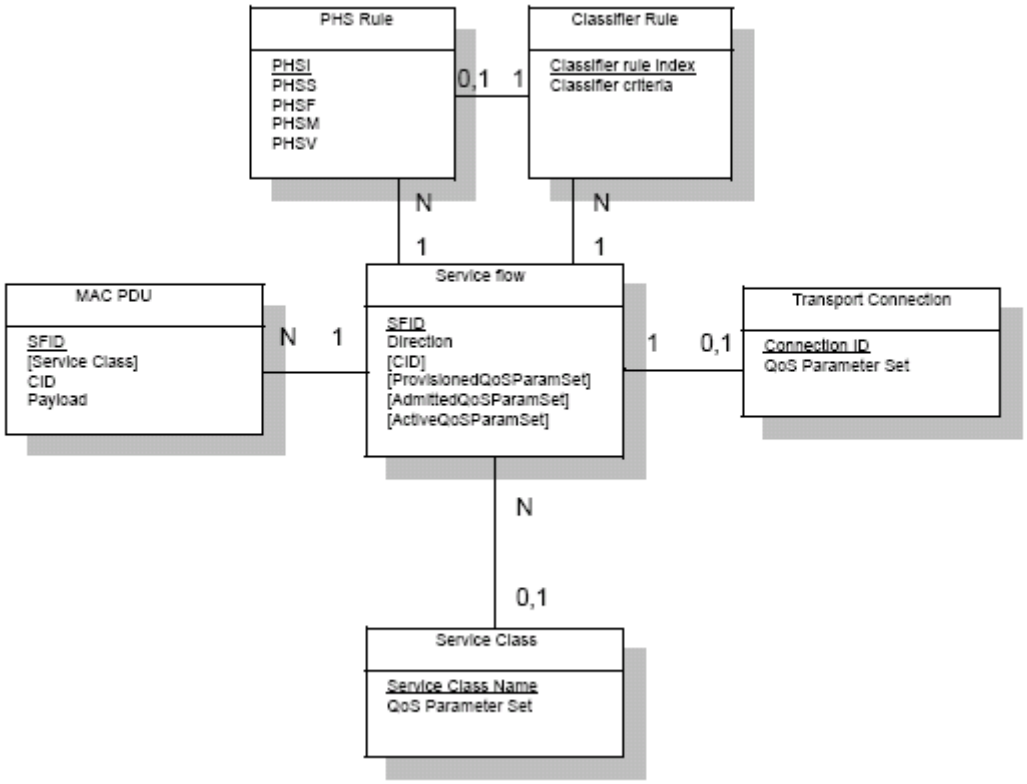


Figura 2.5: Modelo de dados para *Service Flows* (IEEE, 2006)

Ao ser criado, cada *Service Flow* especifica o tipo de camada CS que deve ser utilizada para si. Desta forma, um *Service Flow* já especifica se irá transportar células ATM, pacotes IP ou algum outro conteúdo. Naturalmente os classificadores a serem utilizados para este *Service Flow* deverão utilizar critérios condizentes com o tipo de camada CS especificada para o *Service Flow*.

Um *Service Flow* pode estar em um de três estados: provisionado, admitido ou ativo. Um *Service Flow* se encontra em estado provisionado quando ele existe, mas por algum motivo sua ativação está sendo postergada. Neste estado, o *Service Flow* não chega a possuir uma conexão de transporte associada, visto que não há tráfego sendo enviado através do mesmo. Um *Service Flow* se encontra em estado admitido quando a rede aceitou suprir seus requisitos de QoS, mas a reserva dos recursos ainda não foi confirmada. A idéia deste estado é que se possa solicitar recursos fim-a-fim e, apenas caso todos os recursos necessários estejam disponíveis, efetuar de forma definitiva a reserva de recursos. Por fim, um *Service Flow* em estado ativo possui todos os seus requisitos de QoS atendidos e se encontra em atividade na rede – isto é, há previsão para o envio de tráfego referente a este fluxo.

O padrão IEEE802.16 prevê ainda a possibilidade dos parâmetros de QoS para os estados provisionado, admitido e ativo serem diferentes. Os parâmetros do estado admitido seriam menos exigentes que os do estado provisionado, por exemplo, se a admissão se deu de forma degradada. Os parâmetros do estado ativo seriam menos exigentes que os do estado admitido, por exemplo, em uma situação na qual a aplicação deseje poupar recursos de rede ao ver que não necessitará consumir toda sua reserva. Este seria o caso de uma chamada VoIP que é colocada em *hold*, cortando a transmissão

de voz. A existência dos três conjuntos de parâmetros de QoS pode ser percebida na Figura 2.5.

Além dos parâmetros de QoS, *Service Flows* em *uplink* são associados a um tipo de serviço de escalonamento. Há cinco serviços de escalonamento distintos definidos pelo padrão IEEE802.16: UGS (*Unsolicited Grant Service*), rtPS (*Real-time Polling Service*), extended rtPS, nrtPS (*Non-real-time Polling Service*) e BE (*Best Effort*).

O serviço UGS foi desenvolvido para suportar tráfego de tempo real consistindo de pacotes de tamanho fixo emitidos em intervalos regulares (tipo troncos E1 ou T1). Neste serviço, a estação base concede tempo de transmissão para a estação cliente em intervalos regulares, de forma que os dados possam ser transmitidos.

O serviço rtPS foi desenvolvido para suportar tráfego de tempo real que emite pacotes de tamanhos variados em intervalos periódicos. Neste serviço, a estação base concede tempo de transmissão para a estação cliente em intervalos regulares visando a que a mesma informe suas necessidades de transmissão. A necessidade de transmissão é informada através de uma mensagem específica do nível de enlace que informa à estação base qual a ocupação da fila de transmissão para o *Service Flow* (em termos de *bytes*). Esta informação pode se dar de forma absoluta ou incremental em relação a alguma notificação feita anteriormente. Tendo conhecimento das necessidades de transmissão para cada *Service Flow*, a estação base pode alocar apropriadamente o tempo de transmissão para a estação cliente em questão.

O serviço extended rtPS foi desenvolvido para unir características tanto do UGS quanto do rtPS. Neste caso, há concessões de tempo de transmissão para dados em intervalos periódicos, mas suporta-se pacotes de tamanho variado através do mecanismo de indicação da ocupação da fila de transmissão. Este serviço estava ausente no padrão IEEE802.16d e foi incluído no padrão IEEE802.16e em função de estudos que comprovaram a necessidade de um mecanismo deste tipo (HONG e KWON, 2006). Em particular, considera-se que este mecanismo pode ser útil em VoIP com supressão de silêncio.

O serviço nrtPS foi desenvolvido para suportar aplicações que exijam alguma reserva de banda, mas não possuam requisitos estritos de tempo. Este seria o caso, por exemplo, de uma transferência de arquivo em que se desejasse reservar banda. A diferença para o rtPS é que, neste caso, a estação base não assume nenhum compromisso com o atraso das mensagens de dados do *Service Flow*. Neste tipo de serviço também ocorrem concessões periódicas de tempo para que as necessidades de transmissão da estação cliente sejam informadas.

O serviço BE, por fim, é o serviço padrão sem QoS. Naturalmente *Service Flows* que utilizem este serviço de escalonamento ficarão sujeitos a problemas em situações nas quais a rede apresente congestionamento.

No protocolo de enlace IEEE802.16, há três grupos de mensagens que permitem gerenciar *Service Flows*: DSA (*Dynamic Service Addition*), DSC (*Dynamic Service Change*) e DSD (*Dynamic Service Deletion*). As mensagens DSA são utilizadas para criação de novos *Service Flows*. As mensagens DSC são utilizadas para modificação de *Service Flows*. Um exemplo de modificação seria alterar os requisitos de QoS de um *Service Flow* ou alterar seu estado de admitido para ativo. As mensagens DSD são utilizadas para remover *Service Flows*, de modo que eles não existam mais na rede.

Tanto a estação base quanto as estações cliente podem emitir estes tipos de mensagem, mas é principalmente a estação base que determina o sucesso ou não da operação em função da disponibilidade de recursos e de outras considerações.

Além do modo normal de operação na rede, o padrão IEEE802.16e introduziu novos modos de operação que permitem economia de energia a estações móveis: modo *sleep* e modo *idle*. Uma estação que entre em modo *sleep* passa a apresentar períodos de indisponibilidade previamente combinados com sua estação base. Os períodos de indisponibilidade são estrategicamente selecionados para não interferir com a transmissão dos dados de acordo com os requisitos de QoS. Uma estação que entre em modo *idle* passa a não transmitir na rede. Ela fica apenas escutando os dados em momentos oportunos e, periodicamente, atualiza sua localização com relação ao conjunto de estações base de forma a sinalizar que ainda está ativa. Nos períodos em que não precisa escutar, a estação móvel pode, por exemplo, averiguar qual a estação base que emite o melhor sinal ou simplesmente desligar-se.

3 DESAFIOS PARA GERENCIAMENTO DE QOS EM IEEE802.16

O objetivo deste capítulo é trazer à tona quais são as peculiaridades de uma rede IEEE802.16 que tornam esta tecnologia diferente das demais tecnologias de rede do ponto de vista de gerenciamento. Também são apresentados neste capítulo requisitos a serem considerados em um bom sistema de gerenciamento de QoS para este tipo de rede. A análise feita neste capítulo norteou a decisão sobre o modelo e forma de gerenciamento que serão propostos nos capítulos posteriores.

3.1 Características da rede

A grande diferença da tecnologia IEEE802.16 frente a outras tecnologias de rede é o fato de ela se propor como uma tecnologia de acesso móvel para MANs, permitindo conectar à infra-estrutura de rede fixa (tipicamente a Internet) estações cliente distribuídas em uma vasta área de cobertura, sem a necessidade de utilização de cabos. Desta forma, ela simultaneamente agrega um conjunto de dificuldades inerentes ao gerenciamento de redes sem fio com outras tipicamente encontradas em redes celulares (como gerenciamento de situações de *handover*).

Uma rede IEEE802.16, de fato, pode ser considerada uma rede celular de dados. As estações base (*base stations*) permanecem fixas em pontos estratégicos (maximizando a cobertura) enquanto as estações cliente (*subscriber stations*) se deslocam, possivelmente migrando entre as estações base. Diferentemente de uma rede telefônica celular, no entanto, uma rede IEEE802.16 admite uma gama mais variada de serviços. Esta gama vai desde serviços que exigem requisitos estritos de QoS (como VoIP ou a emulação de um canal E1) até serviços mais tolerantes (como navegação por páginas Web). A tecnologia IEEE802.16 foi planejada de modo a ser uma verdadeira rede de convergência de serviços – voz, dados, multimídia, etc.

Também é possível encontrar, em uma rede IEEE802.16, estações que, embora utilizem o meio aéreo para comunicação, permanecem fixas em um determinado local e não possuem grandes limitações de recursos. Uma situação na qual isto ocorre seria o caso de alguém acessar a Internet através de uma estação cliente IEEE802.16 fixa em sua residência. O fato de a estação cliente permanecer fixa permite ligá-la a rede elétrica e assegurar que não haverão maiores limitações de energia. Se a estação cliente for, em verdade, uma placa de extensão em um computador da casa utilizando software e antena apropriados, também pode-se pensar em uma situação sem grandes restrições de processamento e armazenamento.

Em comparação com outras redes sem fio, uma rede IEEE802.16 tende a ser mais afetada por condições ambientais devido à sua abrangência, o que requer atenção especial a condições de degradação de serviço. Também cabe notar que o protocolo de enlace definido para este tipo de rede possui um bom suporte a QoS no modo PMP. Se bem gerenciado, este suporte a QoS pode trazer reais benefícios aos usuários desta tecnologia (particularmente em situações de congestionamento, quando a priorização de fluxos se faz importante). Outras redes sem fio, como IEEE802.11, não possuem ou possuem um suporte fraco a QoS, deixando a desejar nas situações em que este seria importante.

Ainda em comparação com outras redes sem fio, as redes IEEE802.16 foram projetadas considerando-se a existência de uma infra-estrutura fixa composta pelas estações base e pela rede que as interconecta (mesmo que esta seja uma rede sem fio). Tipicamente, infra-estruturas fixas são menos suscetíveis a restrições de energia, processamento e armazenamento. Esta infra-estrutura fixa pode ser utilizada a favor no momento de gerenciar este tipo de rede, garantindo uma maior disponibilidade de recursos ao sistema de gerenciamento e tornando-o menos suscetível a falhas. Outras redes sem fio, mesmo podendo ser interconectadas a uma infra-estrutura fixa, não costumam considerar esta infra-estrutura fixa como sendo parte da rede, e podem existir independentemente dela.

Devido à variabilidade da capacidade de transmissão da rede (causada por variações de condições ambientais, movimentação de estações cliente, entrada de novas estações), o processo de CAC (Connection Admission Control) final deve obrigatoriamente ser feito pelas estações base. Isto é, para uma dada requisição, o sistema de gerenciamento é capaz de validar os requisitos de QoS com as permissões do requisitante. Entretanto, ele não é capaz de saber se a rede será capaz de prover o serviço naquele momento, visto que uma variação da condição de rede é capaz de inviabilizar algum requisito de QoS. Apenas as estações base possuem o conhecimento em tempo real para, no momento da requisição, identificar se um dado fluxo pode ou não ser admitido com os requisitos de QoS que o mesmo exige. Cabe notar que algum conhecimento sobre as limitações da rede ainda pode ser configurado no sistema de gerenciamento, permitindo a identificação de casos esdrúxulos sem a necessidade de realização de CAC por parte das estações base.

As características principais que diferenciam redes IEEE802.16 de outros modelos de redes e que devem ser consideradas em seu gerenciamento foram sumarizadas abaixo. Algumas características são particularmente relevantes no contexto de gerenciamento de QoS. Nem todas estas características foram abordadas nos parágrafos anteriores por serem de simples compreensão.

- As estações cliente em uma rede IEEE802.16 podem ser bastante heterogêneas, sendo algumas fixas e outras móveis;
- As estações cliente móveis podem migrar entre estações base;
- Algumas estações cliente podem apresentar períodos de indisponibilidade (modos *sleep* e *idle*);
- Estações cliente podem entrar e sair da rede a qualquer momento;
- Algumas estações cliente podem apresentar restrições de energia, processamento e/ou armazenamento;

- A taxa de transmissão pode variar em função de movimentações ou condições ambientais;
- A rede possui uma infra-estrutura fixa que lhe dá suporte;
- A rede deve ser capaz de prover serviço a uma gama variada de aplicações;
- O suporte a QoS do protocolo de enlace deve ser gerenciado para otimizar a utilização de recursos;
- Apenas as estações base podem executar CAC do ponto de vista de recursos.

3.2 Requisitos para gerenciamento de QoS

O levantamento de requisitos é uma etapa importante para a definição da arquitetura a ser utilizada. Devem ficar claro nos requisitos tanto as linhas gerais a serem seguidas para a obtenção de um bom sistema de gerenciamento quanto considerações sobre aspectos importantes presentes apenas no gerenciamento de QoS em redes IEEE802.16.

Em linhas gerais, a arquitetura de gerenciamento a ser definida deverá ser eficiente na realização do gerenciamento. Isto é, o mecanismo de gerenciamento deve consumir o mínimo possível de recursos de rede. A rede existe basicamente para atender às aplicações que rodam sobre a mesma, sendo o tráfego de gerenciamento apenas uma sobrecarga inevitável que deve competir o mínimo possível com o tráfego das aplicações.

Deve ser possível realizar o gerenciamento de uma forma simples e intuitiva. Se um sistema de gerenciamento for muito complicado de compreender, a menos que ele seja a única forma de realizar o gerenciamento, muitos usuários talvez acabem por buscar alternativas que lhes pareçam mais simples, mesmo que estas alternativas sejam, no fundo, mais onerosas. Em outros casos, usuários podem deixar de utilizar algum recurso que a rede seria capaz de lhes proporcionar por não compreenderem bem o sistema de gerenciamento.

Em redes IEEE802.16, é importante que o sistema de gerenciamento apresente baixo custo de processamento e armazenamento para as estações cliente, visto que algumas destas estações podem possuir recursos restritos. Também convém que o sistema de gerenciamento não seja dependente das estações cliente. A existência de períodos de indisponibilidade para algumas estações cliente ou mesmo a saída de alguma delas da rede não pode comprometer a capacidade de se gerenciar a rede.

O sistema de gerenciamento deve estar preparado para o fato de as estações base realizarem CAC nos fluxos que já foram admitidos a nível de gerência. Em função dos recursos disponíveis, o resultado deste CAC poderá ser a negação total do fluxo que deveria ter sido admitido na rede ou a sua admissão final em modo degradado. Este requisito se impõe também, mas de modo inverso, sobre as estações base. As estações base deverão ter, de fato, a capacidade de realização de CAC do ponto de vista de recursos disponíveis. Dado que o sistema de gerenciamento já tenha autorizado a liberação dos recursos para um dado fluxo, as estações base deverão ser capazes de admitir ou não este fluxo na rede dependendo da existência real dos recursos previstos.

O sistema de gerenciamento deverá estar preparado para oferecer tratamento diferenciado a estações clientes fixas e móveis, possibilitando adequar melhor o serviço da rede às necessidades de cada usuário. O sistema também deverá estar preparado para

tratar adequadamente as migrações de estações móveis entre estações base, de modo que a rede sempre provenha o serviço adequado conforme a localização do usuário.

É importante que o sistema de gerenciamento possua a capacidade de auto-adaptar-se às variações de condição da rede. Seria altamente ineficiente uma pessoa ter de tomar atitudes sempre que houver degradação na QoS dos fluxos da rede. O ideal é que o sistema permita que se especifique o que deve ocorrer na rede caso haja uma situação de degradação de QoS.

4 SOLUÇÃO PROPOSTA

Identificados os desafios a serem superados para o gerenciamento de QoS em redes IEEE802.16, o objetivo deste capítulo é o de apresentar uma solução para a realização deste gerenciamento.

4.1 Gerenciamento de QoS baseado em políticas

Após o estudo de diferentes alternativas para o gerenciamento de QoS em redes IEEE802.16, decidiu-se por realizar o gerenciamento baseado em políticas, visto que este apresenta diversas vantagens no contexto de gerenciamento de QoS. A decisão não significa que outras abordagens não sejam válidas, apenas se optou por aquela que pareceu mais apropriada após a análise inicial.

Gerenciamento baseado em políticas torna tudo mais simples ao permitir especificar o que se deseja em um nível de abstração maior. Um sistema de gerenciamento baseado em políticas pode ser construído de forma a não exigir que se faça apontamentos específicos dos dispositivos a serem configurados e quais as suas configurações. Esta flexibilidade permitiria tratar bem questões como a migração das estações móveis entre estações base – situação na qual as reservas de recursos devem ser desativadas a partir da estação base anterior e ativadas a partir da nova estação base.

Algo não usualmente pensado em gerenciamento por políticas, mas que pode ser facilmente adicionado é a questão de localização do usuário. Alguém que utilize IEEE802.16 em uma localização fixa (uma estação cliente fixa) poderia contratar um serviço de banda reservada apenas para a localização onde se encontra. Desta forma, tanto o provedor de IEEE802.16 poderia dimensionar melhor sua rede quanto o cliente poderia reduzir seus custos. Neste caso, se a estação cliente viesse a entrar na rede a partir de uma estação base fora da região contratada (ou migrasse para uma estação base fora da região contratada), o serviço de banda reservada não estaria disponível na nova região. Outra possibilidade é fornecer QoS diferenciada de acordo com a região da rede em que o usuário está acessando.

Ao permitir agrupamentos de conceitos, o gerenciamento baseado em políticas permite uma generalização simples de políticas. Agrupando-se usuários, pode-se definir políticas genéricas por grupo de usuários. Agrupando-se as estações base por região, pode-se definir políticas de QoS diferenciadas por região. Agrupando-se requisitos de QoS em classes de QoS, pode-se utilizar a mesma especificação de serviço em situações distintas. Agrupando-se horários de utilização, permite definir políticas diferenciadas para horários específicos como “horário comercial” ou “finais de semana”. Também se pode generalizar facilmente uma política para toda a rede.

Ao não depender da forma de configuração de cada dispositivo específico, gerenciamento baseado em políticas pode permitir que a QoS seja gerenciada da mesma forma tanto na parte IEEE802.16 da rede quanto na infra-estrutura fixa que lhe dá suporte. Isto permite desburocratizar o processo de gerenciamento, criando uma visão única da rede para o gerente. O sistema de gerenciamento se responsabiliza por garantir que as definições de QoS na parte IEEE802.16 da rede sejam consistentes com a política definida e com as definições de QoS no restante da rede, que pode utilizar a arquitetura DiffServ (BLAKE et al., 1998), por exemplo.

Políticas também permitem definir o que deve ocorrer em uma situação de degradação de QoS. Basta, neste caso, haver a possibilidade de se definir a política a ser seguida em situações de degradação. Idéias como a definição de políticas sobre políticas (PoP) (GRANVILLE et al., 2002) ou a auto-adaptação das políticas de rede (SAMAAN e KARMOUCH, 2003) são exemplos de soluções que tentam tratar a questão da degradação de QoS. Convém notar aqui que degradação de QoS se refere a uma situação na qual fluxos já admitidos na rede não estão recebendo a QoS que a rede havia concordado em lhes assegurar. O tratamento para casos em que a rede não suporta admitir um determinado fluxo para o qual a política define garantia de recursos será abordados mais adiante.

Para ilustrar o tratamento de situação de degradação com políticas, consideremos a idéia de PoPs (GRANVILLE et al., 2002). Na definição de PoPs, são criados conjuntos de políticas para serem utilizados em cada situação da rede. Um conjunto define políticas para operação normal, outro define políticas para uma situação especial A, e assim por diante. As políticas sobre políticas (PoPs) definem quando utilizar cada conjunto. Um exemplo de PoP seria este: se for detectada degradação no serviço A estando em operação normal, utilizar o conjunto de políticas P, que prioriza melhor este serviço. Neste caso, deve haver um script que defina como transformar o conjunto de políticas de operação normal no conjunto de políticas P. O artigo (GRANVILLE et al., 2002) sugere a utilização de uma máquina de estados para simplificar a visualização e definição das PoPs. A Figura 4.1 ilustra uma máquina de estados de PoPs em que P1, P2, P3 e P4 são conjuntos de políticas.

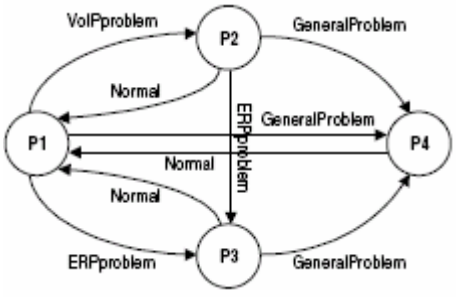


Figura 4.1: Máquina de estados de PoPs (GRANVILLE et al., 2002)

As idéias citadas anteriormente (PoPs e auto-adaptação das políticas de rede), entretanto, não representam uma solução perfeita para o tratamento de situações de degradação. Enquanto a definição e gerenciamento de PoPs pode tornar-se uma tarefa onerosa (por ser necessário especificar como transformar as políticas de uma situação de rede nas políticas das outras situações), a auto-adaptação das políticas da rede pode gerar soluções que não correspondam ao desejo da administração, visto que a rede estaria modificando as políticas por conta própria.

Diante de soluções imperfeitas, pensou-se em uma nova abordagem para tratamento de situações de degradação. Esta abordagem consiste em permitir a redefinição das classes de QoS e níveis mínimos de serviço à medida em que a situação de degradação se intensifica. Isto é, dado que a rede não tenha capacidade de assegurar os níveis de QoS acordados, no fundo o que resta é escolher quem deverá ser degradado primeiro (seja através da desativação ou modificação das políticas ou de parâmetros das mesmas). Este tipo de definição permitiria ao administrador definir quais fluxos (classes de QoS) deverão ser degradados primeiro de forma a manter os níveis de QoS nos demais. Em um caso extremo, uma dada classe poderia tornar-se equivalente à classe melhor esforço.

Com relação à alteração nos níveis mínimos de serviço, considere-se a situação em que a política de rede define por padrão que 10% dos recursos devem ser alocados aos fluxos da classe melhor esforço – obviamente o objetivo desta política é evitar que estes fluxos não recebam recurso algum em função de sucessivas reservas para os demais fluxos. Poderia-se definir um primeiro passo de degradação reduzindo este nível mínimo de serviço para 5% de forma a tentar manter a QoS dos serviços mais críticos durante o período de degradação.

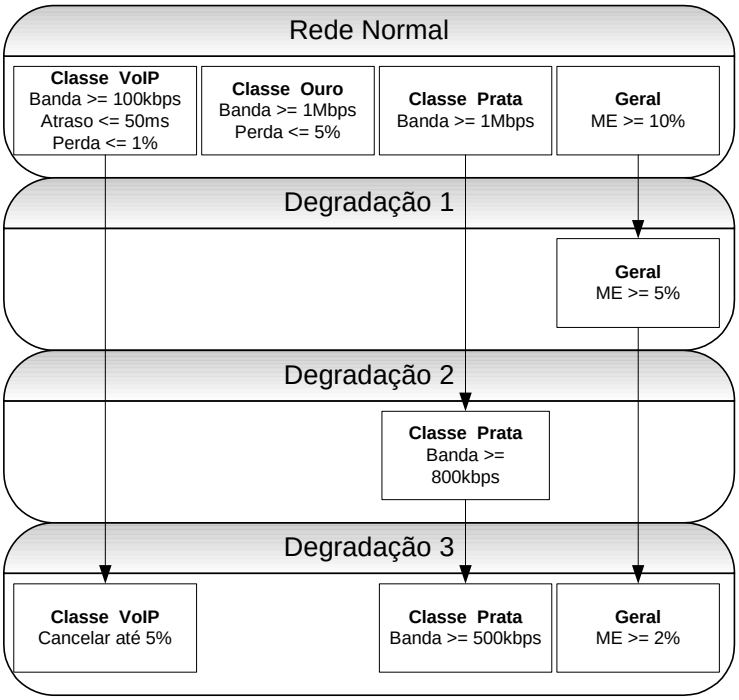


Figura 4.2: Níveis de degradação

A Figura 4.2 ilustra uma definição de níveis de degradação para quatro classes de QoS. A primeira caixa (Rede Normal) especifica os requisitos a serem atendidos para cada classe de QoS em uma situação de operação normal da rede. No primeiro nível de degradação (Degradação 1), é feita uma redefinição da quantidade mínima de recursos reservados para a classe melhor esforço (ME), garantindo que a rede dará manutenção as classes de QoS prioritárias. Caso esta alteração não seja suficiente para a rede cumprir seu contrato, o segundo nível de degradação (Degradação 2) especifica que a banda reservada para os fluxos da classe prata deverá passar a ser de 800kbps. Em uma última tentativa de contornar a situação, de acordo com esta política, no terceiro nível de degradação são reduzidas simultaneamente a reserva mínima dos fluxos melhor esforço

para 2% e a banda reservada para fluxos da classe prata para 500kbps. O terceiro nível de degradação também ilustra uma outra possibilidade, que seria autorizar a rede a cancelar totalmente a reserva de recursos para uma parcela de fluxos de uma dada classe. No caso da classe VoIP, o gerente da rede entendeu que reduzir a reserva de recursos a tornaria inútil, e optou por permitir o cancelamento da reserva de até 5% dos fluxos da classe VoIP (a conforme a necessidade da rede) mantendo a reserva dos demais inalterada. Os requisitos das classes VoIP e ouro foram mantidos inalterados durante todo o tempo em função de ser esta a política da rede.

Um campo de pesquisa em aberto na área de gerenciamento baseado em políticas é como tratar conflitos entre políticas (LUPU e SLOMAN, 1999; MOFFETT e SLOMAN, 1994). No contexto de gerenciamento de redes, um conflito entre políticas ocorre quando as políticas definidas para a rede não podem ser implementada com os recursos existentes ou são contraditórias. Estes conflitos podem ser detectados em dois momentos: durante a definição das políticas ou no momento de sua aplicação. O ideal é detectar todos os conflitos durante a definição das políticas, pois assim eles podem ser corrigidos em tempo de definição, não afetando a operação da rede. Entretanto, em algumas situações os conflitos somente podem ser detectados no momento de aplicação de uma política. Isto pode ocorrer tanto por uma ineficiência do sistema de gerenciamento em detectar alguma condição específica ou porque as políticas apenas se tornam contraditórias em situações específicas. Este último seria o caso em que um novo fluxo não poderia ser admitido na rede porque os recursos já foram reservados a outros fluxos – naturalmente imagina-se que todas as reservas estivessem autorizadas na política de rede, mas que não se esperava que tantas reservas fossem solicitadas à rede simultaneamente.

Por ser um problema em aberto, não há consenso sobre qual seria a melhor solução para resolver conflitos de políticas. No contexto de gerenciamento de QoS, entretanto, onde as políticas definidas costumam ser afirmativas (ex. reservar banda para o fluxo A), um esquema de prioridades entre as políticas pode ser uma maneira simples de tratar o problema. Deve-se, entretanto, assim como é feito para os próprios conflitos, distinguir entre duas prioridades: a prioridade de definição e a prioridade de aplicação das políticas.

A prioridade de definição das políticas serve para dirimir conflitos de definição entre as políticas. Imagine-se a situação em que o gerente da rede deseja indicar o fornecimento de QoS da classe Bronze ao grupo de usuários engenheiros. Imagine-se também que, na mesma situação, o gerente precise indicar que João, um engenheiro, deve receber QoS da classe Gold (obviamente uma exceção à regra anterior). Através da prioridade de definição das políticas, o administrador pode especificar que a política mais específica possui prioridade maior, garantindo que João receba QoS da classe Gold.

Por outro lado, a prioridade de aplicação das políticas serve para dirimir conflitos de aplicação entre as políticas. Neste caso, a situação seria como segue: João deve receber QoS da classe Gold, sendo um usuário prioritário; José deve receber QoS da classe Bronze; para garantir que, caso a rede não tenha condições de atender a ambos, João seja atendido prioritariamente, o gerente da rede poderia configurar a política de João com maior prioridade.

Percebe-se claramente que as prioridades apresentadas tratam conflitos existentes em dimensões diferentes, e, por este motivo, uma solução completa baseada em prioridades requer que ambas existam. Entretanto, é incomum encontrar menções a ambas as prioridades simultaneamente. Normalmente fala-se em prioridade referindo-se a apenas uma das duas (MOORE et al, 2001; MOORE, 2003).

Explorando um pouco mais o uso da prioridade de aplicação de políticas no contexto de IEEE802.16, considere-se, inicialmente, que todas as políticas definidas possuem a mesma prioridade. Desta forma, a reserva para um fluxo já admitido seria mantida frente a um novo fluxo que estivesse por entrar na rede. Esta parece ser a forma adequada de tratar esta questão. O uso de prioridades de aplicação também pode resolver questões pontuais como garantir prioridade a um usuário fixo local frente a um usuário móvel que migrou para uma dada região da rede – mesmo que o usuário móvel tivesse seus fluxos admitidos primeiro. Claro, todas as situações em que uma política não pode ser aplicada devem ser registradas e gerar os alertas administrativos pertinentes – o mesmo pode se dizer de situações de degradação de serviço.

4.2 Linguagem de definição de políticas

O objetivo desta seção não é criar uma nova linguagem de definição de políticas. Já existem muitas linguagens de definição de políticas para os mais variados propósitos (STONE, LUNDY e XIE, 2001). Esta seção apresentará uma reflexão sobre o que é interessante que esteja presente em uma linguagem de definição de políticas pensando no gerenciamento de QoS em redes IEEE802.16. Também serão tratadas algumas questões específicas do mapeamento de conceitos de linguagens de definição de políticas em conceitos de redes IEEE802.16.

Políticas normalmente são especificadas como pares condição ação. No contexto de QoS, o objetivo da condição deve ser o de identificar os fluxos (e mais especificamente os pacotes) para os quais a tal política deverá ser aplicada. Neste mesmo contexto, a ação de uma dada política deve ser capaz de indicar qual o tratamento a ser dado para os fluxos identificados pela condição.

A condição de uma política deve ser composta por testes sobre usuários, aplicações, dispositivos, interfaces de rede, horários, condição atual da rede, entre outros. Convém que se possa agrupar alguns destes itens e definir políticas para os grupos, simplificando o gerenciamento das políticas. Naturalmente as interfaces de rede devem estar vinculadas aos dispositivos aos quais elas pertencem, o que pode ser encarado como um grupo obrigatório.

No que se refere à condição da rede, ela pode ser considerada em dois momentos: avaliada apenas no momento de admissão da política ou avaliada constantemente. No caso de avaliação apenas no momento de admissão, a política não deixaria ser aplicada caso a condição da rede se alterasse posteriormente. No caso de avaliação constante, uma alteração na condição da rede que tornasse falsa a condição de uma política seria motivo para a política deixar de ser aplicada.

A ação de uma dada política é representada por um conjunto de requisitos de QoS. Estes requisitos estabelecem o tratamento a ser dado para os fluxos que satisfazem os critérios da política. Para facilitar o gerenciamento, requisitos de QoS podem ser agrupados em classes de QoS, e estas utilizadas em diversas políticas, eliminando a

necessidade de sempre citar-se individualmente cada requisito. A utilização de classes de QoS também permite o tratamento proposto por este trabalho para situações de degradação. Algumas classes podem possuir nomes padrão pré-definidos. Um exemplo é a classe melhor esforço, que não possui nenhum requisito de QoS.

O Quadro 4.1 apresenta exemplos de políticas simples que poderiam encontrar equivalentes em diversas linguagens de definição de políticas existentes. Na primeira política especificada, os usuários João e Pedro, em horário não comercial, podem utilizar até 500kbps cada um para aplicações P2P em interfaces wireless. A reserva é feita em separado para cada usuário por conta da palavra "an" que precede a palavra "user" na condição da política. Na segunda política definida no Quadro 4.1, é feita uma reserva de 1Mbps para tráfego do protocolo RTP em enlace *wireless*, sendo também especificados que o atraso não pode ser superior a 50ms e que a perda deve ser inferior a 0,2%.

Muitas definições utilizadas em uma política podem requerer maior detalhamento para que a aplicação da política possa ocorrer na prática. No exemplo do Quadro 4.1, BS1.1, BS2.1 e BS3.1 representam interfaces IEEE802.16 de estações base nas quais as reservas para os usuários João e Pedro poderiam ser configuradas, mas não consta no Quadro 4.1 um mapeamento destes nomes para as interfaces reais. Isto deverá ser feito em algum outro lugar.

Um detalhe importante com relação à indicação das interfaces na primeira política do Quadro 4.1 é que ela serve apenas para contextualizar a região da rede onde a reserva pode ocorrer. A reserva está vinculada propriamente aos usuários João e Pedro, devendo ocorrer apenas na estação base sendo utilizada pelo respectivo usuário. Em outras palavras, a determinação de onde a política será aplicada efetivamente pode ser dinâmica, não dependendo de indicação explícita na própria política. Se um destes usuários migrasse de estação base, o sistema de gerenciamento deveria migrar a reserva de mesmo junto.

```
#Definições prévias
user group gold includes João, Pedro
time group non_commercial includes Mon-Fri{18:00-09:00},
Sat{12:00-09:00}, Sun
application group P2P includes emule, kazaa
interface group wireless includes BS1.1, BS2.1, BS3.1
service class premium requires bandwidth>=500kbps
service class VoIP requires delay<=50ms, bandwidth>=1Mbps,
loss<=0.2%

#Políticas para a rede
if an user in gold and time in non_commercial and application in P2P and interface in
wireless then use class premium
if protocol = RTP and interface in wireless then use class VoIP
```

Quadro 4.1: Exemplo de política simples

A definição de onde exatamente uma política deve ser aplicada na rede pode ser dada de diversas formas. Pode-se citar explicitamente as máquinas onde a política deverá ser aplicada, pode-se vincular a política a algum elemento dinâmico (como um usuário) e apenas contextualizar em quais máquinas aquele elemento dinâmico poderia

usufruir a política (este foi o exemplo do parágrafo anterior), e pode-se ainda definir as pontas de um fluxo em termos de algum dos elementos anteriores, deixando a cargo do sistema de gerenciamento a determinação dos elementos intermediários. Este seria o caso, por exemplo, de definir-se uma política que reserve banda em *download* na rede IEEE802.16 para o usuário João para um fluxo *multicast* partindo de um dado servidor. Caso o João entrasse na rede IEEE802.16, o sistema de gerenciamento poderia identificar a rota a partir do servidor especificado até a estação cliente do João e fazer as reservas pertinentes nos servidores ao longo do caminho. Caso João mudasse de localização, as reservas poderiam segui-lo, visto que a reserva é para o João, e não para uma máquina específica.

Cada política definida na rede pode ter associadas uma prioridade de aplicação e uma prioridade de definição. A prioridade de aplicação define qual política tem preferência em caso de disputa por recursos, enquanto a prioridade de definição define qual política tem validade em caso de políticas contraditórias. Caso alguma prioridade não seja definida, um valor padrão de prioridade normal é assumido para a mesma. Para políticas de mesma prioridade de aplicação, assume-se que a primeira a ser aplicada na rede permaneça em caso de haver um conflito que impeça a aplicação simultânea de ambas. Caso ambas devam ser aplicadas ao mesmo tempo, fica a critério do sistema de gerenciamento determinar qual terá sucesso em sua aplicação, podendo ser uma escolha aleatória.

Políticas podem ser separadas entre políticas de obrigação (*obligation policies*) e políticas de autorização (*authorization policies*) (LUPU e SLOMAN, 1999). As políticas de obrigação definem algo que deve ocorrer (ex. reservar banda para uma videoconferência). As políticas de autorização definem algo que se permite que ocorra (ex. o usuário X tem direito de solicitar uma dada quantidade de recursos para sua aplicação). No caso de uma política de autorização, deve haver uma ação para que a política surta efeito (ex. o usuário X solicitar os recursos a que tem direito). A linguagem permitir ambos os tipos de definições a torna mais flexível.

Outras definições que podem ser permitidas por uma linguagem de definição de política são aspectos gerais de operação da rede, como níveis mínimos de serviço para determinadas classes de QoS. Isto permite, por exemplo, definir que a classe melhor esforço tenha no mínimo 10% dos recursos reservados a si de modo a não ser ignorada pela rede através da admissão sucessiva de fluxos com maior prioridade até o esgotamento dos recursos de rede. Políticas deste tipo não entram no formato padrão de condição/ação, mencionado anteriormente.

Dada a abordagem sugerida para tratar os casos de degradação de QoS, a linguagem de definição de políticas também deveria permitir a definição de níveis de degradação. A idéia é que se tenha um nível de degradação que se refere à operação normal de rede. Neste nível são definidos os requisitos padrão das classes de QoS e os aspectos gerais de operação da rede para operação normal. Tendo um nível correspondente à operação normal, os demais níveis são passos de degradação. Isto é, no momento em que a rede sofre uma degradação, as definições correspondentes ao primeiro passo de degradação são utilizadas. Se sofrer nova degradação, as definições do segundo passo de degradação são utilizadas e assim por diante até que se possa retornar ao nível de operação normal.

4.3 Mapeamento de políticas na rede

Embora o uso de políticas permita que se expresse em “alto nível” o que deve ocorrer na rede, alguns conceitos de “alto nível” precisam ser mapeados nos correspondentes conceitos de “baixo nível” para que a política se torne prática. Este mapeamento não precisa estar vinculado à linguagem de definição de políticas, mas deve estar bem definido para o sistema de gerenciamento. Exemplos desta necessidade são os conceitos de interface de rede, usuário e aplicação – os dois primeiros devem estar vinculados a algum elemento de rede e para o último deve estar claro quais fluxos de rede são seus.

No caso de interfaces de rede, sua identificação pode ser feita através de algum endereço vinculado à mesma. Tipicamente a identificação de uma interface é feita pelo seu endereço IP, possivelmente permitindo o gerenciamento do dispositivo através desta interface. Entretanto, em alguns casos, pode ser útil identificar uma interface por MAC. Um exemplo disto é a identificação de estações cliente nas redes IEEE802.16, pois o endereço IP destas pode não ser fixo. Além disto, a identificação da estação cliente frente às estações base está baseada no endereço MAC. No caso das estações base, é mais conveniente identificar suas interfaces por IP. Cabe ao sistema de gerenciamento identificar se possui condições de aplicar as políticas definidas utilizando o tipo de endereçamento que conhece.

No caso de usuários, convém que se saiba onde estão localizados na rede para que se possa reservar recursos para os mesmos. A identificação desta localização pode se dar de forma estática ou dinâmica (no caso de usuários móveis – que trocam de estação de trabalho). Também pode haver o caso de um usuário estar simultaneamente em duas ou mais estações de trabalho.

No caso de aplicações, tipicamente os fluxos das mesmas podem ser definidos por combinações de protocolos e portas envolvidos. Nem sempre, entretanto, é simples para a rede classificar os pacotes como provenientes de uma dada aplicação. Pacotes transmitidos não costumam conter identificadores de protocolo de aplicação e alguns protocolos de aplicação (por exemplo RTP) não possuem um conjunto de portas bem definido.

Outra informação de “baixo nível” que é conveniente que esteja disponível para o sistema de gerenciamento baseado em políticas são os limites de recursos nos equipamentos. Esta informação poderia permitir a identificação de conflitos de aplicação de políticas em tempo de definição de políticas, que é melhor do que descobri-los em tempo de aplicação.

4.4 Arquitetura de gerenciamento proposta

Não é construtivo pensar uma arquitetura de gerenciamento nova sem considerar anteriormente se as alternativas existentes já não são adequadas. Na área de gerenciamento baseado em políticas existe uma arquitetura padrão definida pelo IETF (YAVATKAR, PENDARAKIS e GUERIN, 2000) que pode muito bem ser utilizada para o gerenciamento de QoS em redes IEEE802.16 e é isto que será proposto aqui.

Todo o gerenciamento de QoS da rede IEEE802.16 deverá ser feito através das estações base. Desta forma, o sistema de gerenciamento não comprometerá os

possivelmente escassos recursos das estações cliente e ao mesmo tempo se tornará imune a problemas de falta de disponibilidade destas mesmas estações.

Como pode ser visto na Figura 4.3, as estações base (BS) farão o papel de PEP na arquitetura de gerenciamento do IETF. Supõe-se que os demais componentes da arquitetura tenham acesso às estações base através do *backhaul*.

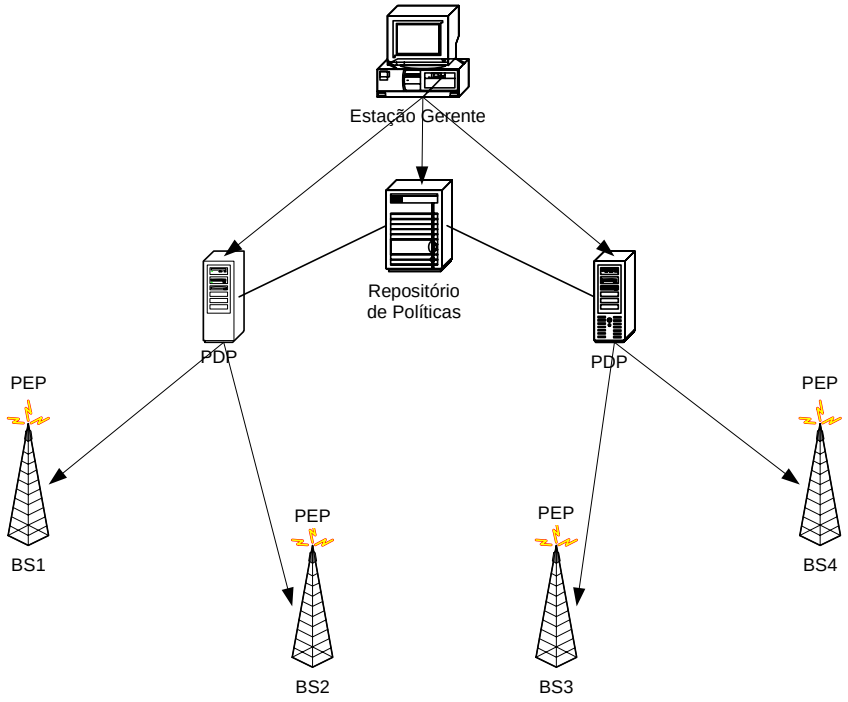


Figura 4.3: Modelo de gerenciamento baseado em políticas do IETF aplicado a redes IEEE802.16

Dentre os dois modos de gerenciamento previstos na arquitetura definida pelo IETF (*provisioning* e *outsourcing*), considerou-se o modo *provisioning* como o mais adequado ao gerenciamento de QoS em redes IEEE802.16, de forma que este deverá ser adotado. Embora o modo *outsourcing* apresente utilidade em algumas situações, ele é muito pouco utilizado na prática para que seja considerado relevante. Solicitações de recursos, feitas para a rede no modo *outsourcing* (e repassadas ao sistema de gerenciamento através da solicitação de uma decisão), poderiam perfeitamente ser feitas diretamente ao sistema de gerenciamento, de modo que este poderia tomar a decisão e, se aceita a solicitação, configurar a rede através do modo *provisioning*. Claro, o ideal seria haver um protocolo que permitisse às aplicações realizarem este tipo de solicitação por conta própria, além de algum mecanismo de descoberta que as permitisse identificar a quem realizar a solicitação.

Deixar de lado o modo *outsourcing* implica algumas restrições. Inicialmente, não é possível trabalhar com protocolos que exijam tomada de decisão por parte da rede (a menos que os critérios para tal tomada de decisão pudessem ser configurados nos dispositivos). O principal exemplo deste tipo de protocolo é o RSVP (BRADEN et al., 1997), utilizado na arquitetura de QoS IntServ (BRADEN, CLARK e SHENKER, 1994). Apesar de o uso de RSVP nas bordas da rede não ser tão problemático, esta limitação foi considerada aceitável em virtude de a arquitetura IntServ estar em decadência por falta de escalabilidade. Cabe notar que algum sistema poderia fazer a

solicitação de recursos ao sistema de gerenciamento em nome de alguma aplicação que anteriormente utilizasse RSVP.

Alguns casos em que o modo *outsourcing* seria útil em redes IEEE802.16 são a situação em que uma estação cliente solicita recursos através de mensagens DSA ou DSC, e a situação em que ocorre a migração de uma estação móvel. O primeiro destes casos é semelhante ao caso do RSVP: como a solicitação de recursos é feita à rede, o modo *provisioning* apenas trataria perfeitamente este caso se fosse possível configurar na rede os critérios de decisão (o que não ocorre). Desta forma, pode-se abordar este caso de duas formas: negando sempre a solicitação de recursos ou autorizando inicialmente e deixando por conta do sistema de gerenciamento o cancelamento ou não dos recursos após recebimento de notificação sobre a aceitação por parte da rede. Por simplicidade, a primeira abordagem será adotada nesta dissertação.

Quanto ao caso de migração de estações móveis, em função da opção pelo modo *provisioning*, ele deverá obrigatoriamente ser tratado configurando-se antecipadamente, na estação base destino, os requisitos de QoS para a estação móvel que irá migrar. Desta forma, a estação base destino saberá exatamente o serviço a ser fornecido à estação móvel que está entrando em sua área de atuação e não necessitará solicitar decisão alguma. A identificação da estação base destino em um *handover* pode ser feita controlando-se a localização das estações móveis através dos processos de *ranging* e *registering*, como será visto adiante.

O mapeamento específico das políticas definidas para uma dada rede IEEE802.16 em configurações de QoS para este tipo de rede dependerá da linguagem de definição de política sendo utilizada. Em geral, cada política definida para a rede (excetuando-se aspectos gerais de operação) deverá ser mapeada em pelo menos um *Service Flow* na rede IEEE802.16, com os requisitos de QoS deste *Service Flow* definidos através da ação especificada na política de rede, e com os filtros que identificam pacotes pertencentes a este *Service Flow* derivados da condição que ativa a política. Cada *Service Flow* poderá ser do tipo *unicast* ou *multicast* dependendo da política sendo mapeada. Como um exemplo simples, suponha-se que a política de rede defina que uma dada estação cliente X deverá ter pelo menos Y de banda reservada para *download*. Isto criaria um *Service Flow unicast* para *download* entre a estação cliente X e sua estação base com os requisitos de QoS indicando a reserva Y de banda.

Para saber quais estações base devem ser configuradas com os *Service Flows* para uma dada estação cliente, além de considerar as condições de aplicação das políticas, o sistema de gerenciamento deverá manter controle sobre a localização de cada estação cliente. Há dois processos do protocolo de enlace IEEE802.16 que permitem ao sistema de gerenciamento saber a localização de cada estação cliente: *ranging* e *registering*.

Na arquitetura proposta, o processo de *ranging* permitirá ao sistema de gerenciamento identificar em quais estações base devem ser configurados os *Service Flows* para uma dada estação cliente. O objetivo do processo de *ranging*, como definido no padrão IEEE802.16, é possibilitar o ajuste da potência de transmissão e a sincronização física da estação cliente com a estação base, bem como iniciar o processo de comunicação entre ambas. É através do processo de *ranging* que são informados os identificadores para as conexões básica e primária da estação cliente. Este processo ocorre tanto na entrada da estação cliente na rede quanto durante um processo de *handover*, caso em que é realizado com a estação base destino. Desta forma, a

configuração antecipada dos *Service Flows* nas estações base com as quais uma dada estação cliente tenha efetuado o processo de *ranging* permite tratar tanto a entrada na rede quanto o processo de *handover* por parte de uma estação cliente.

No caso do processo de *handover*, o *ranging* pode ocorrer ainda antes da decisão pelo *handover*, em uma etapa do processo denominada *cell reselection*. O objetivo da etapa *cell reselection* é que a estação móvel possa determinar a qualidade da comunicação com outras estações base. Nesta etapa, o padrão IEEE802.16e permite que as estações móveis meçam o sinal proveniente de estações base adjacentes à sua e, opcionalmente, realizem uma associação com as mesmas. Em sua versão mais simples, o processo de associação consiste em a estação móvel realizar o processo de *ranging* com aquela que poderá vir a ser sua futura estação base.

Apesar de o processo de associação ser considerado opcional, para que se utilize o processo de *ranging* como marco para configurar antecipadamente os *Service Flows* de uma estação móvel ele deverá ser considerado obrigatório. A utilização do *ranging* que ocorre durante o processo de *handover* como marco introduziria uma restrição de tempo real muito forte para o sistema de gerenciamento, de modo que ele dificilmente conseguiria configurar os *Service Flows* a tempo de o *handover* não sofrer impactos negativos. Nesta situação, os *Service Flows* provavelmente acabariam sendo cancelados por não estarem configurados na estação base destino antes do processo de *registering*, mesmo que posteriormente fossem novamente restabelecidos. Cabe notar que a realização do processo de associação visa agilizar a reentrada na rede da estação móvel quando ela estiver efetivamente realizando o *handover*, o que é algo desejável quando se trabalha com fluxos que apresentam requisitos de QoS. Desta forma, considera-se razoável assumir que as estações móveis serão configuradas de modo a sempre realizarem o processo de associação antes de um *handover*.

Embora a utilização do *ranging* do processo de associação tenda a aumentar o tempo que o sistema de gerenciamento possui para configurar os *Service Flows*, o padrão IEEE802.16e não estabelece um tempo mínimo entre o processo de *ranging* e a decisão da estação móvel pelo *handover*. Nos casos em que este tempo for muito pequeno, *Service Flows* acabariam por ser cancelados durante o *handover*. Este problema pode ser considerado o calcanhar de Aquiles da proposta sendo feita: embora ela possua uma vantagem bastante significativa, como será explorado adiante após a definição do processo de *registering*, há o risco de ela não funcionar em todos os casos. Como resultado da fase de experimentação, este trabalho irá apresentar medições que estimam quais os limites para se utilizar o *ranging* do processo de associação como base para a configuração antecipada dos *Service Flows*. Isto é, que fatores devem ser considerados no sitema de gerenciamento para que a arquitetura proposta não seja comprometida.

O processo de *ranging* cria um estado na estação base a respeito da estação cliente, visto que através dele são fornecidos os identificadores das conexões básica e primária da estação cliente. Considerando-se que os *Service Flows* serão configurados pelo sistema de gerenciamento quando este estado for criado na estação base para a estação cliente, e serão desconfigurados quando este estado expirar, é relevante saber por quanto tempo este estado será mantido. De fato, o padrão IEEE802.16e estabelece que o tempo que este estado é mantido é configurado pelo parâmetro T9 da estação base. Este parâmetro possui um valor mínimo de 300ms e valor máximo não definido. Como não é interessante ficar configurando e desconfigurando os *Service Flows* muito

freqüentemente, o ideal é ajustar este parâmetro para um valor relativamente alto (como 60s).

Com relação à utilização do processo de *ranging* para configurar os *Service Flows*, cabe mencionar que os *Service Flows* não passarão por controle de admissão ou serão tornados ativos enquanto a estação cliente não concluir o processo de entrada na rede ou *handover*, de modo que não haverá consumo de recursos em vão. A estratégia de configuração dos *Service Flows* baseada no *ranging* funciona bem também para prever a reentrada na rede de uma estação móvel em modo *idle*, visto que o processo de *location update*, que deve ser efetuado periodicamente por uma estação móvel para atualizar sua localização, também exige o processo de *ranging*. Convém notar que a configuração de *Service Flows* em cada estação base com a qual uma dada estação cliente tenha efetuado *ranging* poderá ser diferente em função das políticas definidas para a rede – isto afetará o comportamento das estações base em processos de *handover*.

A Figura 4.4 ilustra a movimentação de uma estação cliente (SS) que se comunica através da estação base BS1. Em algum momento, ela passa a receber o sinal das estações BS2 e BS3 e, percebendo que, em breve, poderá ter de realizar um *handover*, resolve efetuar um processo de associação com estas estações. Ao perceber que o *ranging* das associações ocorreu, o sistema de gerenciamento provisiona antecipadamente os *Service Flows* para esta estação cliente em suas possíveis estações base destino.

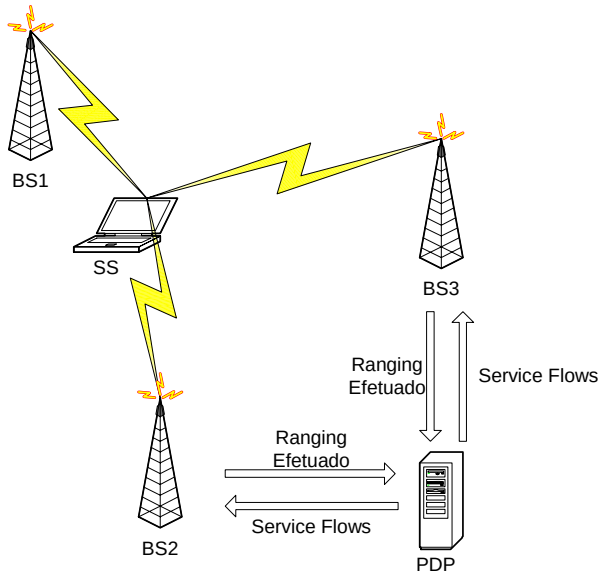


Figura 4.4: Estação cliente (SS) deslocando-se para cobertura de novas estações base (BS)

O processo de *registering* permite ao sistema de gerenciamento identificar a entrada de uma estação cliente na rede (ou conclusão de processo de *handover*) através de uma dada estação base. Este processo permite controlar quais os períodos em que uma dada estação cliente esteve *on-line* e também saber através de quais estações base se comunicou. Uma estação cliente tem seus *Service Flows* estabelecidos com uma dada estação base apenas após concluir o processo de *registering* com a mesma. Desta forma, não há transmissão de dados de usuário antes de efetuado o registro. Sabendo qual a última estação base em que uma estação cliente esteve registrada antes de sair da rede, o sistema de gerenciamento pode deixá-la pré-configurada com os *Service Flows* da estação cliente em questão visando agilizar uma futura reentrada da mesma na rede.

Neste caso, o sistema de gerenciamento possivelmente estará antecipando a configuração que seria disparada apenas mediante um futuro processo de *ranging* da estação cliente com esta mesma estação base.

O processo de *registering* poderia ser utilizado como alternativa ao processo de *ranging* para a configuração dos *Service Flows*. Neste caso, poderia-se configurar os *Service Flows* para uma dada estação móvel na estação base em que ela estivesse registrada e também em todas as estações base adjacentes à mesma. Isto permitiria resolver a restrição de tempo real imposta pela utilização do processo de *ranging*. A grande vantagem da utilização do processo de *ranging* sobre esta alternativa é que o *ranging* permite que o PDP gerencie uma estação base de forma isolada, sem requerer comunicação com outros PDPs. Na alternativa baseada no *registering*, os PDPs teriam de conhecer a topologia lógica entre as estações base (isto é, quais estações base são vizinhas) e trocar informações a respeito das estações móveis registradas em cada estação base, visto que uma estação base vizinha à que dada estação móvel está registrada pode ser gerenciada por um PDP diferente daquele que gerencia a própria estação base a que a estação móvel está registrada.

O padrão IEEE802.16e prevê a possibilidade de se empregar diversidade em um processo de *handover*, caso em que o mesmo é denominado *macro-diversity handover* (MDHO). Quando MDHO é empregado, as estações móveis permanecem em um constante processo de *handover*, no qual realizam transmissão e recepção simultânea via um conjunto de estações base (ao invés de apenas uma). Este conjunto de estações, denominado *diversity set*, é escolhido por cada estação móvel com base na qualidade de sinal e na possibilidade de sincronização física entre todas as estações constantes em seu *diversity set*. No caso específico em que MDHO é empregado, pode ocorrer que uma dada estação móvel esteja registrada em múltiplas estações base simultaneamente (todas as que constam em seu *diversity set*). Nestes casos, ficará a critério de cada configuração específica de MDHO como dirimir eventuais questões como requisitos de QoS distintos em cada estação base participante do *diversity set.* O sistema de gerenciamento não deve tentar resolver conflitos entre as definições de políticas para as estações base no *diversity set* de uma dada estação cliente, pois este *diversity set* pode alterar-se rapidamente e o sistema de gerenciamento não teria a agilidade necessária para adaptar a configuração de rede a cada situação. Convém notar que isto não impedirá o sistema de gerenciamento de saber, para fins de contabilização, quais os *Service Flows* ativos no conjunto de estações base para uma dada estação cliente e quais os requisitos de QoS sendo entregues em cada *Service Flow*.

A Figura 4.5 ilustra uma estação cliente (SS) empregando MDHO para comunicar-se através de duas estações base (BS1 e BS2). Apesar de perceber o sinal da estação base BS3, a estação cliente não o considerou forte o bastante para incluir a estação BS3 em seu *diversity set*. Neste caso, a estação SS estaria registrada simultaneamente com a estação BS1 e com a estação BS2.

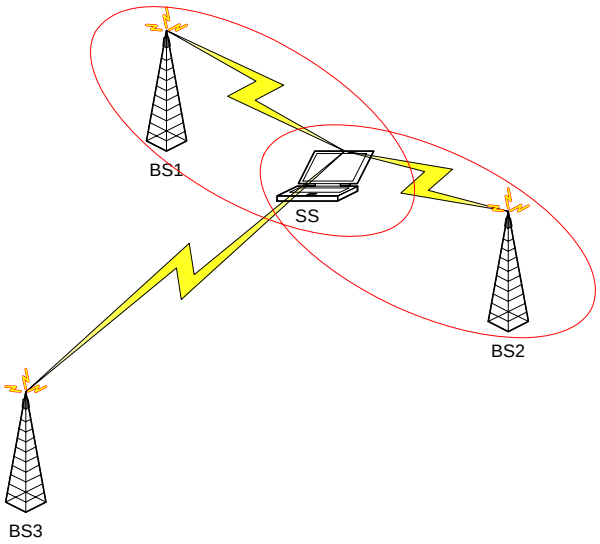


Figura 4.5: Estação cliente empregando MDHO para comunicar-se através de duas estações base

No caso das redes IEEE802.16, a localização dos usuários mencionados nas políticas da rede deve ser dada em termos de um mapeamento entre usuário e endereço MAC da estação móvel associada ao mesmo. Desta forma, o sistema de gerenciamento saberá para qual estação cliente devem ser mapeadas as políticas que se aplicam ao usuário. Este mapeamento pode ser definido estaticamente ou atualizado dinamicamente para usuários móveis (que podem trocar de estação cliente), dependendo da flexibilidade do sistema de gerenciamento. Também é possível que um usuário se encontre em mais de uma estação cliente ao mesmo tempo. Neste caso, o sistema de gerenciamento deverá aplicar a política a todas as estações onde o usuário “se encontre”. Isto deve gerar um consumo maior de recursos, mas pode eventualmente ser considerado aceitável. Uma alternativa para a aplicação obrigatória em todas as estações seria a linguagem de definição de políticas permitir uma definição clara do que deve ocorrer caso o usuário se encontre em mais de uma localização.

Embora não seja propriamente parte da definição da arquitetura de gerenciamento, convém notar que um sistema comercial deve manter registros sobre a utilização de recursos de forma a permitir a cobrança dos usuários e um melhor planejamento de rede. Registros da utilização de recursos por política e estação base permitem identificar as políticas que consomem mais recursos. Registros de utilização de recursos por usuário permitem realizar a cobrança e aperfeiçoar o plano comercial sendo utilizado.

5 MIB PARA GERENCIAMENTO

Tendo-se optado por utilizar a arquitetura de gerenciamento baseado em políticas do IETF em modo *provisioning*, o próximo passo é escolher qual o protocolo mais adequado para a comunicação entre os PDPs (parte integrante do sistema de gerenciamento) e os PEPs (as estações base). Não foi difícil optar pelo protocolo SNMP (CASE et al., 1990), visto que o mesmo se trata tanto de um padrão de direito estabelecido pelo IETF quanto um padrão de fato largamente utilizado no mercado.

Durante a escrita deste trabalho, já havia uma MIB (IEEE, 2005b) definida para o padrão IEEE802.16d (IEEE, 2004), que é o padrão definido para operação fixa (ou nômade). Entretanto, aqui estamos propondo o gerenciamento de QoS para o padrão IEEE802.16e (IEEE, 2006), que é o padrão para operação móvel. A construção da MIB para o padrão móvel ainda está em andamento e sem divulgações de resultados intermediários significativos. Também percebe-se que a MIB do padrão fixo se torna incompleta quando se pensa no gerenciamento de QoS para o padrão móvel. Desta forma, este trabalho irá propor uma série de modificações e adições à MIB do padrão fixo que permitirão o correto gerenciamento da rede IEEE802.16 em operação móvel.

5.1 Requisitos para a MIB

Antes que sejam apresentadas as modificações propostas na MIB de operação fixa, convém que se faça uma análise sobre quais os requisitos que devem ser cumpridos pela mesma para que o gerenciamento de QoS em redes IEEE802.16 possa ocorrer. Os requisitos apresentados também exigirão suporte das estações base, onde a MIB deverá ser implementada.

Considerando-se que as estações base deverão realizar um CAC final nos fluxos provisionados, deve-se manter separado na MIB aquilo que o sistema de gerenciamento deseja que a rede forneça daquilo que a rede realmente está fornecendo. Assim o sistema de gerenciamento poderá sempre identificar o que realmente ocorre na rede, independentemente daquilo que ele próprio configurou. O sistema de gerenciamento deverá também ter a opção de ser notificado sobre a admissão de *Service Flows* na rede, de modo a poder controlar a utilização de recursos. O sistema de gerenciamento necessariamente deverá poder identificar se a admissão se deu em modo normal ou modo degradado, podendo reagir a cada situação de acordo. Outra notificação importante é uma que indique a não admissão, por falta de recursos, de um *Service Flow* provisionado.

Também seria interessante que as estações base tivessem a possibilidade de notificar o sistema de gerenciamento sobre situações de degradação, permitindo ao mesmo a

tomada de ações reativas à situação. Convém notar que isto não seria um requisito obrigatório para a MIB, visto que sistemas de monitoramento externos também poderiam cumprir com esta função. De todo modo, as estações base teoricamente são capazes de indicar situações de degradação de forma mais precisa.

Pensando na abordagem sendo adotada para tratar o processo de *handover*, deve ser possível especificar reservas para uma estação móvel mesmo que ela não se encontre registrada um uma dada estação base. Com isto torna-se possível provisionar antecipadamente os *Service Flows* para uma estação que possa vir a entrar em uma dada célula.

O sistema de gerenciamento deve possuir controle total sobre os *Service Flows* existentes na rede. Sem isto pode ocorrer que determinadas políticas da rede não consigam ser cumpridas.

As estações base devem poder notificar o sistema de gerenciamento sobre todos os eventos que ocorrem com as estações cliente. Deve ser possível saber quando uma estação cliente entrou ou saiu da rede, efetuou *handover* para outra célula, entrou ou saiu do modo *idle*, registrou-se ou desregistrou-se, efetuou *ranging* ou teve seu *ranging* invalidado, passou a operar em modo *sleep*, etc. Convém notar aqui que nem todos estes eventos são significativos para o gerenciamento de QoS, mas permitem um maior controle do sistema de gerenciamento sobre as estações cliente. Na modificação que será proposta a preocupação maior será em disponibilizar os eventos necessários para o gerenciamento de QoS.

No que se refere à situação específica em que estações cliente solicitam recursos através de mensagens DSA ou DSC (uma situação que seria trivialmente tratada no modelo *outsourcing*), duas abordagens haviam sido pensadas para o modelo *provisioning* na definição da arquitetura: negar sempre a solicitação ou autorizar os recursos inicialmente, mas notificar o sistema de gerenciamento para que este decida sobre sua manutenção. É interessante que ambas as abordagens possam ser seguidas, o que exige a possibilidade de configurar a negação de solicitações e a emissão de notificações sobre cada *Service Flow* admitido ou modificado na rede de forma independente do sistema de gerenciamento.

5.2 Proposta de modificação na MIB

Esta seção irá apresentar uma possível modificação na MIB de gerenciamento de redes IEEE802.16 em operação fixa (IEEE, 2005b) de modo que a mesma permita o gerenciamento de redes IEEE802.16 em operação móvel. Nesta alteração serão levados em consideração os requisitos levantados na seção anterior. Também serão apresentados alguns aspectos comportamentais do sistema de gerenciamento frente às novas definições.

As modificações foram agrupadas em subseções para facilitar o entendimento. A MIB de gerenciamento de redes IEEE802.16 em operação fixa será doravante denominada por MIB original, por simplificação. Trechos do arquivo de MIB novo foram incluídos em apêndices com algumas marcações em negrito para facilitar a visualização e as modificações efetuadas no trecho em questão com relação à MIB original sublinhadas e sombreadas. Diversos objetos da MIB original tiveram seus OIDs

renumerados, mas não serão apresentados estes detalhes por serem de baixa relevância para o entendimento das modificações.

Um fator interessante no gerenciamento baseado em políticas é que ele é independente da forma como os dispositivos são configurados. Assim, é irrelevante se a MIB que futuramente será proposta pelo IEEE para gerenciamento de redes IEEE802.16 em operação móvel for diferente da MIB que será sugerida nesta seção. Contanto que se possa configurar os dispositivos adequadamente de acordo com as políticas de rede, o gerenciamento baseado em políticas poderá ser utilizado.

5.2.1 Visão geral da MIB original

Não será feita uma apresentação detalhada das possibilidades oferecidas pela MIB original. A idéia aqui é passar uma visão geral da estrutura da MIB original de forma a facilitar o entendimento das modificações sendo propostas.

Inicialmente, convém notar que o padrão IEEE802.16f (IEEE, 2005b) define duas sub-árvores: *wmanIfMib* e *wmanDevMib*. Enquanto a sub-árvore *wmanIfMib* define objetos mais propriamente relacionados ao gerenciamento de uma rede IEEE802.16, a sub-árvore *wmanDevMib* define objetos mais vinculados a questões físicas dos equipamentos, como gerenciamento de *firmware*. O interesse aqui está na sub-árvore *wmanIfMib*, que pode ser vista na Figura 5.1.

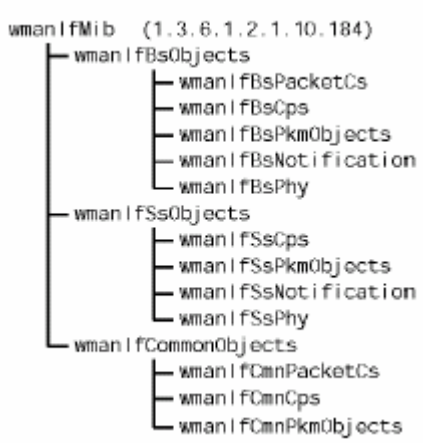


Figura 5.1: Estrutura da sub-árvore *wmanIfMib* (IEEE, 2005b)

A sub-árvore *wmanIfMib* foi dividida em objetos relativos apenas às estações base (*wmanIfBsObjects*), objetos relativos apenas às estações cliente (*wmanIfSsObjects*) e objetos relativos a ambos os tipos de estação (*wmanIfCommonObjects*). Apenas os objetos da estação base e objetos comuns são de interesse para este trabalho, visto que não se pretende gerenciar diretamente as estações cliente.

Dentro de *wmanIfBsObjects*, as sub-árvores *wmanIfBsPacketCs*, *wmanIfBsCps* e *wmanIfBsNotification* são as que representam maior interesse para a configuração de QoS. A sub-árvore *wmanIfBsPkmObjets* apresenta basicamente configurações para o subsistema de segurança e a sub-árvore *wmanIfBsPhy* apresenta basicamente configurações relativas aos padrões de transmissão físicos suportados pela padrão IEEE802.16. As tabelas existentes dentro de *wmanIfBsPacketCs* (relativos à *Packet Convergence Sublayer*) podem ser vistas na Figura 5.2.

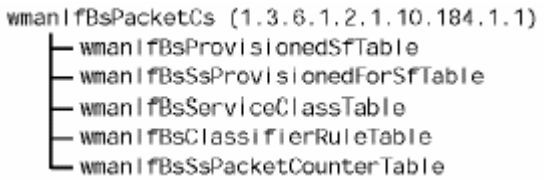


Figura 5.2: Estrutura da sub-árvore *wmanIfBsPacketCs* (IEEE, 2005b)

A sub-árvore *wmanIfBsPacketCs* contém um conjunto de tabelas que permite ao sistema de gerenciamento configurar quais devem ser os *Service Flows* presentes na rede, quais os parâmetros de QoS vinculados aos mesmos e como classificar pacotes nestes *Service Flows*. A tabela *wmanIfBsProvisionedSfTable* é onde o sistema de gerenciamento pode criar novos SFs a serem provisionados para determinadas estações cliente. A tabela *wmanIfBsSsProvisionedForSfTable* faz a ponte entre os SFs criados e quais estações clientes devem ter acesso a tal SF. Esta tabela intermediária é importante porque um SF *multicast* pode estar vinculado a mais de uma estação cliente. Além de ser vinculado a estações cliente, um SF referencia uma entrada na tabela *wmanIfBsServiceClassTable*, indicando qual sua classe de QoS. A classe de QoS é que determina quais são os parâmetros de operação (reserva de banda, atraso máximo, etc.) que a rede se compromete a fornecer aos pacotes pertencentes à fluxos da tal classe. Diferentemente da especificação da camada de enlace, cada SF na MIB obrigatoriamente estará associado a uma classe de QoS. Por fim, a tabela *wmanIfBsClassifierRuleTable* contém os classificadores (conjuntos de condições) que permitem identificar quais pacotes devem ser associados a cada SF a ser provisionado. Cada entrada nesta tabela contém um apontador para o SF identificado na tabela *wmanIfBsProvisionedSfTable*. A tabela *wmanIfBsSsPacketCounterTable* contém apenas contadores de utilização para cada SF presente na rede (mesmo os não provisionados).

O segundo ramo da sub-árvore *wmanIfBsObjects*, denominado *wmanIfBsCps*, contém objetos referentes à *Common Part Sublayer*. As tabelas desta sub-árvore podem ser vistas na Figura 5.3.

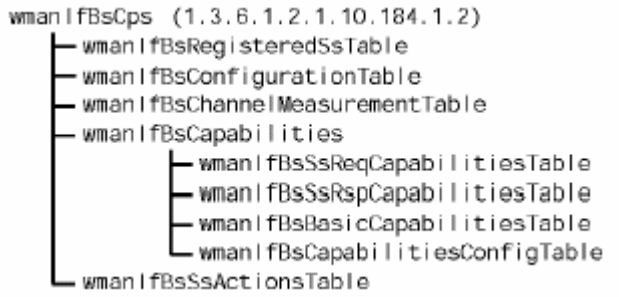


Figura 5.3: Estrutura da sub-árvore *wmanIfBsCps* (IEEE, 2005b)

Desta sub-árvore as tabelas de maior interesse são a *wmanIfBsRegisteredSsTable* e *wmanIfBsConfigurationTable*. A primeira apresenta todas as estações cliente registradas com a estação base. A segunda contém configurações gerais de operação para a estação base, de modo que algumas são relativas ao gerenciamento de QoS. Quanto às demais tabelas, a tabela *wmanIfBsChannelMeasurementTable* apresenta históricos de medições sobre a qualidade do sinal de comunicação com as estações clientes em ambos os sentidos, as tabelas abaixo de *wmanIfBsCapabilities* apresentam configurações e resultados do processo de negociação de capacidades com as estações cliente e a tabela *wmanIfBsSsActionsTable* permite instruir a estação base a enviar comandos (RESET por exemplo) às estações cliente.

O terceiro ramo de interesse na sub-árvore *wmanIfBsObjects* é o ramo *wmanIfBsNotification*. Este ramo contém duas sub-árvores: *wmanIfBsTrapControl* e *wmanIfBsTrapDefinitions*. Basicamente, a primeira sub-árvore contém objetos e tabelas que permitem controlar se e quando notificações são emitidas. A segunda sub-árvore, além de conter o prefixo comum a todas as notificações (*wmanIfBsTrapPrefix*), do qual todas as notificações emitidas por estações base são sub-objetos, contém uma tabela para armazenar os objetos emitidos nas notificações: *wmanIfBsSsNotificationObjectsTable*.

A sub-árvore que mantém os objetos comuns tanto a estações base quanto a estações cliente (*wmanIfCommonObjects*) apresenta basicamente objetos que refletem o estado atual da rede, não sendo permitidas alterações em tais objetos. As duas sub-árvores de interesse, *wmanIfCmnPacketCs* e *wmanIfCmnCps* podem ser vistas na Figura 5.4 e na Figura 5.5 respectivamente.

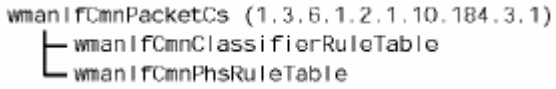


Figura 5.4: Estrutura da sub-árvore *wmanIfCmnPacketCs* (IEEE, 2005b)

Figura 5.5: Estrutura da sub-árvore *wmanIfCmnCps* (IEEE, 2005b)

As tabelas sob *wmanIfCmnPacketCs* (*wmanIfCmnClassifierRuleTable* e *wmanIfCmnPhsRuleTable*) representam, respectivamente, os classificadores e regras de PHS (Payload Header Suppression) em utilização na rede. Já a tabela *wmanIfCmnCpsServiceFlowTable* apresenta todos os *Service Flows* existentes na rede, sejam eles provisionados pelo sistema de gerenciamento ou criados automaticamente. Os parâmetros de QoS sendo aplicados a determinado SF estão juntos em sua entrada nesta tabela. Convém notar que estes parâmetros de QoS podem ser diferentes dos que originalmente o sistema de gerenciamento solicitou que fossem provisionados, pois podem ocorrer modificações externas ao controle do sistema de gerenciamento. Desta forma, os parâmetros de QoS apresentados nesta tabela podem ser considerados como o nível de serviço real que a rede está tentando entregar aos fluxos.

A tabela *wmanIfCmnBsSsConfigurationTable* não é de muito interesse para o gerenciamento de QoS, visto que apresenta apenas a possibilidade de configuração de alguns temporizadores da rede. As próximas seções irão apresentar as modificações sendo propostas nesta MIB.

5.2.2 Ajuste nos identificadores de fluxos

Como visto, a MIB original possui uma tabela que permite ao sistema de gerenciamento provisionar *Service Flows* nas estações cliente via estação base: *wmanIfBsProvisionedSfTable*. Esta tabela originalmente era indexada pela interface de rede na estação base (setor sendo atendido) e pelo SFID. Como os SFIDs são únicos apenas no contexto entre a estação base e uma estação cliente, julgou-se necessário estender o índice desta tabela com um campo adicional (*wmanIfBsSfMacAddress*), de forma que uma estação móvel migrando entre estações base não tenha problemas com conflitos de SFIDs já utilizados por outras estações cliente na estação base destino. Naturalmente o problema de conflito de identificadores afetaria apenas o sistema de

gerenciamento, visto que o protocolo MAC IEEE802.16 não teria nenhum problema com a unicidade que ele definiu para os identificadores. A localização dos elementos modificados descritos nesta seção pode ser vista na Figura 5.6. As modificações podem ser vistas em maior detalhe no APÊNDICE A.

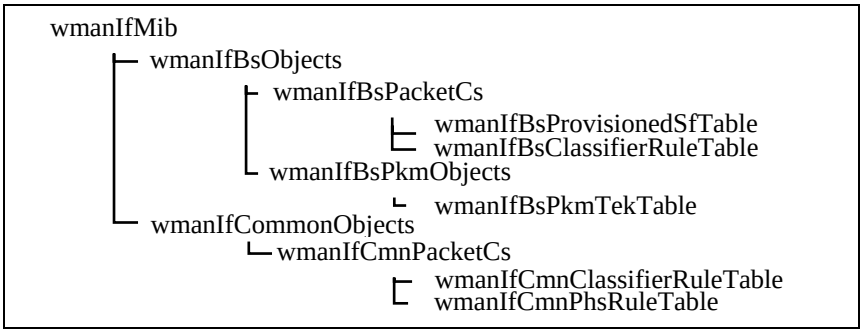


Figura 5.6: Objetos modificados para ajuste nos identificadores de fluxo

O campo *wmanIfBsSfMacAddress* foi definido de forma que seja igual ao endereço MAC da estação cliente sendo provisionada no caso de fluxos unicast e seja o endereço MAC de broadcast no caso de fluxos multicast. Desta forma cria-se um espaço de endereçamento único para cada estação cliente e um espaço compartilhado para fluxos multicast. Caso um mesmo SFID exista tanto no espaço de endereçamento multicast quanto no espaço de endereçamento para uma dada estação cliente, assume-se que o SFID da dita estação cliente tenha precedência no entendimento sobre o que deve ser provisionado para tal estação.

O campo de endereço MAC da tabela *wmanIfBsSsProvisionedForSfTable*, que vincula os SFs às estações cliente, continuará sempre contendo endereços MAC reais das estações cliente, permitindo identificar as estações que receberão fluxos *multicast*. No caso dos fluxos *unicast*, este campo conterá um endereço MAC em redundância com o existente na tabela *wmanIfBsProvisionedSfTable*.

A tabela *wmanIfBsClassifierRuleTable*, que determina como classificar os pacotes nos respectivos *Service Flows*, também teve de ser alterada por conta da modificação na tabela *wmanIfBsProvisionedSfTable*. A alteração foi necessária para que as entradas na tabela *wmanIfBsProvisionedSfTable* fossem referenciadas pelo seu novo índice.

Outras tabelas que tiveram que ser alteradas pelo mesmo motivo são as tabelas *wmanIfCmnClassifierRuleTable* e *wmanIfCmnPhsRuleTable*. A alteração incluiu a referência ao endereço MAC que já fazia parte do índice da tabela *wmanIfCmnCpsServiceFlowTable* (por este motivo, esta última tabela não precisou ser alterada sobre este aspecto).

Na tabela *wmanIfCmnCpsServiceFlowTable*, os endereços MAC são sempre os endereços reais das estações cliente, já que esta tabela contém os SF reais existentes na rede. Um *Service Flow* de *multicast* realmente é repetido para cada estação cliente que o escuta atualmente quando a tabela é acessada na estação base. Quando a tabela é acessada em uma estação cliente, todas as entradas na tabela *wmanIfCmnCpsServiceFlowTable* possuem endereços MAC da própria estação cliente (normalmente apenas um).

Por uma questão de consistência, a tabela *wmanIfBsPkmTekTable* também foi modificada. Esta tabela mantém informações sobre as chaves de encriptação de tráfego

das SAIDs, de modo que não tem propriamente a ver com o gerenciamento de QoS. Entretanto, ela reflete o mesmo problema que ocorre com os SFIDs do ponto de vista dos SAIDs.

5.2.3 Maior controle sobre Service Flows

Um dos requisitos necessários para o gerenciamento era a possibilidade de o sistema de gerenciamento controlar totalmente os *Service Flows*, mesmo aqueles que são criados automaticamente na rede (não provisionados pelo sistema de gerenciamento). Para isto, foram introduzidas modificações nas tabelas *wmanIfCmnCpsServiceFlowTable* e *wmanIfBsConfigurationTable*, cuja localização é dada na Figura 5.7. As modificações podem ser vistas em maior detalhe no APÊNDICE B.

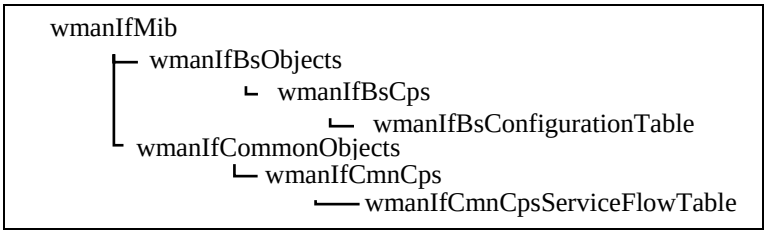


Figura 5.7: Objetos modificados para maior controle sobre *Service Flows*

Na tabela *wmanIfCmnCpsServiceFlowTable*, que lista todos os *Service Flows* existentes na rede, foram introduzidos campos necessários para tornar um *Service Flow* criado automaticamente gerenciável. Convém notar que esta capacidade de gerenciamento somente é disponível no acesso via estação base, já que a MIB para estações cliente não possui nenhum gerenciamento para *Service Flows* provisionados.

O novo campo *wmanIfCmnCpsSfProvisionStatus* permite identificar se um dado *Service Flow* foi provisionado (consta na tabela *wmanIfBsProvisionedSfTable*) ou foi automaticamente criado. Marcando um *Service Flow* criado automaticamente como provisionado, ele se torna automaticamente disponível para gerenciamento nas tabelas para gerência de *Service Flows* provisionados (as únicas cujos parâmetros são modificáveis pelo sistema de gerenciamento). Uma marcação extra também permite eliminar um *Service Flow* sem torná-lo gerenciável.

O novo campo *wmanIfCmnCpsSfType* permite identificar se um fluxo é *unicast* ou *multicast*. Isto é importante porque fluxos multicast, quando procurados na tabela *wmanIfBsProvisionedSfTable*, são indexados utilizando o endereço MAC de broadcast, conforme apresentado na seção anterior. Cabe notar aqui que esta informação pode não ser confiável em acessos via estações cliente porque elas podem ver fluxos *multicast* como sendo fluxos *unicast*.

A alteração na tabela *wmanIfBsConfigurationTable* criou uma nova possibilidade de restrição ao gerenciamento autônomo de *Service Flows* pela rede. O parâmetro *wmanIfBsSsSfMgmtEnabled* permite indicar que não devem ser aceitas solicitações autônomas de criação, modificação ou remoção de *Service Flows* feitas por estações cliente via protocolo MAC da rede IEEE802.16 – isto possibilita a solução de negar sempre solicitações de recursos via este tipo de solicitação. O parâmetro *wmanIfBsAutoSfidEnabled*, que já existia na MIB original, permite indicar se a estação base pode criar autonomamente SFIDs. De fato é uma forma de restringir os *Service Flows* na rede aos provisionados pelo sistema de gerenciamento. O problema com este

parâmetro é que ele não é capaz de rejeitar alterações nos parâmetros de QoS de *Service Flows* existentes ou evitar que algum seja removido – daí a necessidade do novo parâmetro criado.

Uma alteração extra que não tem a ver com o maior controle de *Service Flows*, mas que também foi incluída na tabela *wmanIfBsConfigurationTable*, foi a possibilidade de definir um nível mínimo de serviço para fluxos do tipo melhor esforço – o nome do campo é *wmanIfBsMinBestEffortResources*. Sem a possibilidade de definição deste nível mínimo, haveria o risco de a estação base admitir um número excessivo de *Service Flows* que especificam reservas e causar o não atendimento dos fluxos operando com melhor esforço.

Como últimas alterações para permitir um maior controle sobre os *Service Flows*, foram criadas notificações para indicar quando um *Service Flow* é criado autonomamente, quando um *Service Flow* provisionado é admitido na rede, ou quando sua admissão é negada pela estação base. A apresentação das novas notificações será feita em uma seção posterior.

5.2.4 Maior controle sobre as estações cliente

Para permitir um maior controle das estações cliente, diversas novas notificações foram criadas de forma que a estação base possa indicar ao sistema de gerenciamento quando uma estação cliente conectou-se ou desconectou-se da rede, registrou-se ou desregistrou-se, tentou efetuar *handover* ou teve o mesmo cancelado, sofreu alteração em sua capacidade de transmissão ou recepção. Além das novas notificações, dois campos foram incluídos na tabela *wmanIfBsRegisteredSsTable* para indicar a capacidade de transmissão e recepção corrente das estações cliente e uma nova tabela foi criada para manter o registro de todas as estações cliente conectadas em algum setor da estação base, mesmo que não estejam registradas.

A apresentação das novas notificações será deixada para uma seção posterior, sendo aqui apresentada apenas a nova tabela (*wmanIfBsConnectedSsTable*) e os novos campos. Convém notar que as notificações de registro já existiam e que há também uma notificação específica para acompanhar o progresso (estados) de uma estação cliente no processo de entrada na rede. A localização dos objetos cuja modificação é descrita nesta seção pode ser vista na Figura 5.8. As modificações podem ser vistas em maior detalhe no APÊNDICE C.

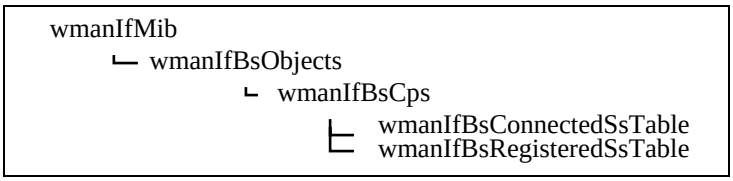


Figura 5.8: Objetos modificados para maior controle sobre as estações cliente

Para entender a nova tabela, ressalta-se que o conceito de conexão aqui significa simplesmente que a estação possui identificadores de conexão básica e primária fornecidos durante o processo de *ranging* e ainda não invalidados pela estação base. Desta forma, uma estação cliente pode estar apenas conectada, conectada e registrada ou nem conectada nem registrada em um dado setor da estação base. Uma estação cliente pode estar registrada em um setor da estação base, mas não estar registrada em outro setor. O mesmo pode ser dito sobre estar ou não conectada em setores diferentes. A

Figura 5.9 ilustra as possibilidades de conexão para uma estação cliente. A estação SS está conectada e registrada ao setor 1 da estação BS1, está apenas conectada ao setor 2 da estação BS2 e não está conectada ou registrada a qualquer setor da estação BS3.

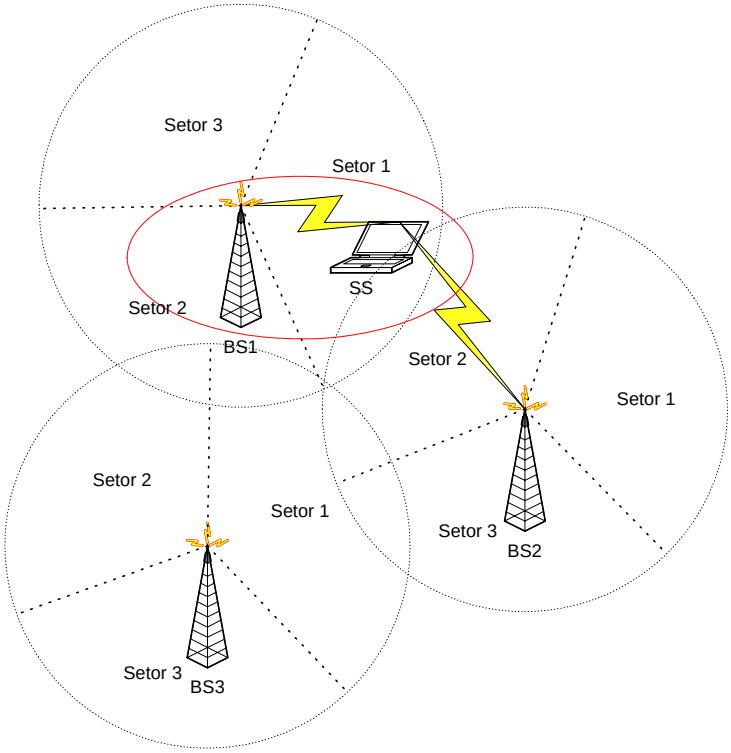


Figura 5.9: Possibilidades de conexão para uma estação cliente (SS)

Há um fato que chamou atenção durante os estudos sobre IEEE802.16 e que levou a conclusão de que o registro das estações é mantido separado para cada setor gerenciado por uma mesma estação base. Embora isto não tenha sido mencionado explicitamente no padrão, a alternância entre setores diferentes de uma mesma estação base parece envolver um processo de *handover*. Este *handover* se processaria como qualquer outro *handover*, ocorrendo inclusive a ilusão de troca de estação base em função de BSIDs diferentes em cada setor (apesar de ser a mesma entidade que controlaria ambos os setores). Uma análise da composição do BSID pode levar a esta conclusão. Os 24 bits mais significativos do BSID devem ser utilizados como *operator ID*, conforme é mencionado na seção 6.3.2.3.2 do padrão IEEE802.16e (IEEE, 2006). Embora não haja menção sobre o significado dos demais 24 bits, a seção 8.2.1.8.1 do padrão IEEE802.16d (IEEE, 2004), que foi mantida inalterada pelo padrão IEEE802.16e, menciona que o *Sector ID* representa pelo menos (trata-se de uma especificação de compressão) os 8 bits menos significativos do BSID. Isto leva a crer que uma mesma estação base pode ter mais de um BSID em função de controlar vários setores, o que realmente seria necessário para que houvesse um processo de *handover* entre estes setores. Isto também justifica a total separação de informações sobre estações cliente por interface de rede da estação base (basta olhar que o índice de tabelas como a que mantém as estações cliente registradas inclui o índice da interface de rede).

Continuando com os detalhes sobre a nova tabela, todas as informações que se tornam disponíveis no processo de *ranging* de uma estação cliente que esteja entrando em determinado setor de uma estação base foram incluídas nesta tabela. De certa forma estas informações se tornaram redundantes com as informações equivalentes

apresentadas na tabela *wmanIfBsRegisteredSsTable*, que mantém o registro das estações registradas, mas julgou-se que pode ser útil ter estas informações para estações que ainda não chegaram a se registrar.

Os dois novos campos na tabela *wmanIfBsRegisteredSsTable* representam respectivamente a taxa de transmissão e de recepção corrente para cada estação cliente registrada. São eles: *wmanIfBsSsMaxUplinkTransmission* e *wmanIfBsSsMaxDownlinkTransmission*. O conhecimento da capacidade de transmissão e recepção de cada estação cliente pode ser útil ao sistema de gerenciamento, entre outras possibilidades, para dar o tratamento adequado a uma situação de degradação – como será visto adiante.

5.2.5 Novas notificações

Esta seção apresentará todas as novas notificações criadas para emissão por estações base. Optou-se por agrupar todas em uma única seção porque as modificações são todas no mesmo trecho do arquivo de MIB e as tabelas de configuração e objetos são compartilhadas. Também é útil que estejam agrupadas para uma referência rápida sobre as notificações existentes.

A MIB original possuía cinco notificações: *wmanIfBsSsStatusNotificationTrap*, *wmanIfBsSsDynamicServiceFailTrap*, *wmanIfBsSsRssiStatusChangeTrap*, *wmanIfBsSsRegistrerTrap* e *wmanIfBsSsPkmFailTrap*. A primeira é enviada para relatar alterações de estado das estações cliente ao longo do processo de entrada na rede. A segunda relata falhas na criação, alteração ou remoção de *Service Flows* dinamicamente. A terceira notifica sobre alterações na potência do sinal de recepção vindo de alguma estação cliente. A quarta indica registro ou desregistro de uma estação cliente em dado setor da estação base. A última notifica sobre falha em alguma operação do subsistema de segurança.

Poucas modificações foram feitas nas notificações originais. Além de uma mudança no nome da notificação sobre registros para *wmanIfBsSsRegisterTrap* (havia um “r” a mais – ver no parágrafo anterior), a definição das notificações *wmanIfBsSsRegisterTrap* e *wmanIfBsSsPkmFailTrap* foi modificada para incluir o índice da interface de rede, que não era incluído na MIB original.

São sete as novas notificações: *wmanIfBsDynamicServiceSuccessTrap*, *wmanIfBsSsServiceAdmissionTrap*, *wmanIfBsSsConnectTrap*, *wmanIfBsSsHandoverTrap*, *wmanIfBsSsBurstProfileChangeTrap*, *wmanIfBsSsServiceDegradationTrap* e *wmanIfBsNetDegradationTrap*. A localização das modificações detalhadas nesta seção pode ser vista na Figura 5.10. Todas as notificações ficam sob o ramo *wmanIfBsTrapPrefix*. Os detalhes destas modificações podem ser observados no APÊNDICE D. As novas notificações podem ser observadas no APÊNDICE E.

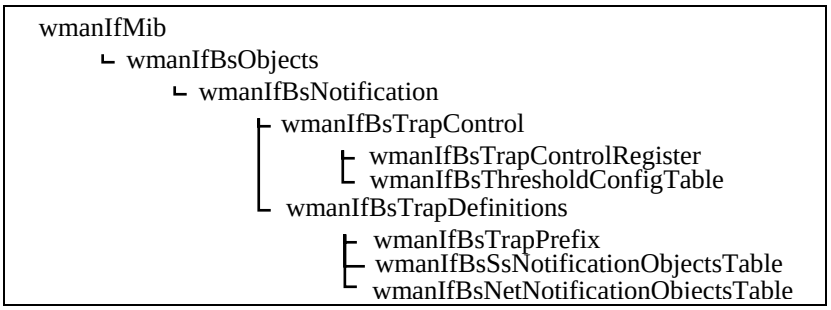


Figura 5.10: Objetos modificados para novas notificações

A notificação *wmanIfBsDynamicServiceSuccessTrap* reporta sucesso em operações de criação, alteração ou remoção de *Service Flows* que não ocorreram em função das configurações na tabela de *Service Flows* provisionados pelo sistema de gerenciamento (*wmanIfBsProvisionedSfTable*). Esta notificação faz parte da estratégia de aumentar o controle sobre os *Service Flows* da rede.

A notificação *wmanIfBsSsServiceAdmissionTrap* reporta sucesso ou falha na admissão pela rede dos *Service Flows* que foram provisionados pelo sistema de gerenciamento. Ela também pode indicar sucesso parcial, indicando que a admissão de determinado *Service Flow* se deu de forma degradada (ex. deveria ser reservado 512kbps, mas a estação base somente pode reservar 256kbps). Isto permite ao sistema de gerenciamento tanto contabilizar a utilização de reservas quanto tomar atitudes frente a problemas na admissão de novos fluxos provisionados.

A notificação *wmanIfBsSsConnectTrap* permite ao sistema de gerenciamento saber quando uma nova estação cliente conectou-se a rede ou quando uma estação cliente previamente conectada desconectou-se. Isto é importante, no caso deste trabalho, devido à estratégia de provisionamento dos *Service Flows* para uma estação cliente. De um modo geral, isto aumenta o controle do sistema de gerenciamento sobre quais os dispositivos conectados à rede.

A notificação *wmanIfBsSsHandoverTrap* permite ao sistema de gerenciamento saber quando uma estação cliente está efetuando um processo de *handover* (solicitou remoção de seu estado na estação base) ou se o mesmo foi cancelado (o padrão permite o arrependimento até certo limite de tempo). Esta notificação também é útil para fins de contabilização, já que uma estação em processo de *handover*, ainda que permaneça registrada (enquanto seu estado não for removido de fato), não consome recursos de rede.

A notificação *wmanIfBsSsBurstProfileChangeTrap* permite ao sistema de gerenciamento saber quando alguma estação cliente alterou seu perfil de transmissão ou recepção. A conseqüência desta alteração é que a taxa de transmissão ou recepção é alterada. Como foi mencionado anteriormente, o conhecimento sobre esta taxa pode ser útil ao sistema de gerenciamento na hora de resolver algum problema de degradação. Os novos valores de taxa são incluídos na notificação.

Por fim, as notificações *wmanIfBsSsServiceDegradationTrap* e *wmanIfBsNetDegradationTrap* servem para indicar situações de início de degradação, evolução de degradação e fim de degradação de QoS respectivamente em um *Service Flow* específico ou na rede como um todo. Como foi mencionado no levantamento de requisitos para a MIB, seria interessante que as estações base pudessem notificar o sistema de gerenciamento sobre este tipo de situação de forma que ele pudesse tomar

providências de acordo com a política da rede. Tentou-se, neste trabalho, incluir esta capacidade na MIB de gerenciamento através destas novas notificações.

A primeira alteração para a criação das novas notificações foi na tabela que controla quando gerar notificações (*wmanIfBsTrapControlRegister*). Em geral, há um bit para ativar ou desativar cada notificação. Entretanto, a notificação *wmanIfBsSsServiceAdmissionTrap* é controlada pelos bits *wmanIfBsSsServiceAdmissionSuccess* e *wmanIfBsSsServiceAdmissionFail*. Isto permite ao sistema de gerenciamento indicar em separado se é desejável receber notificações de sucesso ou falha na admissão de *Service Flows* provisionados. Cabe notar que a admissão em modo degradado é considerada simultaneamente um sucesso e uma falha, sendo gerada a notificação para estes casos sempre que pelo menos um dos bits esteja marcado. A admissão em modo degradado pode ser considerada um item opcional nas estações base, visto que o sistema de gerenciamento também pode buscar este tipo de alternativa via configuração.

Outra tabela alterada foi a tabela que controla os limiares para geração de notificações (*wmanIfBsThresholdConfigTable*). Esta tabela possuía os valores de limiares para a notificação sobre alterações no sinal de recepção. Foram incluídos limiares para a geração das notificações *wmanIfBsSsServiceDegradationTrap* e *wmanIfBsNetDegradationTrap*. Estes limiares incluem o nível de serviço mínimo para considerar o início de uma situação de degradação (nível mínimo aceitável), o passo mínimo para que uma estação base emita nova notificação de degradação e o tempo mínimo em situação normal (isto é, acima do nível mínimo aceitável) para que a estação base emita uma notificação de fim de degradação. Estes três limiares existem em separado para as duas notificações e são denominados como *StartThreshold*, *StepThreshold* e *HysteresisTime*.

A tabela *wmanIfBsSsNotificationObjectsTable*, já existente na MIB original, serve para agrupar dados que são enviados em notificações relativas a estações cliente. O índice desta tabela inclui o endereço MAC da estação a que os dados de uma linha específica se referem. Outra modificação efetuada foi incluir dados de novas notificações nesta tabela.

O campo *wmanIfBsDynamicServiceSfId*, incluído na notificação *wmanIfBsDynamicServiceSuccessTrap*, indica qual o SFID ao qual a notificação se refere. Os campos *wmanIfBsServiceAdmissionSfId*, *wmanIfBsServiceAdmissionStatus* e *wmanIfBsServiceAdmissionFailReason*, incluídos na notificação *wmanIfBsSsServiceAdmissionTrap*, foram criados para manter, respectivamente, o SFID, status de admissão (admitido, admitido em modo degradado ou não admitido) e razão de falha (caso uma tenha ocorrido). O campo *wmanIfBsSsConnectStatus* indica se a notificação *wmanIfBsSsConnectTrap* se refere à conexão ou desconexão de uma estação cliente. O campo *wmanIfBsSsHandoverStatus* indica se a notificação *wmanIfBsSsHandoverTrap* se refere ao início de um *handover* ou ao seu cancelamento. Os campos *wmanIfBsServiceDegradationSfId*, *wmanIfBsServiceDegradationStatus* e *wmanIfBsServiceDegradationUsefulPercentage*, incluídos na notificação *wmanIfBsSsServiceDegradationTrap*, foram criados para manter, respectivamente, o SFID, o status de degradação sendo reportado (início ou fim) e o nível de serviço sendo entregue pela rede ao fluxo em questão.

Como a notificação *wmanIfBsNetDegradationTrap* se refere à rede como um todo e não a uma estação cliente específica, os dados para esta notificação foram incluídos em uma nova tabela criada para este fim: *wmanIfBsNetNotificationObjectsTable*. Foram incluídos nesta tabela os campos *wmanIfBsNetDegradationStatus* e *wmanIfBsNetDegradationUsefulPercentage*, que representam, respectivamente, o status de degradação sendo reportado (início ou fim) e o nível de serviço sendo entregue pela rede em geral.

5.3 Detalhes operacionais da MIB e comparação com requisitos

A idéia desta seção é apresentar como o sistema de gerenciamento pode utilizar os recursos presentes na MIB definida e de que forma os requisitos levantados para a mesma são atingidos pelas novas definições.

5.3.1 Operação Geral da MIB

Antes de configurar uma dada estação base de acordo com as políticas da rede, convém que o sistema de gerenciamento ajuste as configurações nas tabelas *wmanIfBsConfigurationTable* e *wmanIfBsThresholdConfigTable* para os valores apropriados para sua operação. Também convém indicar quais as notificações que espera receber através dos objetos *wmanIfBsTrapControlRegister* e *wmanIfBsStatusTrapControlRegister*.

Configurado o modo de operação para a estação base, o sistema de gerenciamento pode descobrir quais as estações cliente interagindo com a estação base obtendo os registros das tabelas *wmanIfBsConnectedSsTable* e *wmanIfBsRegisteredSsTable*. Ele também pode descobrir quais os fluxos existentes na rede e seus parâmetros de QoS através da tabela *wmanIfCmnCpsServiceFlowTable*. As tabelas *wmanIfCmnClassifierRuleTable* e *wmanIfCmnPhsRuleTable* apresentam informações adicionais sobre os fluxos e também podem vir a serem observadas.

Tendo obtido o estado atual da rede, o sistema de gerenciamento deve buscar configurar a rede de acordo com a política definida para a mesma. Isto é feito através das tabelas *wmanIfBsProvisionedSfTable*, *wmanIfBsSsProvisionedForSfTable*, *wmanIfBsServiceClassTable* e *wmanIfBsClassifierRuleTable*. Pode ser necessário remover algum fluxo existente, o que é feito diretamente na tabela *wmanIfCmnCpsServiceFlowTable* no caso de fluxos não provisionados, e através das tabelas *wmanIfBsProvisionedSfTable* e *wmanIfBsSsProvisionedForSfTable* no caso de fluxos provisionados (basta remover a linha referente ao fluxo, neste último caso). Se houver a necessidade de modificar algum fluxo não provisionado pode-se marcar o mesmo como provisionado e gerenciá-lo diretamente através tabelas de gerência de fluxos provisionados. Cabe notar que os fluxos provisionados provavelmente foram configurados pelo próprio sistema de gerenciamento em um momento anterior, de forma que pode ser importante que o mesmo tenha alguma memória sobre o que lhe representa cada SFID.

Depois de configurado o estado inicial de acordo com a política de rede, o sistema de gerenciamento passa a operar basicamente em um modo reativo. Isto é, passa a tomar atitudes frente às notificações recebidas das estações base. Claro, eventualmente, se a política de rede ditar, pode ser necessário modificar a configuração da rede mesmo sem

que notificações tenham sido recebidas (por exemplo, alguma configuração de fluxo dependente de horário). De todo modo, abaixo será ilustrado como se daria a operação com base nas notificações.

Ao receber uma notificação do tipo *wmanIfBsSsConnectTrap*, o sistema de gerenciamento pode, dependendo se o que ocorreu foi a conexão ou desconexão de uma estação cliente, configurar ou desconfigurar os *Service Flows* referentes àquela estação cliente.

Um *Service Flow* é configurado nas tabelas *wmanIfBsProvisionedSfTable* e *wmanIfBsSsProvisionedForSfTable* e pode ser marcado para permanecer em três estados de acordo com o campo *wmanIfBsSfState*: autorizado, admitido ou ativo. Autorizado significa que o SF existe, mas não há reservas para o mesmo. Admitido significa que o SF possui recursos reservados para si, mas não está ativamente enviando pacotes. Ativo, por sua vez, significa que o SF possui recursos reservados e que a estação base está atendendo ativamente este fluxo. Apenas SFs admitidos e ativos possuem um identificador de conexão alocado para si.

Apesar de existir nas tabelas *wmanIfBsProvisionedSfTable* e *wmanIfBsSsProvisionedForSfTable*, um SF não existe realmente na rede enquanto a estação cliente dona de tal fluxo não se registrar no setor em questão com a estação base. Desta forma, pode-se considerar que estas tabelas representem apenas um estado administrativo desejado, sendo que o estado real de um fluxo somente pode ser observado na tabela *wmanIfCmnCpsServiceFlowTable*. Isto atinge dois requisitos para o gerenciamento mencionados anteriormente: é possível configurar os SFs na rede mesmo que as estações cliente não estejam registradas (possibilidade de configuração antecipada para um *handover*) e o controle de admissão dos fluxos provisionados (cujos recursos já foram liberados pelo sistema de gerenciamento) pode ser feito pelas estações base no momento em que a estação cliente se registra no setor em questão.

Embora o controle de admissão possa ser feito pela estação base, o sistema de gerenciamento necessita saber caso algum SF provisionado não tenha sido admitido de modo a poder tomar providências. Por este motivo é que foi criada a notificação *wmanIfBsSsServiceAdmissionTrap*. Esta notificação permite ao sistema de gerenciamento saber se algum SF provisionado não foi admitido ou foi admitido em modo degradado. Neste último caso, os parâmetros de QoS que foram admitidos poderiam ser buscados na tabela *wmanIfCmnCpsServiceFlowTable*. Como o sistema de gerenciamento possui total controle sobre os *Service Flows* (mesmo os não provisionados podem ser tornados gerenciáveis), ele possui total flexibilidade para tentar resolver o problema de acordo com a política de rede, o que cumpre mais um dos requisitos estabelecidos.

Para controlar a localização de uma dada estação cliente, as notificações *wmanIfBsSsRegisterTrap* e *wmanIfBsSsHandoverTrap* permitem saber quando uma estação cliente entrou ou saiu da região de abrangência de dada estação base. Todos os eventos relevantes sobre o estado de uma estação cliente podem ser capturados através do conjunto de notificações *wmanIfBsSsStatusNotificationTrap*, *wmanIfBsSsConnectTrap*, *wmanIfBsSsRegisterTrap* e *wmanIfBsSsHandoverTrap*, de modo que mais um dos requisitos de gerenciamento foi atendido. A entrada no modo *idle* é marcada por uma operação de desregistro (já contemplada) e, por não ser

relevante no contexto de gerenciamento de QoS, não se chegou a criar notificações e tabelas para registrar o modo *sleep* de estações cliente.

As duas abordagens de tratamento para situações em que estações cliente solicitam recursos via mensagens DSA ou DSC foram contempladas. A abordagem de negar sempre os recursos pode ser configurada através do parâmetro *wmanIfBsSsSfMgmtEnabled* da tabela *wmanIfBsConfigurationTable*. Caso seja permitida a aceitação destas mensagens pela estação base, a notificação *wmanIfBsSsDynamicServiceSuccesTrap* permite ao sistema de gerenciamento ficar sabendo que alguma destas mensagens foi aceita e qual o *Service Flow* criado, modificado ou removido por ela. Mais uma vez, os parâmetros de QoS ativos para o *Service Flow* alvo destas mensagens pode ser observado na tabela *wmanIfCmnCpsServiceFlowTable*.

Por fim, diversas notificações permitem ao sistema de gerenciamento registrar as mais diversas situações de rede. Podem ser casos como início de consumo de recursos através da admissão de um SF provisionado (*wmanIfBsSsServiceAdmissionTrap*) ou ocorrência de algum problema ou limitação na rede (*wmanIfBsSsDynamicServiceFailTrap*).

5.3.2 Tratamento de Situações de Degradação

Como foi visto anteriormente, foram criadas duas notificações que identificam situações de degradação distintas na rede. Estas notificações foram pensadas de forma que contivessem a informação necessária para que o sistema de gerenciamento pudesse tentar resolver o problema. Nesta seção será feito um estudo sobre como isto pode ser feito.

A notificação *wmanIfBsSsServiceDegradationTrap* é emitida para cada *Service Flow* degradado, indicando que o *Service Flow* específico não está recebendo a QoS acordada. A idéia desta notificação é fornecer uma granularidade mais fina sobre quais fluxos estão sofrendo degradação, possibilitando que o sistema de gerenciamento, de acordo com a política da rede, tente resolver o problema de cada fluxo individualmente através de alguma redistribuição de recursos.

A notificação *wmanIfBsNetDegradationTrap* se refere a uma situação de degradação mais generalizada, que afeta todas as estações. O objetivo é discernir entre casos como o que uma dada estação se distancia da região de acesso (e por conta disto tem seus fluxos degradados), dos casos como o que o início de uma chuva começa a afetar todas as estações. No primeiro caso, identificável pela recepção apenas da notificação *wmanIfBsSsServiceDegradationTrap*, não adianta redistribuir recursos entre todas as estações, pois a degradação é por motivo intrínseco à estação dona do fluxo sendo degradado. Caso o sistema de gerenciamento receba a notificação *wmanIfBsNetDegradationTrap*, ele saberá que está diante de uma situação de degradação generalizada e poderá redistribuir recursos de forma a priorizar fluxos que a política da rede definiu como mais importantes. Este trabalho não faz uma análise sobre como a estação base poderá identificar um ou outro caso de degradação, mas fica visível que o sistema de gerenciamento deve comportar-se de forma distinta frente a um ou outro caso.

Além de informar sobre a existência da degradação, ambas as notificações incluem um percentual que representa o nível de serviço sendo entregue. Este percentual tenta

passar ao sistema de gerenciamento a quantidade de recursos extras que seriam necessários para que não houvesse mais degradação. Ele pode ser utilizado pelo sistema de gerenciamento para traçar a melhor estratégia para combater o problema da degradação de acordo com a política da rede. Também é com base neste percentual que se pode configurar quando exatamente as notificações de degradação devem ser emitidas, de forma que não se tente solucionar algo que não é considerado um problema pela política da rede.

Basicamente o recurso que se considera para melhorar o serviço sendo entregue a um determinado *Service Flow* é tempo. Se todo o tempo fosse dedicado a uma única estação cliente ela teria vazão equivalente a sua taxa de transmissão – obviamente isto não deve ocorrer, pois o meio é compartilhado e todos devem ter oportunidade de transmissão. O que se deve descobrir, então, é quanto tempo está sendo alocado para dado fluxo e quanto tempo seria necessário para que não houvesse degradação. Duas análises serão feitas abaixo. O foco foi dado em reserva de banda, mas, concedendo-se mais tempo de rede a determinado fluxo (isto é, reduzindo os recursos reservados aos demais fluxos), problemas como atraso também podem ser resolvidos. Não se considera perda aqui porque as classes de QoS nas redes IEEE802.16 não permitem a definição de limites de perda – embora permitam a ativação de ARQ.

Considere-se o caso de um *Service Flow* que possui reserva de U bit/s para upload. Considere-se, ainda, que a taxa de transmissão em upload atingível pela estação cliente dona deste fluxo é de T_U bit/s. Isto significa que uma razão U/T_U do tempo de quadro seria suficiente para garantir a reserva deste fluxo (assumindo que não haja perdas). Se a notificação *wmanIfBsSsServiceDegradationTrap* indicar que o nível de serviço sendo entregue a este fluxo é de 90%, então significa que apenas 90% do tempo que seria necessário está sendo entregue a ele. O restante pode eventualmente ser obtido degradando reservas de outros fluxos da mesma estação (conforme a política de rede) de modo a liberar o tempo necessário para este fluxo – o cálculo do tempo consumido pelos outros fluxos pode ser feito da mesma forma.

Outro caso é aquele em que o sistema de gerenciamento recebe uma notificação do tipo *wmanIfBsNetDegradationTrap*. Seja D a representação para taxa de download reservada para um fluxo e T_D a representação para a taxa de recepção em download da estação cliente dona do fluxo de taxa D. Pensando-se em uma operação TDD, deve-se somar as proporções U/T_U e D/T_D para todos os fluxos existentes na rede. Isto representaria a proporção do tempo de quadro que seria necessária para atender a todas as reservas aceitas pela rede. Não seria inesperado obter um valor superior a 1, ainda mais considerando-se uma situação de degradação de rede (em que provavelmente os valores de T_U e T_D tenham sofrido alguma queda). De todo modo, um nível de serviço de 90% (indicado na notificação) indicaria que é necessário reduzir em 10% o "consumo de tempo" dos fluxos ativos na rede para que a rede não considere mais a existência de degradação. Note que não é possível atender a todos os fluxos. Como foi mencionado em um capítulo anterior, o que o sistema de gerenciamento pode fazer é escolher quais fluxos devem ser degradados primeiro para manter o nível de serviço dos fluxos que a política de rede julgar mais críticos. A forma como são escolhidos os fluxos depende do sistema de gerenciamento. Uma abordagem para tratar esta questão já foi mencionada na seção "Gerenciamento de QoS baseado em políticas".

As taxas de *upload* e *download* para cada estação cliente podem ser obtidas na tabela *wmanIfBsRegisteredSsTable*, através dos campos *wmanIfBsSsMaxUplinkTransmission* e *wmanIfBsSsMxDownlinkTransmission* respectivamente. Para o controle de alterações nas taxas de transmissão foi criada a notificação *wmanIfBsSsBurstProfileChangeTrap*, permitindo ao sistema de gerenciamento manter-se atualizado sobre a capacidade de transmissão e recepção de cada estação cliente.

6 PROTÓTIPO IMPLEMENTADO

A implementação da proposta de gerenciamento alvo desta dissertação se deu em duas fases: primeiramente foi implementado um simulador da MIB definida no capítulo "MIB PARA GERENCIAMENTO"; em seguida, foi criado um PDP que suportasse o gerenciamento de redes IEEE802.16 através da MIB definida. O PDP foi criado para integrar-se ao sistema de gerenciamento QAME (*QoS-Aware Management Environment*) (MARQUEZAN et al., 2005; GRANVILLE et al., 2001a; GRANVILLE et al., 2001b), que acabou também sendo modificado para incluir funções necessárias ao gerenciamento de redes IEEE802.16.

Os detalhes da implementação do simulador foram incluídos no APÊNDICE F. As seções que seguem abordarão os seguintes assuntos: uma visão geral do sistema de gerenciamento QAME e o motivo de sua escolha; as extensões que tiveram de ser feitas no ambiente QAME, bem como a forma como o PDP reage às ações neste ambiente; alguns cenários de utilização do QAME; detalhes da implementação do PDP; e algumas conclusões obtidas durante as implementações.

6.1 Visão geral do sistema de gerenciamento QAME

O QAME (MARQUEZAN et al., 2005; GRANVILLE et al., 2001a; GRANVILLE et al., 2001b) é um sistema *web* de gerenciamento baseado em políticas que segue o modelo do IETF (YAVATKAR, PENDARAKIS e GUERIN, 2000) e opera através de quatro conceitos: fluxos (*flows*), ações (*actions*), temporizadores (*timers*) e políticas (*policies*). A Figura 6.1 ilustra a tela de entrada do sistema.

Um fluxo no sistema QAME representa um conjunto de filtros conjuntivos que permitem identificar pacotes sendo trafegados na rede. Este conjunto de filtros pode incluir endereços IP de origem e destino, portas de origem e destino, marcações nos pacotes utilizando DSCP, entre outros atributos de pacotes. Naturalmente que a especificação de mais de um valor para o mesmo atributo é uma condição disjuntiva (como filtrar por dois IPs de destino). O formulário de cadastro de fluxos do QAME pode ser visto na Figura 6.2.

Uma ação no sistema QAME permite especificar o tratamento a ser dado a um determinado fluxo. Uma ação pode indicar tratamentos como reserva de banda, limitação de latência, priorização, marcação de pacotes, entre outras, de modo que requisitos de QoS são especificados através de ações. O formulário de cadastro de ações do QAME pode ser visto na Figura 6.3.

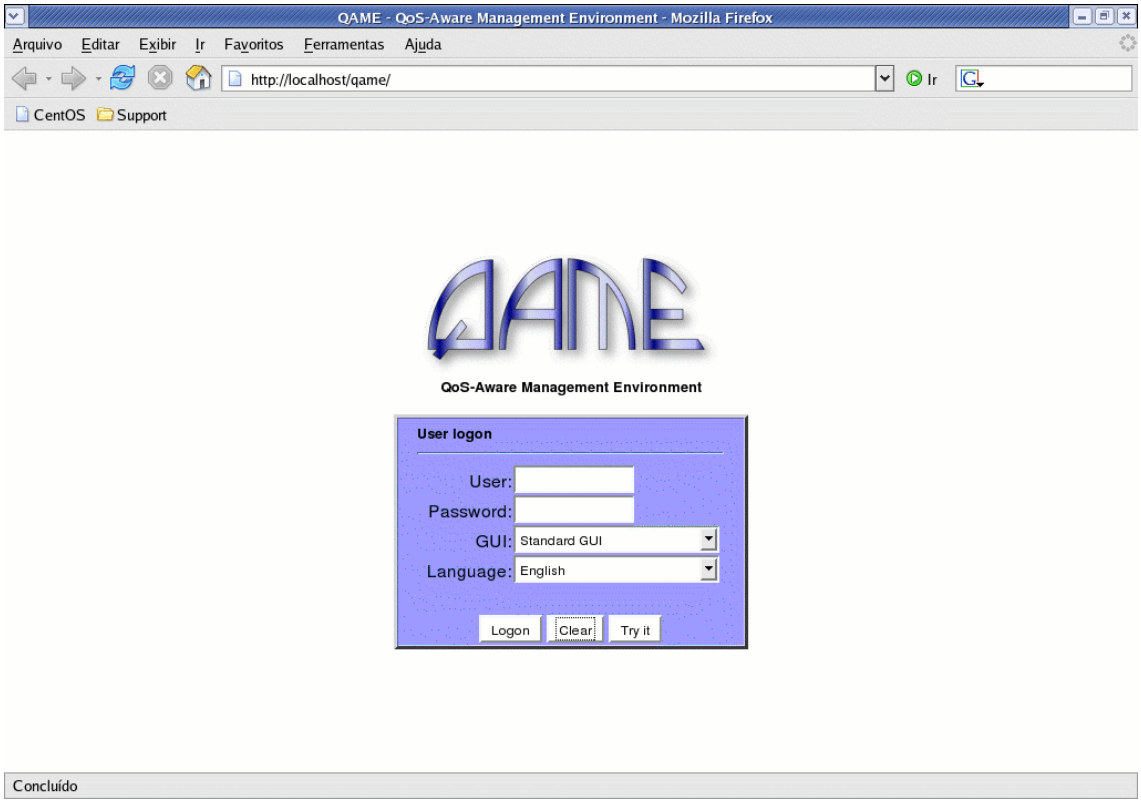


Figura 6.1: Tela inicial do QAME

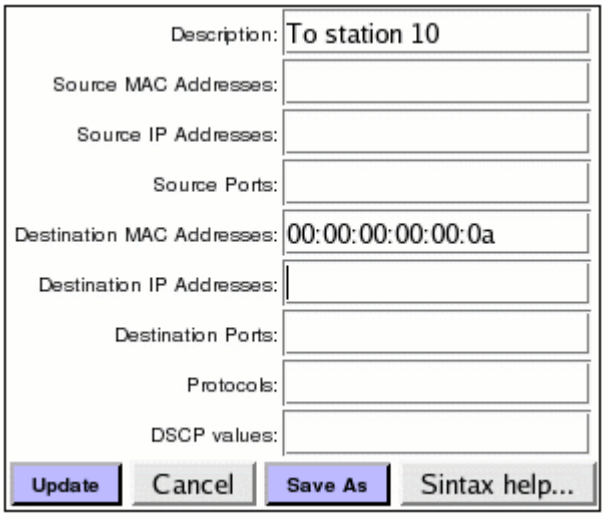


Figura 6.2: Formulário de cadastro de fluxos do QAME

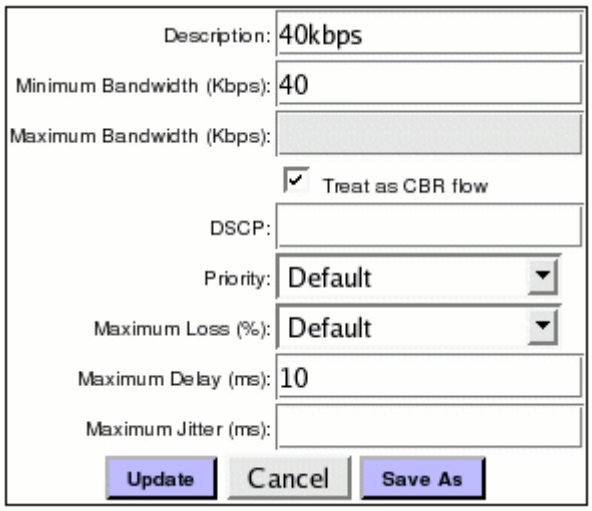


Figura 6.3: Formulário de cadastro de ações do QAME

Um temporizador no sistema QAME representa um conjunto de períodos no qual determinada política deve estar ativa. Um exemplo seria um temporizador que indicasse a ativação de políticas de segunda à sexta durante o horário comercial. De fato, o QAME permite especificar horários de uma forma bastante completa, incluindo a indicação de dias da semana ou do mês, meses do ano, horários dentro do dia e tudo com diversas possibilidades de combinações. Há também espaço para indicar o período de validade de uma política. O formulário de cadastro de temporizadores do QAME pode ser visto na Figura 6.4.

Utilizando os conceitos apresentados anteriormente, o QAME constrói seu principal conceito: o de política. Uma política no sistema QAME reúne um conjunto de fluxos, um conjunto de temporizadores e uma ação. O significado de uma política é que os pacotes identificados através dos fluxos deverão perceber o tratamento especificado na ação durante os períodos indicados pelos temporizadores. O formulário de cadastro de políticas do QAME pode ser visto na Figura 6.5. Cabe notar que, embora a interface sugira que pode haver mais de uma ação na mesma política, o sistema restringe a somente uma ação. O QAME também dispõe de uma interface alternativa na forma de um *wizard*, o que facilita a criação dos elementos que irão compor uma política durante a criação da mesma.

Após a definição de uma política, o gerente da rede deve especificar em quais PEPs a política deverá ser aplicada. Para facilitar esta tarefa, o QAME dispõe de um recurso de mapa de rede (Figura 6.6) que permite ao gerente observar a rede de uma forma gráfica. O mapa de rede é hierarquizado, de modo que mais detalhes podem ser vistos ao adentrar-se em uma rede específica (a Figura 6.7 ilustra a rede ArgentonNet). É através do mapa de rede que o gerente pode cadastrar ou remover nodos a serem gerenciados, indicar as capacidades de cada nodo (como PEP ou PDP) e aplicar ou remover políticas.

Para um nodo que possua capacidade de gerenciamento via SNMP, entrando na tela que exibe suas propriedades a partir do mapa de rede (Figura 6.8), é possível configurar a string de comunidade SNMP que deverá ser utilizada para acessá-lo. Também é possível configurar qual o PDP que ficará responsável pelo seu gerenciamento e realizar a ativação ou desativação de políticas, entre outras atividades. Naturalmente que, no

contexto desta dissertação, o PDP associado às estações base foi o PDP criado com capacidade de gerenciamento de redes IEEE802.16. No QAME, as políticas são aplicadas nas interfaces de rede dos PEPs, como é ilustrado no formulário de aplicação de políticas (Figura 6.9). Na Figura 6.9, as políticas são aplicadas na interface de *loopback* porque o simulador implementado utiliza esta interface como se fosse a interface IEEE802.16 da estação base.

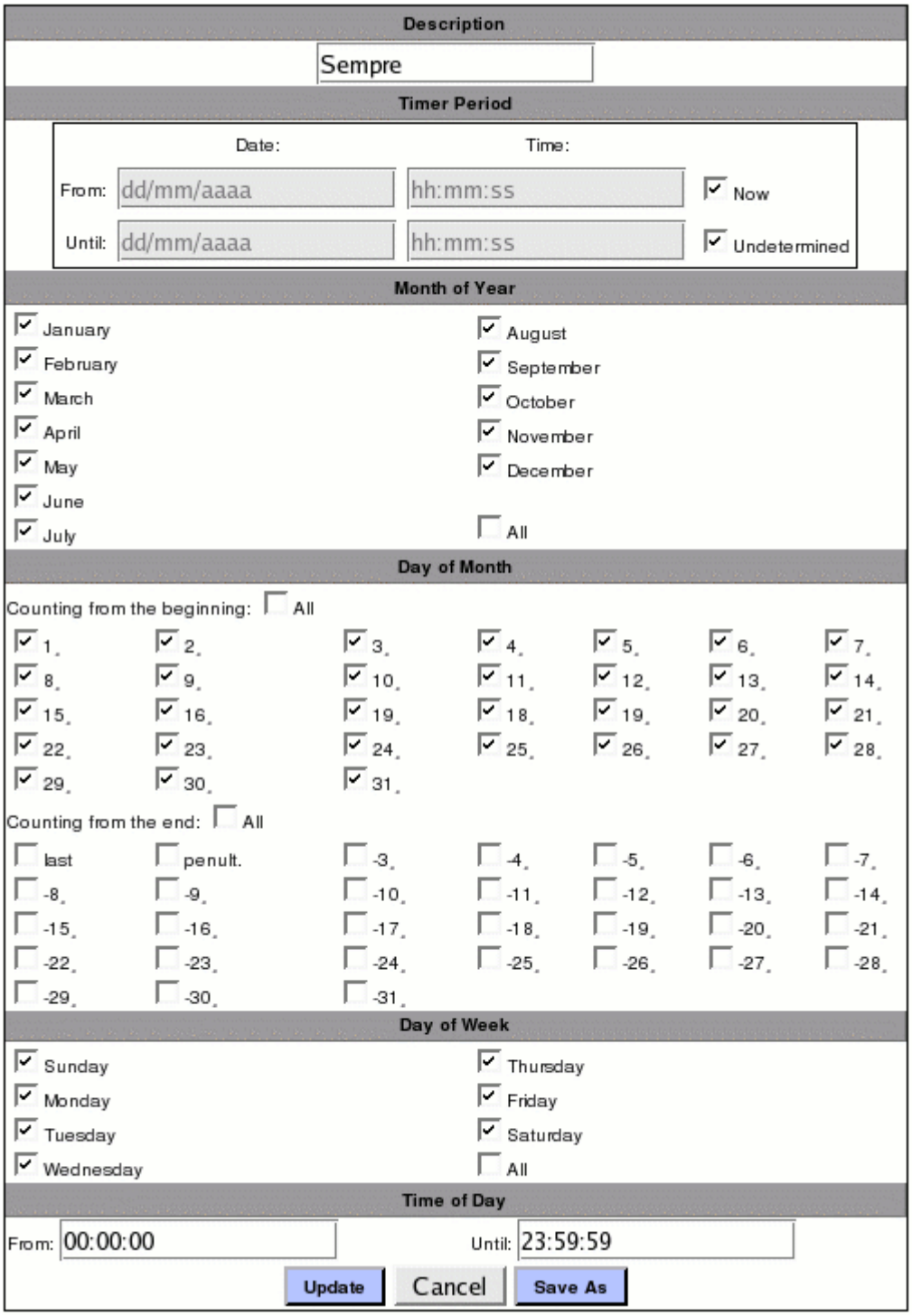


Figura 6.4: Formulário de cadastro de temporizadores do QAME

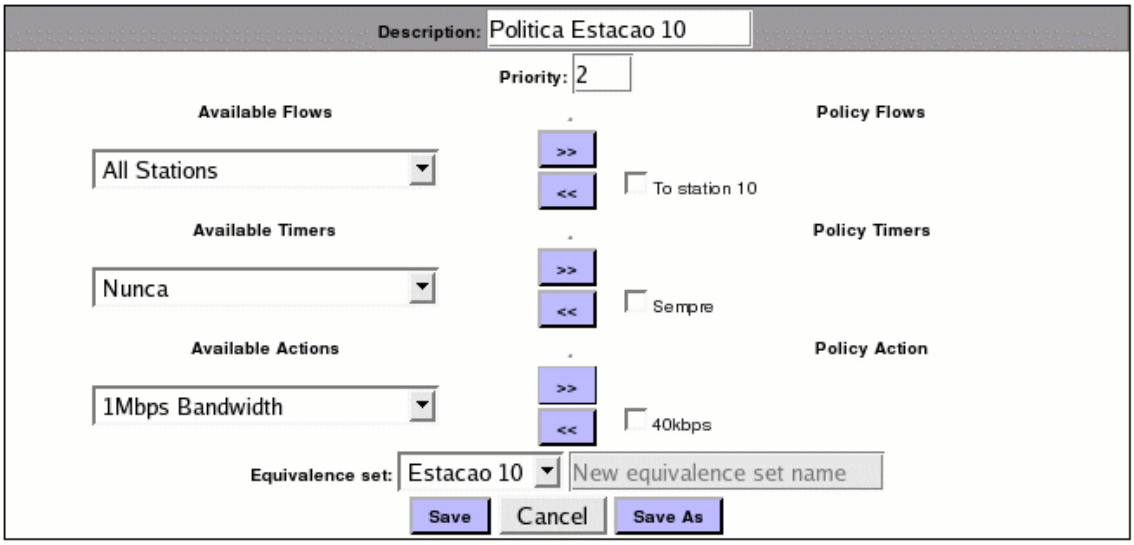


Figura 6.5: Formulário de cadastro de políticas do QAME

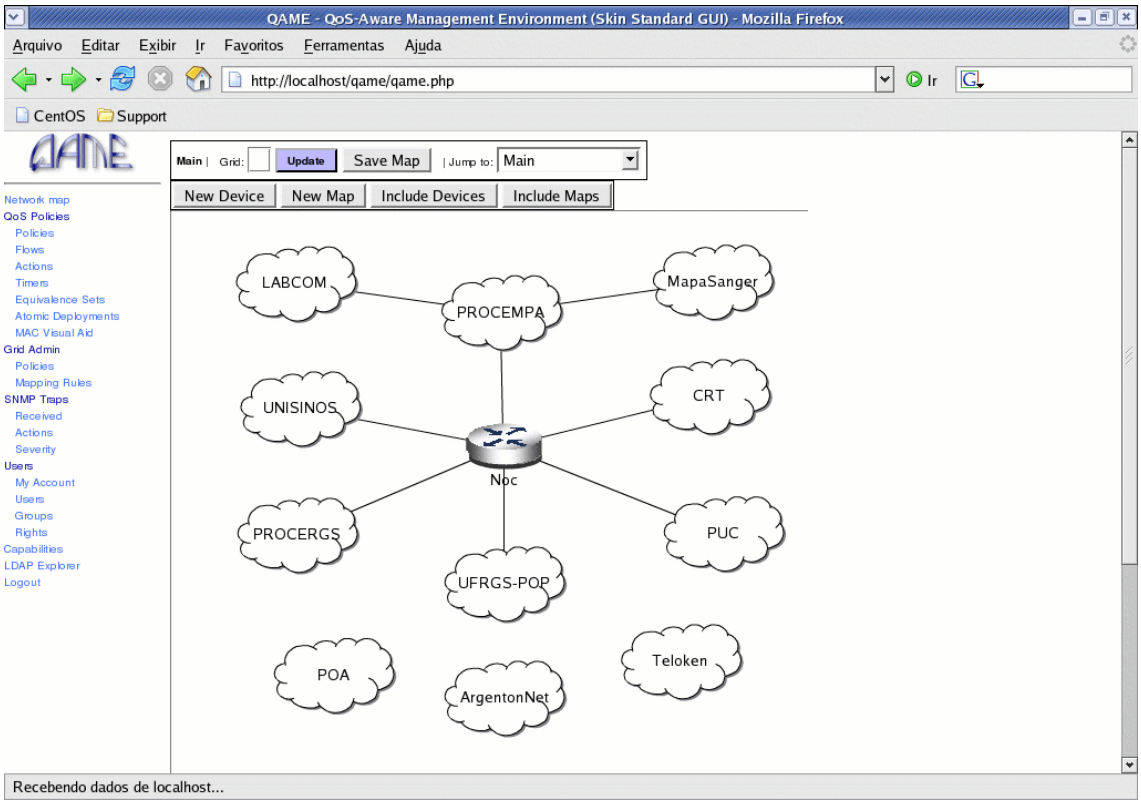


Figura 6.6: Mapa de rede principal do QAME

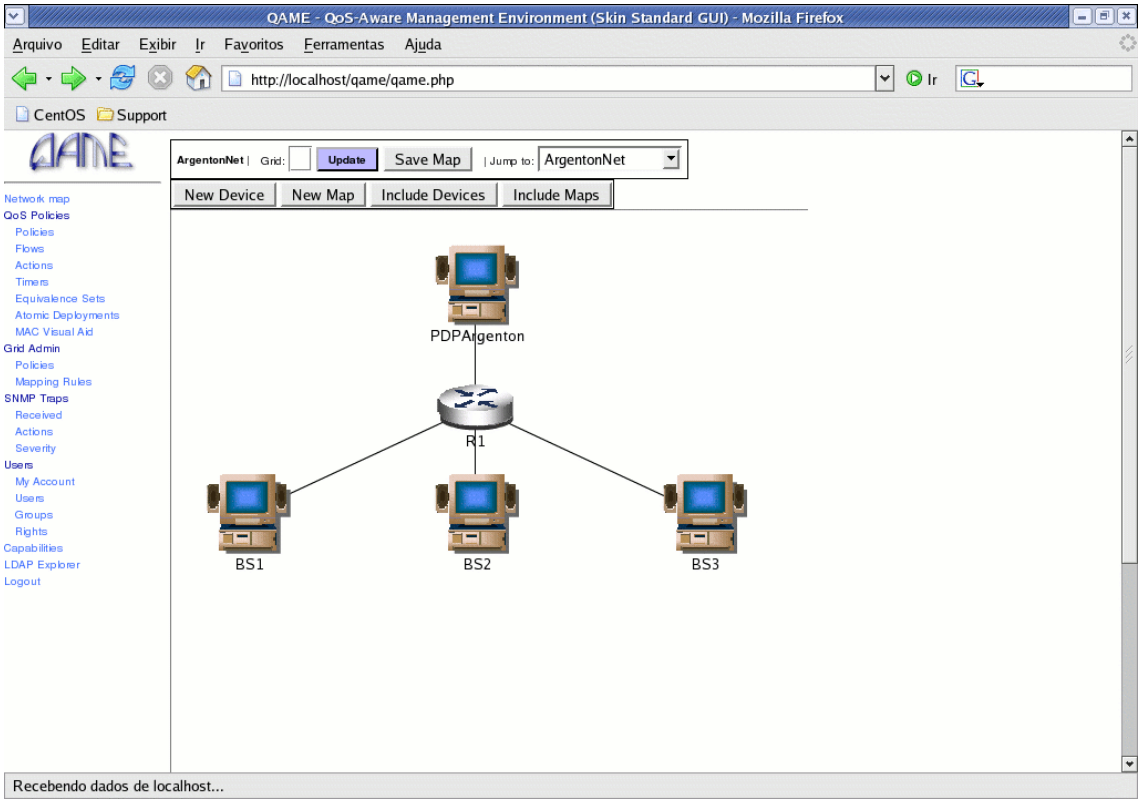


Figura 6.7: Mapa da rede ArgentonNet

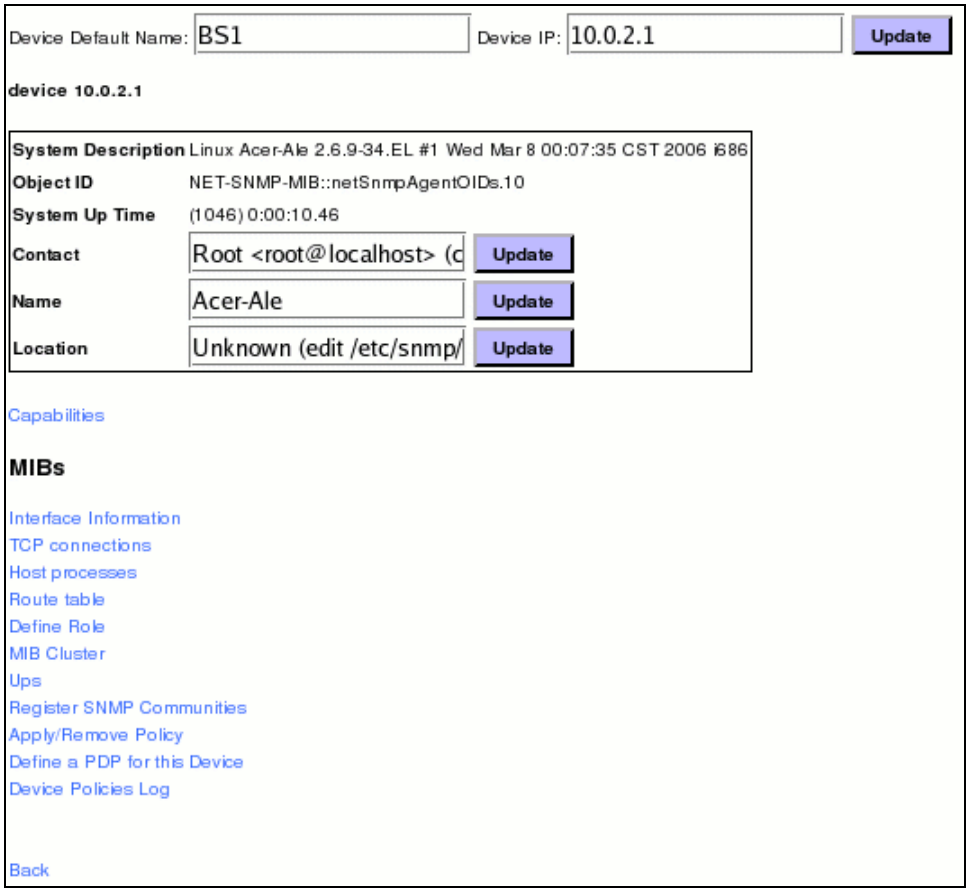


Figura 6.8: Propriedades do nodo BS1

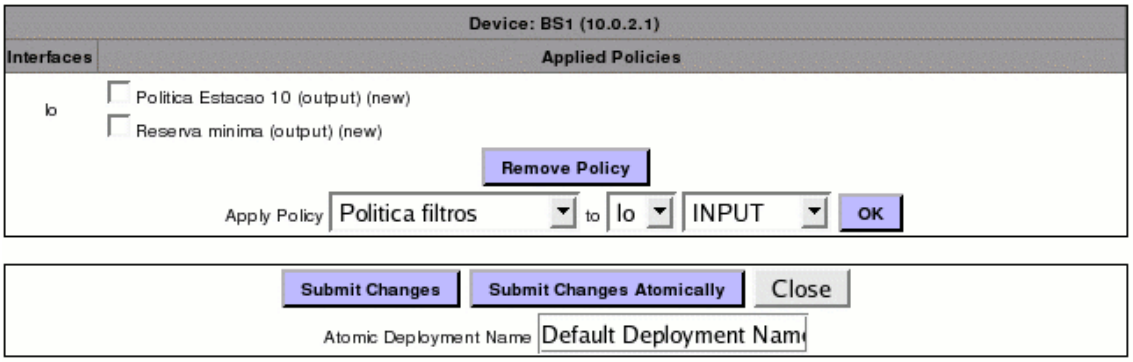


Figura 6.9: Formulário de aplicação de políticas do QAME

A comunicação entre a estação de gerenciamento e os PDPs no âmbito do sistema QAME é feita utilizando-se o protocolo SOAP (GUDGIN et al., 2007) sobre HTTP (FIELDING et al., 1999), em um modelo de Web Services (CURBERA et al., 2002). A utilização deste modelo de comunicação é muito vantajosa neste contexto, pois o protocolo HTTP é *firewall-friendly* e facilita que o PDP e a estação de gerenciamento estejam em redes distintas. O posicionamento ideal para o PDP é próximo às estações (PEPs) que ele irá gerenciar, até porque o protocolo utilizado nesta comunicação pode ser diverso e não passar tão facilmente por *firewalls* (o protocolo SNMP tipicamente é barrado).

O QAME foi desenvolvido em PHP4 e possui suporte a internacionalização. O modelo de desenvolvimento *web* permite que o gerente acesse o sistema de qualquer lugar, caso este esteja disponibilizado na Internet.

Apesar de ser um sistema pré-implementado, nem sempre se tem a solução mais adequada ao utilizar o QAME para gerenciar QoS em redes IEEE802.16, ou gerenciar QoS de modo geral. Abaixo são colocados alguns pontos em que o QAME se mostra deficiente neste contexto.

- O QAME exige que os PEPs a serem configurados sejam explicitamente apontados. Embora este fato não comprometa o gerenciamento da migração das estações na parte móvel da rede, visto que as estações base não alocarão recursos para estações móveis não registradas, o gerenciamento da migração na parte fixa da rede é inviabilizado, pois o sistema de gerenciamento não é capaz de selecionar os dispositivos a serem configurados com base na rota dos fluxos de interesse. Perde-se também a garantia de consistência da configuração de QoS da parte fixa da rede com a parte móvel, visto que o sistema de gerenciamento as trata de forma independente. Contudo, o gerenciamento da parte móvel é suficiente para validar a utilização de gerenciamento baseado em políticas em redes IEEE802.16. Cabe notar que a questão da diferenciação de políticas de acordo com a localização das estações móveis pode ser adequadamente tratada utilizando-se ações diferentes em cada estação base.
- O QAME não permite agrupar as estações base em regiões, visto que as políticas de uma estação base devem ser configuradas explicitamente na mesma.
- O QAME não implementa o conceito de usuário da rede (e, logicamente, também não implementa grupos de usuários da rede). Os usuários da rede devem, neste caso, ser identificados por filtros nos fluxos vinculados às estações

móveis. Como já foi mencionado anteriormente, a identificação de estações móveis IEEE802.16 deve ser feita com base em seu endereço MAC. Vale notar que, apesar de não permitir filtros baseados nos usuários da rede, o QAME utiliza controle de acesso baseado em usuários.

- O QAME não permite agrupar filtros de forma disjuntiva, exceto na especificação da própria política (através da inclusão de mais de um fluxo na mesma política). O agrupamento de requisitos de QoS pode ser feito em ações, e o agrupamento de horários pode ser feito majoritariamente através de temporizadores. Entretanto, em alguns casos, pode ser necessário agrupar temporizadores nas próprias políticas para produzir determinados agrupamentos de horários.
- O QAME não permite definir o tratamento a ser dado em situações de degradação de rede. A não implementação de tal mecanismo impede que se possa validar a proposta de tratamento de degradação sugerida anteriormente. A implementação deste mecanismo, da forma como foi proposto, exigiria uma alteração estrutural mais profunda no QAME, e optou-se por não realizá-la.
- O QAME não permitia tratar conflitos entre políticas. Após as extensões realizadas, que serão descritas na seção seguinte, foi implementado um mecanismo de prioridades de aplicação de políticas. Não foram implementadas prioridades de definição de políticas.
- O QAME não permite avaliar a condição da rede no filtro de uma política.
- O QAME funciona apenas em modo *provisioning*, não permitindo a definição de políticas de autorização. Entretanto, esta capacidade não é importante para a realização deste trabalho em função do modelo de gerenciamento adotado.
- O QAME não permite definir parâmetros gerais de operação da rede, como reserva mínima para fluxos da classe melhor esforço.

Embora o QAME apresente todos os problemas mencionados acima, nenhum destes problemas inviabiliza a validação da utilização de gerenciamento baseado em políticas em redes IEEE802.16, que passou a ser o foco desta dissertação a partir do momento em que se optou por este modelo de gerenciamento.

O QAME foi escolhido como base para a implementação por ser um sistema de gerenciamento baseado em políticas pré-implementado e mantido pelo grupo de redes da UFRGS. O uso de um sistema pré-implementado permite identificar até que ponto um sistema de gerenciamento baseado em políticas consegue ser genérico, além de contribuir com a redução do esforço de implementação e aproveitamento de recursos que eventualmente não seriam implementados. Ao utilizar o sistema mantido pelo grupo de redes da UFRGS, este trabalho também contribui com sua extensão e aprimoramento, agregando valor a um sistema que poderá também ser utilizado em futuras pesquisas.

6.2 Extensões realizadas no QAME e comportamento do PDP

Embora o QAME seja um sistema de gerenciamento baseado em políticas, algumas modificações tiveram de ser feitas para que o mesmo pudesse gerenciar redes

IEEE802.16. Outras modificações foram feitas visando uma maior facilidade de operação.

A modificação mais significativa foi a inclusão da possibilidade de filtrar por endereços MAC nos fluxos (Figura 6.2). Como foi mencionado anteriormente, em redes IEEE802.16, as estações clientes devem ser identificadas por MAC. Sem este filtro seria impossível indicar que determinado fluxo refere-se a uma dada estação cliente. No caso de um fluxo sem filtro de MAC pertencer a uma política sendo aplicada na rede IEEE802.16, o PDP considerará que este fluxo vale para todas as estações clientes.

O PDP implementado faz distinção entre dois tipos de políticas com base nos endereços IP de destino configurados nos fluxos: políticas *unicast* e políticas *multicast*. Uma política é dita *multicast* quando seus fluxos possuem filtro por endereços *multicast* de destino. Nas demais situações, a política é dita *unicast*. O tipo de política definirá, posteriormente, o tipo dos *Service Flows* que serão configurados na rede: *unicast* ou *multicast*. A arquitetura do QAME permite aos PDPs rejeitar políticas que não possam ser implementadas em sua tecnologia alvo. No caso do PDP implementado, ele informará erro quando uma parte dos fluxos da política for *multicast* e outra parte for *unicast*.

Na aplicação de políticas *unicast*, um *Service Flow* é configurado para cada estação cliente relacionada à política (contanto que a estação esteja conectada à estação base). Consideram-se estações relacionadas à política todas as estações para as quais há algum fluxo que filtre pelo seu endereço MAC. A existência de fluxos sem filtro por MAC indica que todas as estações clientes estão relacionadas à política. A Figura 6.10 ilustra uma situação em que as estações SS1 e SS3 estão relacionadas a uma dada política, sendo criado um *Service Flow* para cada uma delas, e a estação SS2, embora esteja comunicando-se com a estação base, não está relacionada a política em questão.

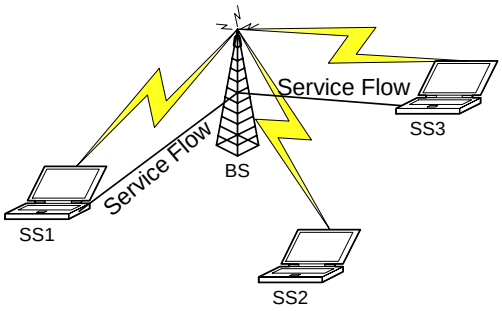


Figura 6.10: *Service Flows unicast*

Embora todas as estações relacionadas a um política *unicast* venham a ter um *Service Flow* configurado para si, as *Classifier Rules* (filtros) do *Service Flow* de uma estação somente irão considerar os fluxos que filtram pelo MAC da estação e os fluxos que não possuem filtro de MAC. Neste sentido, uma política pode apresentar filtros distintos para cada estação cliente a que ela se aplica. Os itens que sempre valerão para todas as estações em uma política são os períodos em que a mesma estará ativa e as ações (requisitos de QoS) sobre os pacotes.

Uma diferença da aplicação de uma política *unicast* na rede IEEE802.16 para a aplicação da mesma política em outra tecnologia de rede é que, ao criar um *Service Flow* para cada estação associada a política, o PDP acaba efetuando mais de uma reserva. Supondo que a ação garanta 100kbit/s de vazão mínima para os fluxos, cada

Service Flow terá reservado para si esta quantidade. Ao aplicar a política em outra tecnologia de rede, possivelmente haveria apenas uma reserva para todos os fluxos, independente dos endereços MAC constantes nos filtros dos fluxos. Esta diferença revela que, mesmo utilizando-se gerenciamento baseado em políticas, particularidades de cada tecnologia podem emergir e trazer a tona comportamentos diferentes do que o gerente poderia imaginar em um primeiro momento.

O QAME foi projetado inicialmente para trabalhar com fluxos agregados em políticas. A falta de expressividade na "linguagem de definição de políticas" do QAME é que acabou gerando a diferença citada no parágrafo anterior. A linguagem poderia permitir uma definição mais precisa sobre quando os recursos previstos em uma política são compartilhados e quando devem ser reservados separadamente. Pode-se dizer que o QAME não foi pensado para trabalhar com redes IEEE802.16, o que é um fato curioso, visto que gerenciamento baseado em políticas deveria ser algo genérico.

Na aplicação de políticas *multicast*, por outro lado, apenas um *Service Flow* é criado para todas as estações clientes associadas à política (o que garante a existência de uma reserva única de recursos). As estações cliente relacionadas que estiverem conectadas à estação base são associadas a este único *Service Flow*. Neste caso, entretanto, criou-se a necessidade de garantir que as *Classifier Rules* (filtros) associadas a todas as estações fossem as mesmas. Para garantir isto, no caso de políticas *multicast*, exige-se que os fluxos com filtros não apresentem filtro por endereço MAC e que haja um conjunto extra de fluxos que filtram apenas por MAC, indicando as estações relacionadas à política. A não existência de fluxos com filtro por MAC indica que todas as estações cliente estão relacionadas à política *multicast*. Esta "solução de contorno", embora funcione bem para especificar uma política *multicast* para redes IEEE802.16, torna a política não compatível para utilização em outras tecnologias de rede, o que vai contra os objetivos de se utilizar o gerenciamento baseado em políticas. De fato, a própria utilização de filtros por MAC já torna as políticas da parte IEEE802.16 da rede não compatíveis com o restante da rede, pois o MAC das estações cliente não identifica os pacotes a serem filtrados quando estes estão fora da rede IEEE802.16.

Outra modificação significativa no QAME foi a inclusão de novos atributos nas ações (Figura 6.3). Os atributos *Maximum Bandwidth*, *Maximum Jitter* e *Maximum Delay* não existiam na versão original do QAME. Apesar deste fato, mesmo sem o atributo *Maximum Bandwidth*, o QAME permitia que se indicasse que a vazão máxima (*Maximum Bandwidth*) deveria ser igual à vazão mínima (*Minimum Bandwidth*). A semântica desta indicação foi alterada para indicar que o fluxo deveria ser tratado como CBR (*Constant Bit Rate*). Todas estas modificações foram importantes para permitir que os requisitos de QoS fossem configurados conforme as exigências dos serviços de escalonamento das redes IEEE802.16 (descritos na seção "Visão Geral de IEEE802.16").

Os fluxos do tipo UGS exigem que se informe, no mínimo, os atributos de QoS *Maximum Sustained Traffic Rate* (*Maximum Bandwidth* no QAME), *Maximum Latency* (*Maximum Delay* no QAME) e *Tolerated Jitter* (*Maximum Jitter* no QAME).

Os fluxos do tipo rtPS exigem que se informe, no mínimo, os atributos de QoS *Minimum Reserved Traffic Rate* (*Minimum Bandwidth* no QAME), *Maximum Sustained Traffic Rate* (*Maximum Bandwidth* no QAME) e *Maximum Latency* (*Maximum Delay* no QAME). A mesma exigência é feita para os fluxos do tipo *extended* rtPS.

Os fluxos do tipo nrtPS exigem que se informe, no mínimo, os atributos de QoS *Minimum Reserved Traffic Rate* (*Minimum Bandwidth* no QAME), *Maximum Sustained Traffic Rate* (*Maximum Bandwidth* no QAME) e *Traffic Priority* (*Priority* no QAME).

Naturalmente não há exigências para fluxos do tipo melhor esforço (BE). Entretanto, a necessidade de conhecer tais atributos para configurar serviços de escalonamento que fornecessem QoS exigiu a modificação feita no QAME. Vale notar que os atributos adicionados fazem sentido no contexto de gerenciamento de QoS, mas não eram incluídos pelo QAME. Isto demonstra que, mesmo tendo-se um conjunto de atributos pensado como genérico durante a definição inicial da ferramenta, determinadas tecnologias podem necessitar de outros atributos não previstos para poder operar.

Tendo os atributos necessários, o PDP mapeia as ações em políticas de escalonamento da seguinte forma (em ordem de avaliação):

- Quando a ação indica tratamento do fluxo como CBR, o *Service Flow* é categorizado como UGS; os demais atributos obrigatórios são exigidos pelo PDP para aplicar a política;
- Quando o atributo *Minimum Bandwidth* é especificado em conjunto com o *Maximum Delay*, o *Service Flow* é categorizado com rtPS; os demais atributos obrigatórios são exigidos pelo PDP para aplicar a política;
- Quando o atributo *Minimum Bandwidth* é especificado em conjunto com o *Priority*, o *Service Flow* é categorizado como nrtPS; os demais atributos obrigatórios são exigidos pelo PDP para aplicar a política;
- O PDP proíbe a aplicação de políticas que definam os atributos DSCP e *Maximum Loss*, visto que estes atributos não são suportados em redes IEEE802.16; a especificação dos atributos *Minimum Bandwidth*, *Maximum Delay* ou *Priority* de forma isolada, impedindo a classificação do *Service Flow* em uma das categorais que fornecem QoS também é proibida;
- Nos casos remanescentes, o *Service Flow* é categorizado como BE.

Observa-se na lógica de categorização de *Service Flows* que o serviço de escalonamento *extended* rtPS não foi considerado. Isto é uma conseqüência da abordagem de gerenciamento baseado em políticas, visto que, na tentativa de manter as políticas o mais genéricas possível, eventualmente perde-se a possibilidade de explorar particularidades oferecidas por determinadas tecnologias.

Dois conceitos novos foram incluídos na definição de uma política: a prioridade de aplicação e o conjunto de equivalência. A prioridade de aplicação já foi abordada anteriormente na seção “Gerenciamento de QoS baseado em políticas”. Quanto ao conjunto de equivalência, trata-se de um conceito que se mostrou útil para gerenciar redes IEEE802.16. O objetivo do conjunto de equivalência é agrupar políticas relacionadas, como é detalhado abaixo.

Quando uma política é aplicada na rede IEEE802.16, mesmo nos casos em que ela se torna mais de um *Service Flow*, um único SFID (*Service Flow Identifier*) é utilizado para todos os *Service Flows*. Isto é possível porque a unicidade do SFID é garantida no contexto entre a estação base e uma estação cliente, de modo que ele pode ser repetido

no contexto de outras estações. No PDP implementado, o SFID é associado à política que originou o *Service Flow*.

A utilidade do conjunto de equivalência surge quando uma estação móvel faz *handover*. Nesta situação, *Service Flows* com os mesmos SFIDs dos configurados para a estação móvel devem estar pré-configurados na estação base destino para que não ocorra o cancelamento de *Service Flows* durante a migração. Se o SFID fosse puramente derivado da política, isto implicaria que exatamente a mesma política deveria estar configurada na estação base destino.

Como foi apresentado na seção “Gerenciamento de QoS baseado em políticas”, pode ser interessante fornecer uma QoS diferente a determinado fluxo dependendo da região da rede onde o usuário está realizando o acesso. De mesma forma, também pode ser desejável configurar uma prioridade de aplicação diferente para a política em outra região da rede. Para atingir estes objetivos, considerando o modelo de definição de políticas do QAME, seria necessário configurar o fluxo com uma política em uma dada região da rede, e configurá-lo com outra política em outra região. Entretanto, a utilização de duas políticas para especificar o mesmo fluxo é contrária à conclusão do parágrafo anterior, a menos que ambas as políticas pudessem gerar *Service Flows* com o mesmo SFID. O conceito de conjunto de equivalência foi criado justamente para permitir que duas políticas gerem *Service Flows* com o mesmo SFID.

No contexto do gerenciamento baseado em políticas, as políticas em um mesmo conjunto de equivalência são, essencialmente, variações da mesma política para utilização em regiões da rede distintas. Neste sentido, o conceito de conjunto de equivalência adquire um caráter de organização do conjunto de políticas. O QAME foi estendido para emitir um alerta quando duas políticas pertencentes ao mesmo conjunto de equivalência estão sendo configuradas na mesma interface de rede (mas não proíbe que isto seja feito).

No contexto de gerenciamento de redes IEEE802.16, quando uma política faz parte de um conjunto de equivalência, o SFID utilizado para configurar os *Service Flows* é derivado do conjunto de equivalência (ao invés da política), permitindo que todas as políticas sob o mesmo conjunto de equivalência estejam se referindo ao mesmo *Service Flow*. A Figura 6.11 ilustra a utilização de um conjunto de equivalência para que as políticas 1 e 2 tenham ambas o SFID x (um número qualquer). Desta forma, ambas as políticas estarão se referindo ao mesmo *Service Flow* em uma situação de *handover*. O PDP proíbe que se configure duas políticas pertencentes ao mesmo conjunto de equivalência em uma mesma interface de rede.

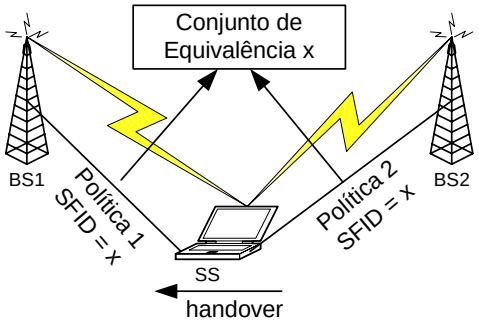


Figura 6.11: Utilização de um Conjunto de Equivalência no *handover*

O QAME utiliza uma base LDAP (ZEILENGA, 2006) para armazenar as políticas. O esquema da base LDAP utilizada pelo QAME é baseado nos modelos PCIM (*Policy Core Information Model*) (MOORE et al., 2001) e PCIME (*Policy Core Information Model Extensions*) (MOORE, 2003), ambos derivados de trabalhos do DMTF (DMTF, s.d.). Há uma RFC que estabelece o mapeamento destes modelos para um esquema LDAP (PANA et al., 2005).

O problema que havia no esquema LDAP utilizado pelo QAME é que, embora fosse um esquema padronizado, ele não era capaz de suportar os novos conceitos adicionados a definição de uma política. O esquema até possuía o conceito de prioridade, mas a semântica dada ao conceito era aquela de prioridade de definição de políticas, enquanto a prioridade que se tinha interesse no momento era a prioridade de aplicação de políticas.

Para permitir que as informações de prioridades de aplicação e conjunto de equivalência fossem associadas a uma política na base LDAP, optou-se por criar uma classe auxiliar (qamePolicyRuleParameters) que contivesse estas informações no esquema LDAP. O uso de uma classe auxiliar evitou que se tivesse que alterar o esquema padronizado já utilizado pelo QAME e permitiu que os objetos que representam políticas na base LDAP tivessem os novos atributos adicionados a sua estrutura. Também foi criada uma nova classe estrutural (qameEquivalenceSet) para servir de base ao conceito de conjunto de equivalência. As extensões específicas do QAME ao esquema LDAP padrão podem ser vistas no APÊNDICE G. A prioridade de aplicação das políticas foi chamada de prioridade de recursos (qameResourcePriority) no esquema LDAP porque os conflitos de aplicação de políticas são inerentes a insuficiência de recursos.

Uma modificação feita para simplificar a forma de expressar as políticas de QoS foi a criação de um auxílio visual para endereços MAC (*MAC Visual Aid*). A idéia é que, ao invés de tratar com os MACs das estações cliente diretamente, o gerente possa configurar uma *string* para representar um MAC e utilizar esta *string* em qualquer contexto onde o MAC seria necessário (por ex. nos filtros de um fluxo). O QAME também utiliza a *string* no lugar do MAC nos momentos em que este seria exibido. A Figura 6.12 exibe o formulário de cadastro de auxílios visuais para MAC do QAME. Se o auxílio indicado na Figura 6.12 estivesse cadastrado, a *string* “Estação 10” seria exibida (e poderia ser utilizada) no lugar do MAC no formulário da Figura 6.2. O significado da *string* é dado pelo gerente da rede, de modo esta poderia também representar o nome de um usuário, por exemplo.

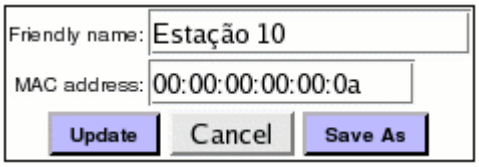


Figura 6.12: Formulário de cadastro de auxílio visual para MAC do QAME

A última extensão feita no QAME foi um mecanismo para atualização de políticas. O QAME foi concebido originalmente para considerar as políticas como imutáveis. Uma vez construídas e configuradas através de um PDP, o PDP nunca mais baixaria a mesma política e seus componentes. Esta decisão foi baseada na idéia de que, se fosse desejável alterar as políticas da rede, novas políticas seriam configuradas, ao invés de as

existentes serem alteradas. A única forma de alterar uma política no PDP, então, seria cancelar a aplicação da política e aplicá-la novamente após a modificação.

O mecanismo implementado permite que o administrador selecione as políticas (ou fluxos, ou ações, ou timers) que deseja reaplicar e indique que estas políticas devem ser atualizadas na base local de um PDP. Ao receber esta solicitação, o PDP baixa novamente as políticas e atualiza a configuração dos PEPs apropriadamente (evitando remover a configuração, se possível). No caso específico em que são políticas que estão sendo atualizadas, pode-se optar por apenas atualizar a composição da política (como apresentado pelo formulário da Figura 6.5), ou atualizar a composição e também todos os componentes da política (fluxos, ações e timers). O mecanismo de atualização de políticas é importante em função dos cenários de utilização imaginados, como será apresentado na seção seguinte.

6.3 Cenários de utilização do QAME

Esta seção irá apresentar alguns cenários de utilização do QAME, ilustrando como determinadas políticas poderiam ser criadas utilizando os recursos existentes.

Como um primeiro cenário, imaginemos que se deseja configurar uma reserva de 100kbps em *download* para a estação móvel 00:00:00:00:00:0a utilizar FTP. No espírito da utilização de um auxílio visual para o MAC, vamos utilizar a string “estação A” no lugar do MAC.

Neste caso, o administrador teria que criar um fluxo (FTP estação A) com os seguintes valores preenchidos (ver Figura 6.2): **Description** – “FTP estação A”; **Destination MAC Addresses** – “estação A”; **Destination Ports** – “20”; **Protocols** – “TCP”. Cabe notar que o fluxo FTP, neste caso, estaria sendo selecionado através de sua porta de dados (20), já que o QAME não dispõe de um mecanismo para especificar diretamente este protocolo.

Para especificar a QoS, o administrador teria, ainda, que criar uma ação (100kbps não RT) correspondente com os seguintes valores preenchidos (ver Figura 6.3): **Description** – “100kbps não RT”; **Minimum Bandwidth (Kbps)** – “100”; **Maximum Bandwidth (Kbps)** – “100”; **Treat as CBR flow** – desmarcado; **Priority** – “5”. Esta ação especifica requisitos de QoS para um fluxo não *real-time*. *Priority* é um atributo obrigatório em redes IEEE802.16 para fluxos não *real-time* com QoS. Ela se refere à prioridade de escalonamento de pacotes entre fluxos deste tipo.

Não foi definido o período em que a política deve estar ativa, de modo que podemos utilizar o temporizador apresentado na Figura 6.4 (“Sempre”), que torna a política sempre ativa. Como a manipulação de temporizadores é trivial, não se detalhará muito o seu uso nesta seção.

Finalmente, constrói-se uma política como a que foi definida indicando-se o fluxo “FTP estação A”, o temporizador “Sempre” e a ação “100kbps não RT” no formulário de criação de políticas (Figura 6.5). Neste momento, não estamos preocupados com o campo *Priority* e com o *Equivalence set* existentes no formulário, de modo que pode-se utilizar o valor “0” para o primeiro e “none” para o segundo (significando que a política não pertence a nenhum conjunto de equivalência). Chamemos esta política de “*Silver*” de agora em diante.

Para aplicar a política *Silver* na rede, basta configurá-la nas estações base relevantes utilizando a interface de aplicação de políticas (Figura 6.9), acessível para cada estação base a partir do mapa de rede. Deve-se selecionar a política *Silver*, a interface apropriada na estação base (o simulador implementado utiliza a interface de *loopback* como interface IEEE802.16) e a direção OUTPUT, que fará a política ser aplicada em *download* do ponto de vista da estação móvel. Cabe notar que não seria possível aplicá-la com a direção INPUT porque o filtro de "*Destination MAC Addresses*" está preenchido no fluxo, e o PDP não aceita este filtro nesta direção.

Suponhamos, agora, que esta mesma reserva deverá passar a valer para duas novas estações: "estação B" e "estação C". Neste caso, bastaria criar dois novos fluxos, equivalentes ao criado para a estação A, incluí-los na política *Silver*, e atualizar a política na configuração das estações base.

A utilidade da extensão que permite atualizar políticas fica evidente neste momento. Se este recurso não tivesse sido implementado, seria necessário cancelar a aplicação da política para poder reaplicá-la (causando uma interrupção no serviço da estação A). Outra alternativa seria criar uma nova política para as estações B e C, mas isto tornaria o gerenciamento do grupo de políticas mais difícil. Fica óbvio que a própria política está sendo utilizada para agrupar estações móveis, visto que o QAME não possui outra forma de agrupar fluxos.

Para fazer o PDP baixar a nova versão da política e atualizá-la nos correspondentes PEPs, utiliza-se o recurso de atualização de políticas, adicionado à interface de gerenciamento de políticas do QAME (Figura 6.13). O gerente da rede seleciona as políticas que deseja atualizar, seleciona o PDP que deverá ser atualizado (na parte de baixo da interface) e clica em "*Propagate modifications*" para atualizar a composição da política. Alternativamente o gerente pode clicar em "*Propagate all modifications*" para atualizar a composição da política e também os atributos de todos os componentes da política (fluxos, ações, temporizadores). Nas interfaces de gerenciamento de fluxos, ações e temporizadores também há a possibilidade de acionar o botão "*Propagate modifications*".

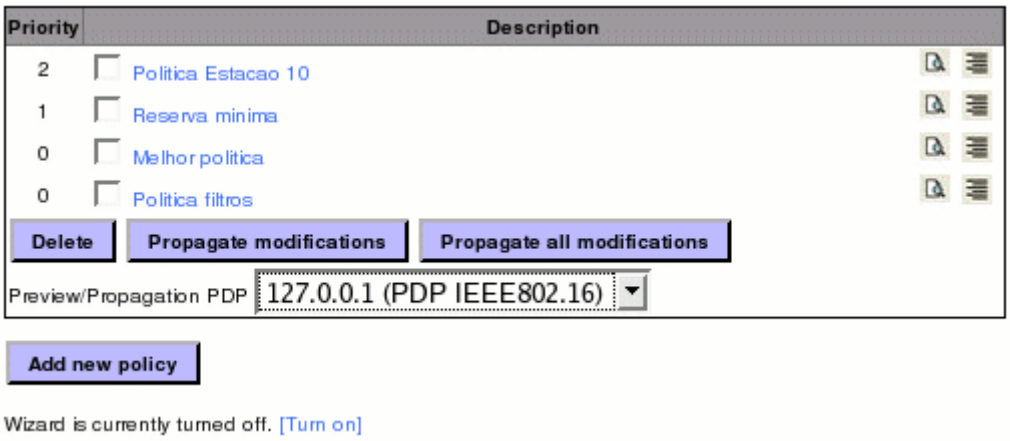


Figura 6.13: Interface de gerenciamento de políticas do QAME

O cenário de configuração de uma política *multicast* seria um pouco diferente. Imaginemos que se deseja configurar uma reserva de 250kbps em *download* para as três estações móveis acima assistirem a uma videoconferência que será feita através do endereço IP *multicast* 224.0.1.2. Neste caso, o fluxo terá restrições de tempo real, visto

que os participantes estarão interagindo ao vivo com o emissor do fluxo através de outros canais (não abordados aqui). Assumiremos que o fluxo poderá ter uma flutuação de vazão que chegue a 300kbps e que se deseja restringir o atraso no trecho IEEE802.16 da rede a 10ms.

O primeiro passo a ser feito é criar um fluxo para identificar os pacotes a serem priorizados. Utilizaremos os seguintes atributos (Figura 6.2): **Description** – "Vídeoconferência multicast"; **Destination IP Addresses** – "224.0.1.2"; **Protocols** – "UDP". Em seguida, é necessário criar um fluxo para identificar cada uma das estações participantes. No caso da estação A, o fluxo poderia ser assim: **Description** – "Destino estação A"; **Destination MAC Addresses** – "estação A". Note que também se poderia criar um único fluxo para as três estações, visto que o campo *Destination MAC Addresses* admite mais de um endereço.

A ação no cenário da videoconferência seria criada com os seguintes atributos: **Description** – "250kbps real-time"; **Minimum Bandwidth (Kbps)** – "250"; **Maximum Bandwidth (Kbps)** – "300"; **Treat as CBR flow** – desmarcado; **Maximum Delay (ms)** – "10". Não é possível configurar um limite de perda para este fluxo porque isto não é suportado em redes IEEE802.16.

Juntando-se os fluxos "Videoconferência multicast", "Destino estação A", "Destino estação B" e "Destino estação C", bem como a ação "250kbps real-time" em uma política, teríamos a configuração que desejamos preparada, restando apenas aplicá-la nas estações base relevantes na rede.

Como um último cenário relevante, consideremos o caso de uma operadora IEEE802.16 que forneça a seus usuários, no mínimo, 500kbps de banda em *donwload*, contanto que os mesmos estejam acessando a rede a partir da estação base próxima à sua residência. A reserva é reduzida para 200kbps e não é 100% garantida se o usuário estiver acessando de outro local.

A configuração deste cenário poderia ser feita com duas políticas por estação base: uma política local e uma política de *roaming*. A política local é configurada na própria estação base, contendo fluxos que identificam as estações móveis locais àquela estação base e contendo como ação a reserva de 500kbps. A política de *roaming* é configurada em todas as outras estações base (exceto a própria estação base local). Ela deve possuir os mesmos fluxos da política local e uma ação que indique a reserva de 200kbps.

Para garantir que os usuários locais terão preferência sobre usuários que estejam fazendo *roaming* em sua estação base, as políticas locais devem ter uma prioridade maior que as políticas de *roaming*. Com uma configuração assim, o PDP jamais aceitaria que um fluxo local fosse rejeitado havendo fluxos em *roaming* suficientes para liberar os recursos necessários ao fluxo local (os fluxos em *roaming* seriam cancelados). Note que uma dada estação base estará configurada com sua política local e com as políticas de *roaming* de todas as outras estações base.

Para que os *Service Flows* dos usuários possuam o mesmo SFID em qualquer estação base na rede (fazendo funcionar o processo de *handover*), as políticas local e de *roaming* associadas a uma dada estação base X deverão fazer parte do mesmo conjunto de equivalência (políticas X). Desta forma, também fica ilustrado o uso do conceito de conjunto de equivalência.

Um detalhe importante a considerar nas redes IEEE802.16 é que as conexões básica, primária e secundária, criadas automaticamente quando a estação cliente entra na rede, não carregam dados de usuário (apenas dados de gerenciamento e controle). Deve haver pelo menos duas conexões de transporte (*Service Flows*) para este fim (uma em *download* e outra em *upload*). Desta forma, devem ser criadas e publicadas em todas as estações base políticas que englobem todas as estações cliente e lhes ofereçam serviço do tipo melhor esforço para todos os seus pacotes (se isto estiver de acordo com a política da rede).

6.4 Detalhes de implementação do PDP

Seguindo a linha do ambiente QAME, o PDP também foi implementado em PHP, mas utilizou-se a versão 5 desta linguagem (ao invés da versão 4). A vantagem da versão 5 neste contexto foi a possibilidade de utilizar nomes de objetos SNMP carregados a partir de um arquivo de MIB. Uma desvantagem da utilização de PHP é a necessidade de configurar os objetos SNMP um a um, visto que esta linguagem não permite o envio de vários OIDs em uma única mensagem SNMP.

Sempre que um novo PEP é adicionado sob o controle do PDP, ele realiza um processo de inicialização sobre este PEP e começa a monitorar os eventos relativos ao mesmo através de notificações (conforme descrito na seção “Operação Geral da MIB”). O PDP controla as estações clientes conectadas e registradas, bem como o consumo de recursos na rede, o que permite que o PDP saiba como reagir em cada situação.

Para receber notificações, o PDP utiliza o utilitário *snmptrapd* do pacote NET-SNMP (NET-SNMP, s.d.). O PDP possui um processo principal, que está sempre rodando e verificando se não deve alterar a configuração dos PEPs, e um processo extra, que é criado pelo *snmptrapd* sempre que este recebe uma notificação (de fato, o *snmptrapd* pode criar mais de um processo extra simultaneamente). O processo extra organiza as notificações recebidas do *snmptrapd* em uma estrutura de dados e as repassa ao processo principal utilizando mecanismos de comunicação entre processos. O processo principal é que toma atitudes conforme as notificações recebidas. O próprio processo principal inicia (e termina) o *snmptrapd* que irá receber as notificações.

O PDP foi implementado conforme a arquitetura padrão definida para PDPs do QAME. Esta arquitetura possui diversas camadas, o que torna os PDPs bastante modulares. Uma camada em especial (a camada de adaptação) é a responsável por realizar a comunicação com o PEP utilizando um protocolo compreensível por este. No caso do PDP implementado, esta camada comunica-se com o PDP utilizando SNMP. Entretanto, ela poderia facilmente ser substituída para que a comunicação ocorresse através de outro protocolo. Como já foi mencionado anteriormente, a comunicação com a estação gerente se dá através de Web Services oferecidos pelo PDP.

O mecanismo de prioridades de aplicação para resolução de conflitos foi implementado no PDP. Se um *Service Flow* de maior prioridade é rejeitado, o PDP verifica se há algum *Service Flow* de menor prioridade admitido que poderia ser cancelado para liberar recursos ao *Service Flow* de maior prioridade.

Não foram implementados no PDP mecanismos de prioridade de definição de políticas ou resolução de situações de degradação, até porque estes mecanismos também não foram implementados no QAME. No caso do mecanismo para tratar situações de

degradação, percebeu-se, ainda que tardiamente, que ficou faltando, na MIB definida, uma tabela que permitisse ao PDP identificar uma situação pré-existente de degradação. Esta tabela seria importante no momento em que o PDP fosse inicializar seu estado com relação a uma dada estação base.

6.5 Conclusões obtidas após as implementações

A principal conclusão obtida com a implementação foi que, apesar de o gerenciamento baseado em políticas facilitar o gerenciamento de uma rede IEEE802.16, o gerenciamento de redes IEEE802.16 apresenta uma série de particularidades que devem ser tratadas e que podem forçar o sistema de gerenciamento baseado em políticas a perder um pouco de sua generalidade. Algumas conclusões afetam gerenciamento baseado em políticas de um modo geral.

Inicialmente, observou-se que os atributos das ações do QAME eram insuficientes para definir as classes de QoS existentes em IEEE802.16. É verdade que os novos atributos adicionados são perfeitamente genéricos no contexto de gerenciamento de QoS. Entretanto, o conjunto inicialmente definido também o era, mas não era suficientemente completo para que pudesse ser utilizado em redes IEEE802.16. Percebe-se que, mesmo em gerenciamento baseado em políticas, situações específicas podem exigir adaptações.

Também convém notar que, apesar de incompleto, o conjunto de atributos de ações inicialmente suportado pelo QAME não podia ser suportado integralmente em uma rede IEEE802.16. Os atributos de limite de perda e marcação de pacotes não poderiam ser configurados. Estas diferenças entre o que pode ser suportado por cada tecnologia levanta dúvidas quanto a real possibilidade de definirem-se políticas de QoS genéricas para toda a rede.

Outra observação feita, mas que até certo ponto já era esperada, é que a utilização de gerenciamento baseado em políticas pode impedir que se explore toda a capacidade de uma dada tecnologia. O exemplo mais chamativo no caso de IEEE802.16 é a impossibilidade de utilizar *Service Flows* com política de escalonamento *extended* rtPS através do QAME. Em função da generalidade do QAME, foi, inclusive, difícil relacionar os atributos das ações do QAME às outras políticas de escalonamento do IEEE802.16.

A velha relação do cobertor curto é que impera no gerenciamento baseado em políticas: quanto maior a generalidade, mais abrangente é o sistema e menor sua capacidade de expressão. Para explorar toda a capacidade das redes IEEE802.16, poderia-se definir um conjunto de atributos que conseguisse expressar os recursos adicionais. O problema é que isto causaria a perda da possibilidade de utilizar-se as mesmas definições em outras situações.

Olhando por outro lado, entretanto, vemos que encontrar um conjunto de atributos que seja necessário e suficiente para abordar todos os casos parece ser uma utopia, como foi explorado nos primeiros parágrafos. Talvez o melhor seja permitir a utilização de extensões específicas de cada tecnologia, sempre tentando ser o mais genérico possível, e valer-se principalmente da maior facilidade de expressão e gerenciamento integrado existentes em sistemas de gerenciamento baseado em políticas. O conceito de conjunto de equivalência poderia auxiliar na organização de políticas relacionadas.

O modelo adotado pelo QAME, apesar de bastante genérico, não permite expressar as políticas de uma forma única e consistente para toda a rede. As políticas definidas para a parte IEEE802.16 da rede são incompatíveis com as políticas definidas para a parte fixa da rede. Em políticas para IEEE802.16, existe a possibilidade de utilizarem-se endereços MAC, o que não é algo tipicamente aceito na parte fixa (mesmo que fosse aceito, não haveria como o MAC referir-se a uma estação móvel visto que o MAC da mesma somente existe no enlace aéreo). Outro ponto a ser observado é a possível multiplicação de reservas de uma política na parte móvel da rede em função da criação de diversos *Service Flows* (um para cada estação cliente), algo que tipicamente não ocorreria na parte fixa da rede. Pode-se dizer que o QAME não estava preparado para tratar IEEE802.16 em função da forma como ele operava. Se a definição de políticas fosse feita de outra forma, talvez estas diferenças fossem menos acentuadas.

Com relação ao gerenciamento de IEEE802.16, observa-se que a exigência de um tratamento adequado às situações de *handover* é uma das características que mais afeta o sistema de gerenciamento. A utilização do mecanismo baseado no *ranging*, como está sendo avaliado neste trabalho, simplifica muito este tratamento ao permitir que o PDP preocupe-se com cada estação base isoladamente. Neste caso, basicamente foi necessária a criação do conceito de conjunto de equivalência para tratar as situações de *handover*. Por outro lado, a utilização do mecanismo baseado no *registering*, que visa provisionar os *Service Flows* em todas as estações base vizinhas àquela a que uma dada estação móvel está registrada, exigiria uma verdadeira reviravolta na arquitetura de gerenciamento.

Para se utilizar o mecanismo de provisionamento baseado no *registering*, seria necessário que os PDPs recebessem informações sobre a topologia lógica das estações base (isto e, que estações base são vizinhas de que outras). Além da topologia lógica, eles teriam que saber que PDPs são responsáveis por controlar as estações base vizinhas às suas. Qualquer notificação de registro ou desregistro por parte de uma estação móvel teria de ser comunicada pelo PDP aos PDPs que controlam as estações base vizinhas daquela onde ocorreu o registro ou desregistro (assume-se que a comunicação via estação de gerenciamento é inviável por falta de escalabilidade). Isto é algo que sai fora do usual para a arquitetura de gerenciamento baseado em políticas do IETF.

Por fim, durante os trabalhos de desenvolvimento, observou-se que alguns elementos poderiam ser melhorados na definição da MIB para gerenciamento:

- Faltou uma tabela que permita ao PDP identificar a situação de degradação da rede no momento em que ele vai inicializar seu estado com relação a uma dada estação base.
- Faltou uma notificação que indique que um *Service Flow* deixou de existir. Isto é particularmente importante quanto o PDP não é o único a criar e remover *Service Flows* na rede. Se os *Service Flows* somente puderem ser criados e removidos via sistema de gerenciamento, o PDP pode contornar esta situação observando o registro e desregistro das estações cliente.
- Faltou a possibilidade de indicar quais atributos são relevantes na especificação de uma *Service Class*. Dado que o PDP poderia criar classes que não possuam todos os atributos, deveria ser possível indicar quais atributos são válidos. A impossibilidade desta indicação complica a implementação de um mecanismo de

alteração de uma *Service Class* quando se deseja indicar que um atributo não estará mais definido.

- A notificação que indica registro de uma estação poderia conter mais informações. Em particular, houve interesse em que fossem incluídas as taxas de *upload* e *download* da estação cliente que se registrou.
- A MIB classifica os fluxos como *authorized*, *admitted* ou *active*, mas a nomenclatura correta, definida no padrão IEEE802.16, é que os fluxos podem estar em estado *provisioned*, *admitted* ou *active*. Seria interessante que se utilizasse a mesma nomenclatura, até porque a denominação *authorized* pode transmitir a idéia de que existe uma autorização para uso dos recursos, o que não condiz com o que a documentação do campo *wmanIfBsSfState*.

7 RESULTADOS EXPERIMENTAIS

Este capítulo apresenta os resultados obtidos através da avaliação da implementação apresentada no capítulo 6. As seções que seguem abordarão a descrição do cenário experimental e a descrição das experiências realizadas.

7.1 Cenário experimental

Em linhas gerais, o cenário experimental utilizado consiste em três estações base simuladas (PEPs), sendo gerenciadas através de um mesmo PDP. Naturalmente uma estação de gerenciamento QAME também foi necessária para poder controlar todo o sistema.

Diversos serviços precisam estar no ar para possibilitar o cenário proposto acima. Deve haver um servidor LDAP configurado com os esquemas apropriados para armazenar a base de políticas. Utilizou-se o Open LDAP neste caso. Tanto o QAME, quanto o PDP possuem bases de dados locais, as quais, por questões de agilidade, devem idealmente ficar nas próprias máquinas do QAME e do PDP. Em ambos os casos, utilizou-se um servidor MySQL, conforme planejado originalmente para as aplicações. Nas máquinas onde rodam o QAME e o PDP também há a necessidade de servidores *web*, visto que o primeiro é, de fato, uma aplicação *web*, e o segundo possui serviços disponibilizados através de Web Services. Em ambos os casos, um servidor Apache foi utilizado como servidor *web*.

Havia três máquinas disponíveis para a realização dos experimentos. Um *notebook* situado no Rio de Janeiro, e dois *desktops* situados em uma mesma rede local em Porto Alegre, separados por um *switch*. A comunicação entre as máquinas de Porto Alegre e o *notebook* no Rio de Janeiro somente poderia ocorrer via Internet. O *notebook* possuía 512Mb de memória e processador Turion 64 rodando a 1,8Ghz com um sistema de 32 bits. O *desktop* 1 possuía 256Mb de memória e processador Athlon XP 1700+ rodando a 1,1Ghz. O *desktop* 2 possuía 2Gb de memória e processador Celeron 2,66Ghz rodando em sua freqüência habitual.

O QAME foi instanciado em uma máquina virtual executando sobre o *notebook* do Rio de Janeiro. O fato de estar distante do sistema gerenciado não preocupava, pois isto seria esperado em um ambiente de produção. Esta máquina virtual rodava um sistema CentOS 4.3 e eram simulados 160Mb de memória. A comunicação com o mundo externo ocorria através de NAT pela máquina hospedeira, a qual também passava por um NAT no roteador de saída de sua rede. O banco de dados local e o servidor *web* do QAME, bem como o servidor LDAP, foram instanciados todos nesta mesma máquina

virtual. A proximidade do QAME com o servidor LDAP permite agilidade na manipulação das políticas.

O PDP foi instanciado em uma máquina virtual executando sobre o *desktop* 2. Esta máquina virtual rodava um sistema CentOS 4.5 e eram simulados 256Mb de memória. A comunicação com o mundo externo ocorria através de *bridge* para sua rede local pela máquina hospedeira, a qual passava por um NAT se desejasse acessar a Internet. A comunicação com o *desktop* 1 ocorria diretamente através do *switch*, sem a necessidade de NAT. O banco de dados local e o servidor *web* do PDP foram instanciados nesta mesma máquina.

Por fim, as três estações base simuladas foram instanciadas em IPs diferentes no próprio *desktop* 1, o qual rodava um sistema CentOS 4.3. Os IPs foram propositalmente alocados em sub-redes classe C distintas sobre a mesma rede local para causar a ilusão de que se tratavam de três máquinas diferentes. Os IPs eram 10.0.1.8, 10.0.4.1 e 10.0.5.1 e a máquina virtual que rodava o PDP possuía IPs correspondentes nestas três sub-redes (10.0.1.9, 10.0.4.2 e 10.0.5.2).

A Figura 7.1 ilustra o cenário experimental descrito. A Figura 7.2 apresenta o mapa de rede montado no QAME para representar a rede gerenciada.

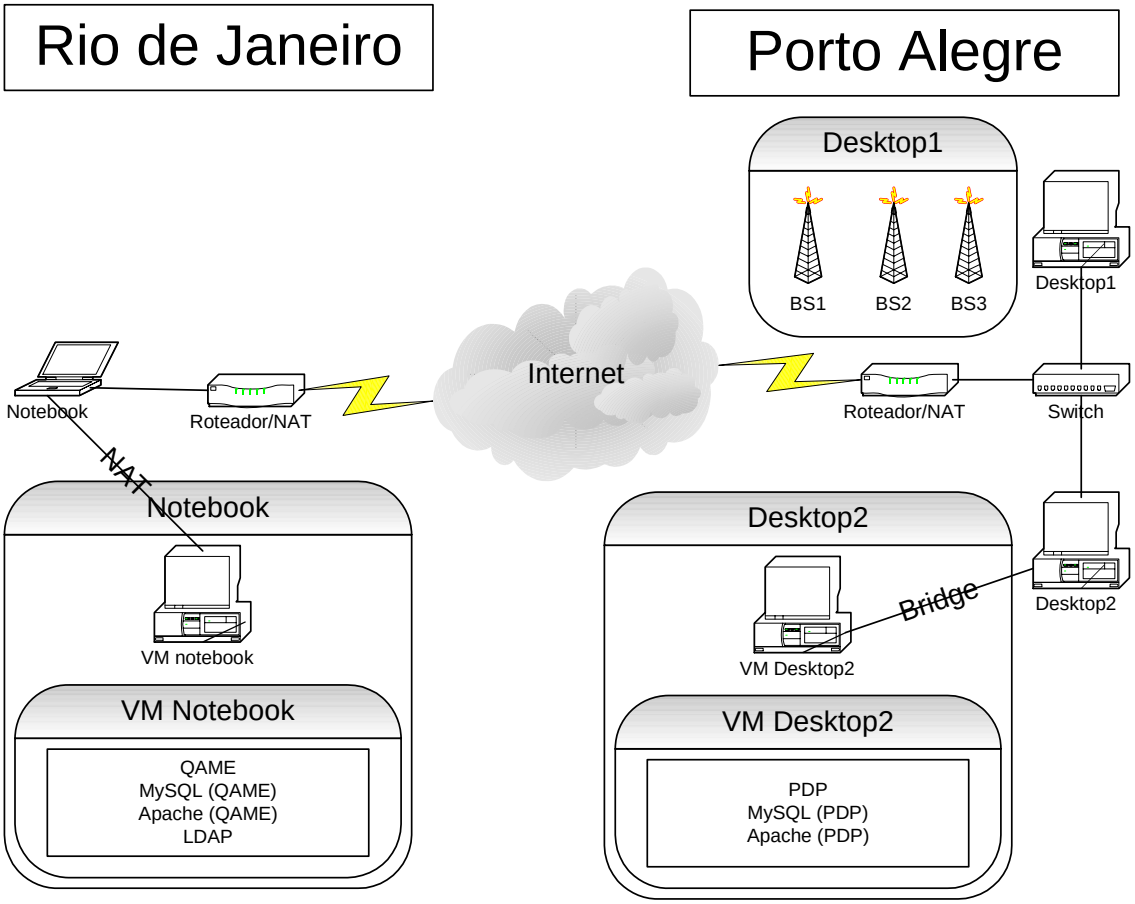


Figura 7.1: Cenário experimental utilizado

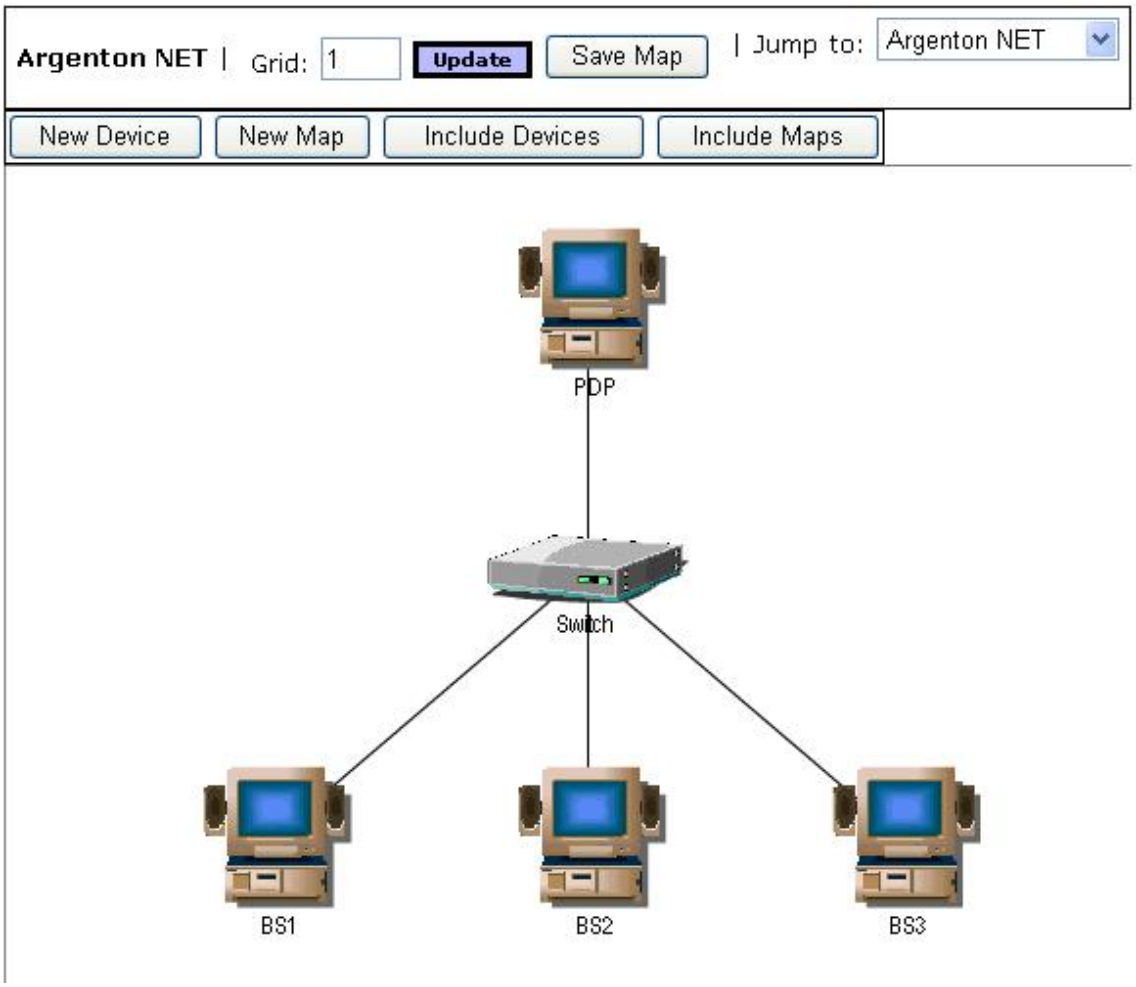


Figura 7.2: Mapa da rede gerenciada no QAME

O último ponto a ser esclarecido é como as máquinas se comunicavam, dado que estavam todas sob NAT. O serviço de SSH existente no *desktop* 1 foi disponibilizado externamente no roteador de sua rede, permitindo a conexão entrante a partir do *notebook*. Através do protocolo SSH, foram criados túneis que permitiam tanto o acesso HTTP da máquina virtual do *notebook* à máquina virtual do *desktop* 2 (QAME <-> PDP), quando o acesso LDAP da máquina virtual do *desktop* 2 à máquina virtual do *notebook* (PDP <-> LDAP). A comunicação SNMP entre o PDP e as estações base ocorreu diretamente na rede local, o que é um cenário esperado na prática.

Os serviços de HTTP e LDAP não foram disponibilizados externamente porque as máquinas utilizadas não serviam exclusivamente como servidores, e seria um risco de segurança torná-las acessíveis abertamente na *web*. Entretanto, durante as avaliações, evitou-se que fossem utilizadas para outros fins.

7.2 Experimentos realizados

Foram realizados diversos experimentos visando responder às seguintes perguntas:

1. Quanto tempo o PDP leva para configurar determinadas políticas após uma estação móvel conectar-se em uma estação base (tempo de resposta)?
2. Quantas mensagens são necessárias para configurar determinadas políticas?

3. Qual a banda consumida na comunicação PDP <-> PEP?
4. Quanto tempo o PDP leva para reconfigurar os Service Flows em função da rejeição de um fluxo associado a uma política de alta prioridade (tempo de reação à inversão de prioridade)?
5. Qual seria o tempo de resposta (referente à questão 1) em um cenário em que há várias estações se movimentando aleatoriamente pela rede?

As subseções que seguem tratarão dos grupos de experimentos.

7.2.1 Avaliação de tempo de resposta, número de mensagens e banda consumida

Para responder às três primeiras questões, foram preparados *scripts* de simulação para apenas duas estações base. Nestes *scripts*, configurou-se um cenário no qual uma estação móvel fica efetuando *handover* entre as duas estações base continuamente. Após registrar-se em uma estação base, a estação móvel aguarda 25s e efetua novo *handover* para a outra estação base. Ao todo são realizadas 20 conexões em função de *handover* e 1 conexão extra representando a entrada inicial da estação na rede, o que permite obter uma boa idéia do tempo de resposta do PDP.

Estes *scripts* de simulação foram rodados com diferentes configurações de políticas, de modo a capturar a variação do número de mensagens, da banda consumida e do tempo de resposta. As seguintes políticas foram avaliadas em seqüência:

1. Reserva para Vídeo on Demand com banda mínima de 200kbps, banda máxima de 500kbps e atraso limitado a 500ms. Filtro realizado pelo MAC da estação destino e pelo DSCP 2;
2. Mesma reserva do item 1, mas com a inclusão do filtro pelo protocolo UDP;
3. Mesma reserva do item 2, mas com a inclusão do filtro pelo IP de destino 192.168.0.1 e pela porta de destino 2000;
4. Mesma reserva do item 3, mas com a inclusão do filtro pelo IP de origem 192.168.0.2 e pela porta de origem 3000;
5. Mesma reserva do item 3, mas com a inclusão do filtro pela porta de destino 2001;
6. Mesma reserva do item 5, mas com a inclusão do filtro pelo DSCP 3;
7. Mesma reserva do item 6, mas com a inclusão do filtro pelos DSCPs 4 e 5;
8. Reserva do item 1 e uma réplica que filtra pelo DSCP 6;
9. Reserva do item 1 e três réplicas que filtram pelos DSCPs 6, 7 e 8, respectivamente;
10. Reserva do item 1 e sete réplicas que filtram pelos DSCPs de 6 a 12 respectivamente;
11. Mesma reserva do item 10, mas com a utilização de duas ações distintas, uma para cada quatro políticas;
12. Mesma reserva do item 10, mas com a utilização de quatro ações distintas, uma para cada duas políticas;

13. Mesma reserva do item 10, mas com a utilização de uma ação distinta para cada política;
14. Reserva para VoIP com banda mínima e máxima de 50kbps (CBR), atraso máximo de 25ms e *jitter* máximo de 5ms. Filtro realizado pelo MAC da estação e pelo DSCP 1. Duas políticas são configuradas, pois a reserva é feita em *Upload* e *Download*;
15. Reservas 1 e 14 simultaneamente;
16. Mesma reserva do item 15, mas com a inclusão de duas políticas com ação melhor esforço para representar a política de *upload* e *download* padrão para os pacotes de dados.

A escolha das configurações de políticas de 1 a 13 foi feita de modo a provocar variações em quatro eixos de avaliação: número de atributos em cada *Classifier Rule*, número de *Classifier Rules*, número de políticas e número de ações. As configurações de 14 a 16 foram incluídas para avaliar outras configurações plausíveis na rede. Os valores dos eixos de avaliação para cada configuração de política podem ser vistos na Tabela 7.1.

Tabela 7.1: Valores dos eixos de avaliação para cada configuração de política

Configuração	**Ações**	**Atributos Ação**	**Políticas**	***Classifier Rules***	**Atributos *Classifier Rule***
1	1	3	1	1	1
2	1	3	1	1	2
3	1	3	1	1	4
4	1	3	1	1	6
5	1	3	1	2	2x 4
6	1	3	1	4	4x 4
7	1	3	1	8	8x 4
8	1	3	2	2x 1	2x 1
9	1	3	4	4x 1	4x 1
10	1	3	8	8x 1	8x 1
11	2	2x 3	8	8x 1	8x 1
12	4	4x 3	8	8x 1	8x 1
13	8	8x 3	8	8x 1	8x 1
14	1	4	2	2x 1	2x 1
15	2	3 e 4	3	3x 1	3x 1
16	3	0, 3 e 4	5	5x 1	2x 0 e 3x 1

A avaliação do número de mensagens, da banda consumida e do tempo de resposta para cada configuração de política foi feita com o auxílio de um *sniffer*. O *sniffer* foi posicionado na máquina em que estavam rodando as estações base, de modo a permitir a observação do tempo entre o envio de uma notificação de conexão pela estação base e o término da configuração da política pelo PDP.

Para a primeira configuração de política, uma análise mais detalhada será apresentada com o intuito de ilustrar o que ocorre no nível de mensagens SNMP entre o PDP e a estação base (PEP). A seqüência de eventos para a configuração de política 1 pode ser vista no Quadro 7.1. Para cada evento, o número seqüência da mensagem

SNMP observada é dado entre parênteses, seguido pelo tempo (em segundos a partir da primeira mensagem) em que a mesma foi observada. Eventos que duram mais de uma mensagem possuem a indicação da primeira e da última mensagens, seguida pelo número total de mensagens que compõem o evento. Deve-se atentar que, em alguns casos, pode haver mensagens de outros eventos intercaladas. Os eventos de configuração possuem número de mensagens par, pois envolvem pares requisição/resposta. Os eventos de notificação envolvem apenas a mensagem de notificação. Linhas iniciadas por "#" são comentários, não representando evento algum.

```
#Inicio da execução dos simuladores
coldStart BS1: (1) 0.000000
coldStart BS2: (2) 1.887032

#Configuração da política pelo sistema de gerenciamento
Busca de interfaces na BS1: (3) 16.391108 -> (26)
    26.115071: 24 mensagens
Inicialização BS1: (27) 38.546978 -> (46) 38.580885: 44
    mensagens
Busca de interfaces na BS2: (47) 43.237205 -> (70)
    52.173373: 24 mensagens
Inicialização BS2: (71) 54.924382 -> (90) 54.955045: 44
    mensagens

#Entrada da estação na rede pela BS1
Conexão SS<->BS1: (91) 160.126792
Registro SS<->BS1: (92) 160.127428
Configuração dos fluxos na BS1: (93) 160.263105 -> (151)
    160.339290: 58 mensagens
Admissão de serviço na BS1: (150) 160.338965
Busca das taxas de serviço na BS1: (152) 160.455318 ->
    (155) 160.457736: 4 mensagens

#Handover para a BS2
Conexão SS<->BS2: (156) 172.020949
Configuração dos fluxos na BS2: (157) 172.222073 -> (214)
    172.293470: 58 mensagens
Registro SS<->BS2: (215) 182.029522
Admissão de serviço na BS2: (216) 182.030188
Indicação de handover na BS1: (217) 182.145110
Busca das taxas de serviço na BS2: (218) 182.182944 ->
    (221) 182.185369: 4 mensagens
Desregistro SS<->BS1: (222) 186.148761
Desconexão SS<->BS1: (223) 186.149336
Desconfiguração dos fluxos na BS1: (224) 186.543559 ->
    (239) 186.582762: 16 mensagens

#Handover para a BS1
Conexão SS<->BS1: (240) 196.157201
Configuração dos fluxos na BS1: (241) 196.303595 -> (298)
```

```
    196.373540: 58 mensagens
Indicação de handover na BS2: (299) 207.050571
Registro SS<->BS1: (300) 207.166542
Admissão de serviço na BS1: (301) 207.167213
Busca das taxas de serviço na BS1: (302) 207.331784 ->
    (305) 207.334010: 4 mensagens
Desregistro SS<->BS2: (306) 212.054808
Desconexão SS<->BS2: (307) 212.055378
Desconfiguração dos fluxos na BS2: (308) 212.347859 ->
    (323) 212.387751: 16 mensagens

# Uma série de handovers ocorrem até o final da simulação
# (...)
```

Quadro 7.1: Eventos SNMP para a configuração de política 1

Diversos eventos podem ser observados no Quadro 7.1. A Tabela 7.2 descreve mais detalhadamente cada evento.

Tabela 7.2: Descrição dos eventos SNMP para a configuração de política 1

Evento	**Descrição**
coldStart	*Trap* que indica a inicialização do processo *snmpd* que roda o simulador.
Busca de interfaces	Mensagens *snmpget* e respostas para descoberta de interfaces. O PDP realiza a descoberta e retorna o resultado para que a estação de gerenciamento QAME possa aplicar políticas. O evento engloba mais de uma descoberta de interface, sendo que cada descoberta é composta por oito mensagens no experimento. O PDP não faz cache do mapeamento entre o nome das interfaces (utilizado nas políticas) e seus respectivos índices (utilizados em mensagens SNMP).
Inicialização	O PDP realiza o procedimento de inicialização com a estação base, conforme descrito na seção “Operação Geral da MIB”.
Conexão	*Trap* que indica a conexão da estação móvel com a estação base.
Registro	*Trap* que indica o registro da estação móvel com a estação base.
Configuração dos fluxos	Mensagens *snmpset* e respostas para configurar os *Service Flows* conforme a política da rede. O tempo entre o evento de conexão e o término da configuração dos fluxos é o tempo de resposta que se deseja medir. No início da configuração, um procedimento de descoberta de interfaces é realizado para mapear os nomes de interfaces em seus respectivos índices.
Admissão de serviço	*Trap* que indica que um *Service Flow* foi admitido na rede
Busca das taxas de serviço	Mensagens *snmpget* e respostas para obter as taxas de *upload* e *download* para a estação móvel, de modo a auferir consumo de recursos.

Indicação de *handover*	*Trap* que indica que a estação móvel efetuou *handover* para outra estação base.
Desregistro	*Trap* que indica o desregistro da estação móvel com a estação base.
Desconexão	*Trap* que indica a desconexão da estação móvel com a estação base.
Desconfiguração dos fluxos	Mensagens *snmpset* e respostas para remover a configuração dos *Service Flows* de uma estação móvel que não está mais vinculada à estação base. Envolve poucas mensagens, visto que exige basicamente um *snmpset* para destruir cada registro em tabela SNMP. No início da desconfiguração, um procedimento de descoberta de interfaces é realizado para mapear os nomes de interfaces em seus respectivos índices.

Apresentado em detalhes o experimento realizado, chegou o momento de apresentar os resultados consolidados. A Figura 7.3 apresenta a variação do número de mensagens SNMP conforme o número de atributos nas *classifier rules* (configurações de política 1, 2, 3 e 4).

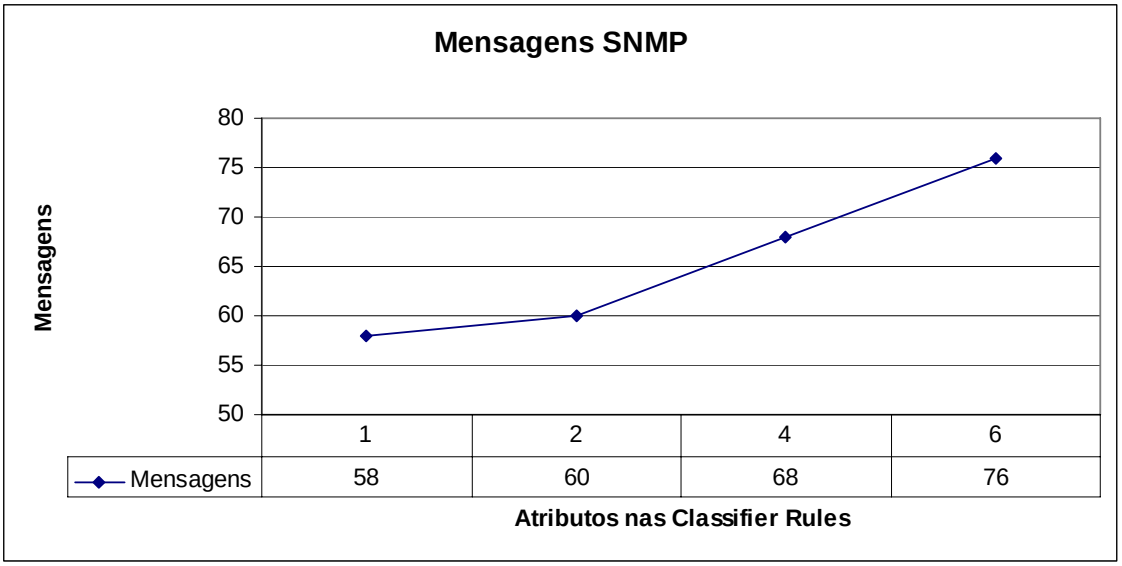


Figura 7.3: Número de mensagens SNMP em relação ao número de atributos nas *classifier rules*

O aumento no número de atributos de 1 para 2 provocou uma variação de duas mensagens SNMP. Isto já era esperado, visto que haveria pelo menos uma mensagem SNMP de requisição e uma de resposta para configurar o atributo adicional. Entretanto, ao passar de 2 para quatro atributos (e de 4 para 6), houve um amento de oito mensagens. Isto revela que nem sempre a relação entre número de atributos e mensagens SNMP é direta. No exemplo em questão, os atributos IP e porta exigem a configuração de dois objetos SNMP cada. O IP exige a configuração da máscara de rede e a porta é configurada sempre como um range, o que envolve dois objetos SNMP.

A Figura 7.4 apresenta a variação no tempo de resposta em função do número de atributos nas *classifier rules*. Percebe-se na figura que o tempo de resposta não varia muito com o número de atributos nas *classifier rules*, uma conseqüência da pequena

representatividade destes atributos frente ao total de mensagens SNMP necessárias à configuração das políticas.

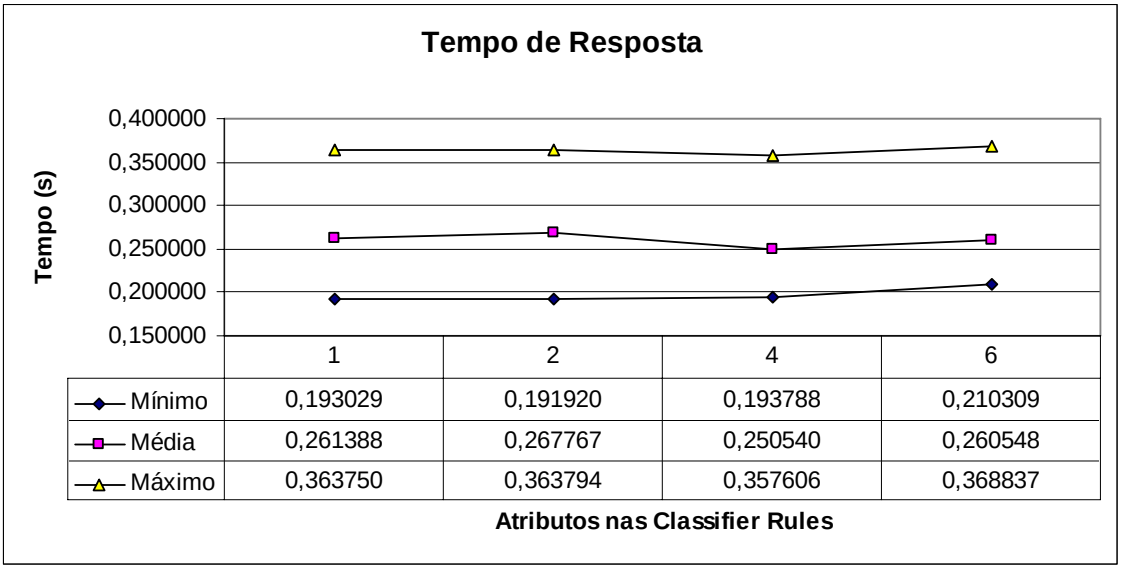


Figura 7.4: Tempo de resposta em relação ao número de atributos nas *classifier rules*

A Figura 7.5 apresenta a variação do número de mensagens SNMP em relação ao número de *classifier rules* (configurações de política 3, 5, 6 e 7). Cada *classifier rule* nas políticas do exemplo exigem 24 mensagens para sua configuração. Desta forma, o gráfico abaixo poderia ser facilmente previsto.

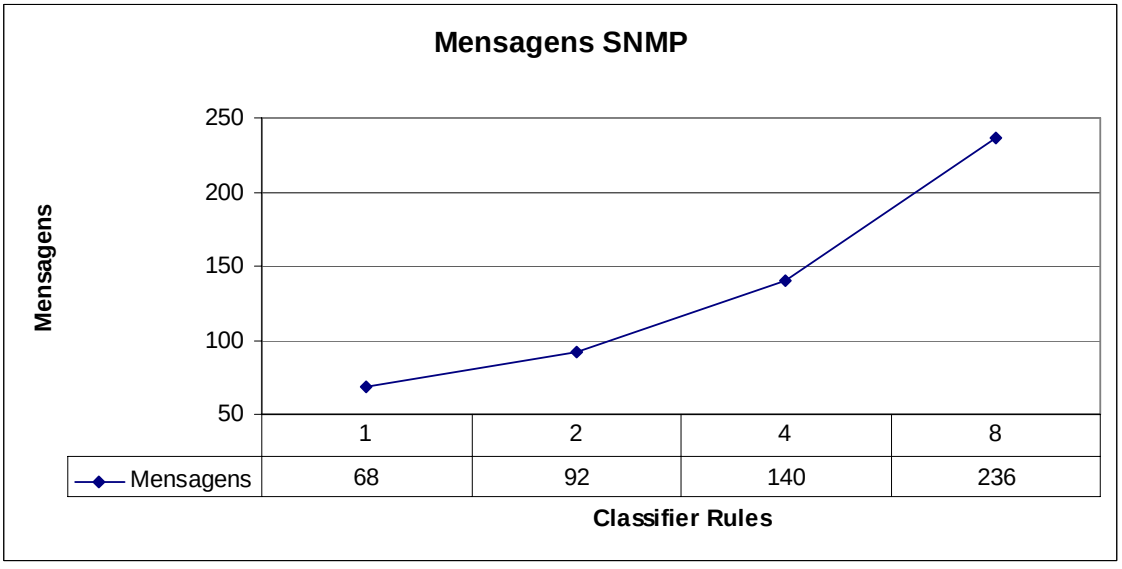


Figura 7.5: Número de mensagens SNMP em relação ao número de classifier rules

A Figura 7.6 apresenta a variação no tempo de resposta em função do número de *classifier rules*. Devido ao número de mensagens necessárias para sua configuração, o aumento no número de *classifier rules* consegue provocar um aumento perceptível no tempo de resposta, chegando a um pico de aproximadamente meio segundo para a configuração de política 7. Naturalmente que a influência depende de quantos objetos SNMP são necessários para configurar cada *classifier rule*.

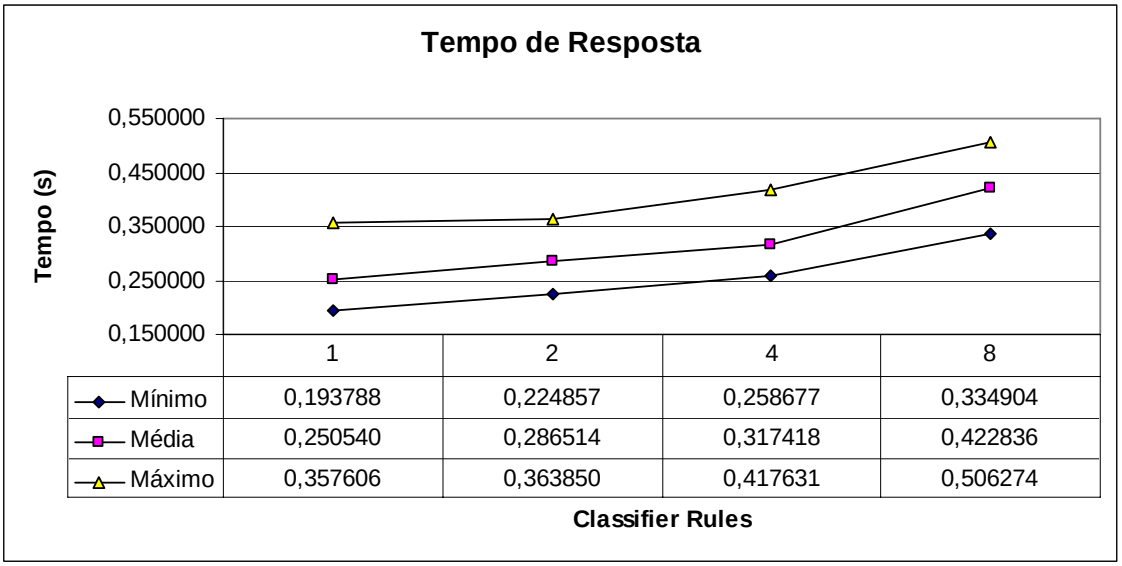


Figura 7.6: Tempo de resposta em relação ao número de *classifier rules*

A Figura 7.7 apresenta a variação do número de mensagens SNMP em relação ao número de políticas configuradas (configurações de política 1, 8, 9 e 10). Cada política do exemplo exige 30 mensagens para sua configuração. Desta forma, o gráfico abaixo poderia ser facilmente previsto.

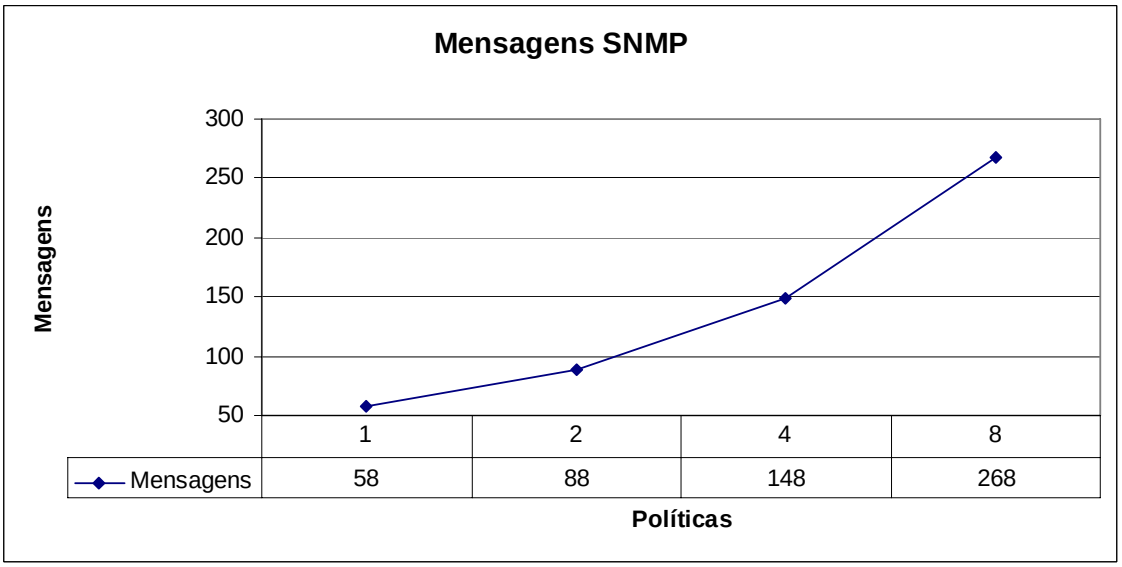


Figura 7.7: Número de mensagens SNMP em relação ao número de políticas

A Figura 7.8 apresenta a variação no tempo de resposta em função do número de políticas a serem configuradas. A influência de uma política adicional no tempo de resposta é altamente ligada às *classifier rules* desta política, visto que a maior parte das mensagens necessárias para configurar uma nova política são relativas a *classifier rules*. No exemplo, 24 das 30 mensagens que configuram uma política são relativas às *classifier rules*. Para a configuração de política 10, o pico no tempo de resposta chegou a quase 0,7s.

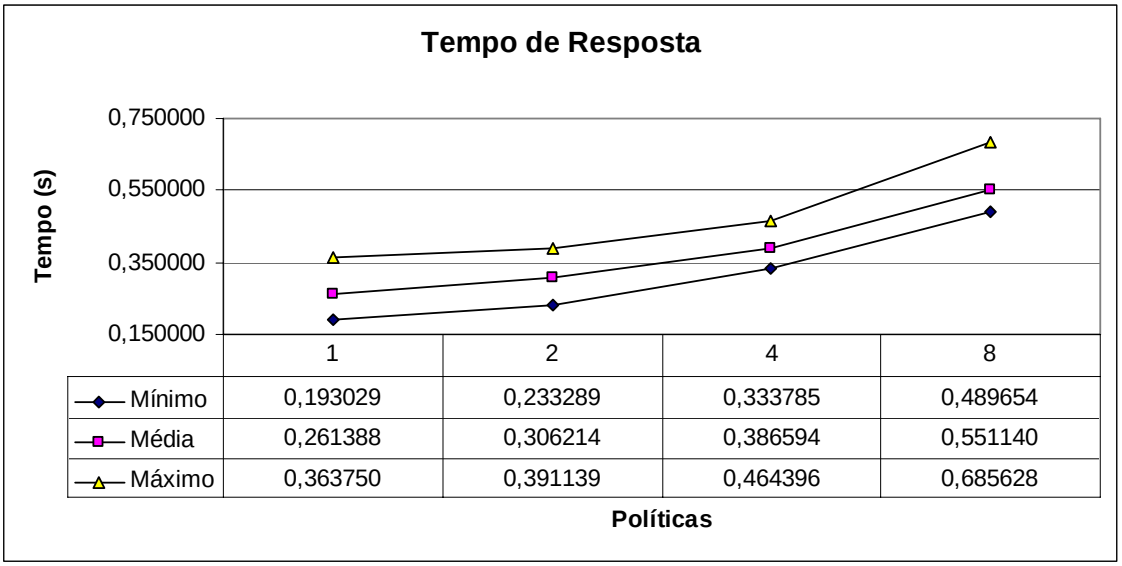


Figura 7.8: Tempo de resposta em relação ao número de políticas

Nos exemplos apresentados até então, todas as políticas compartilhavam a mesma definição de ação (ou classe de QoS). A Figura 7.9 apresenta a variação do número de mensagens SNMP conforme se aumenta o número de ações utilizadas (diminuindo o compartilhamento de ações entre políticas). As configurações de política relativas ao gráfico são as seguintes: 10, 11, 12 e 13. Cada ação do exemplo exige 20 mensagens para sua configuração. Desta forma, o gráfico abaixo poderia ser facilmente previsto.

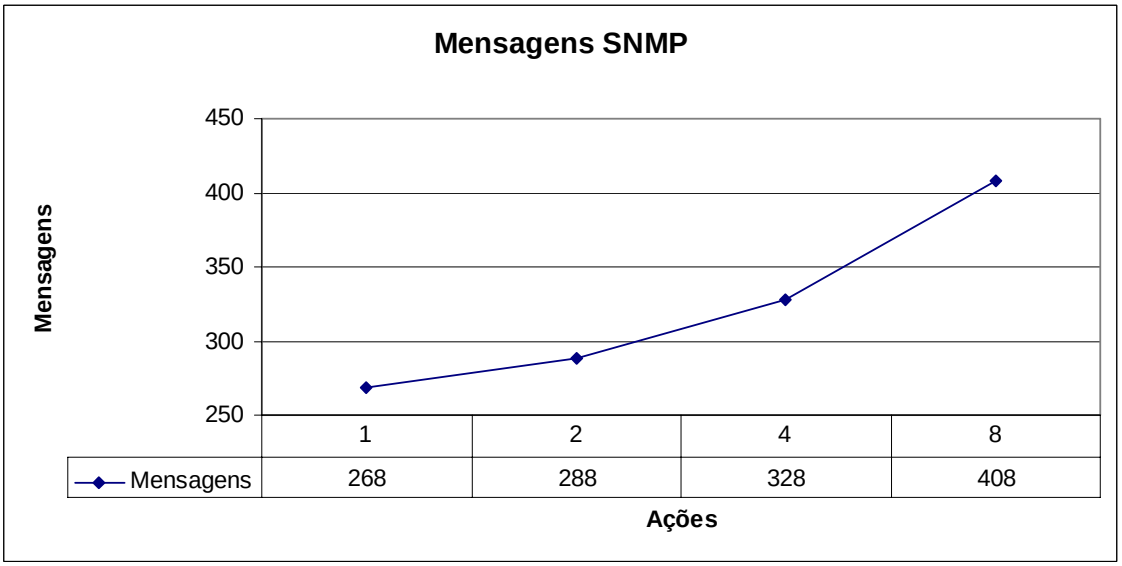
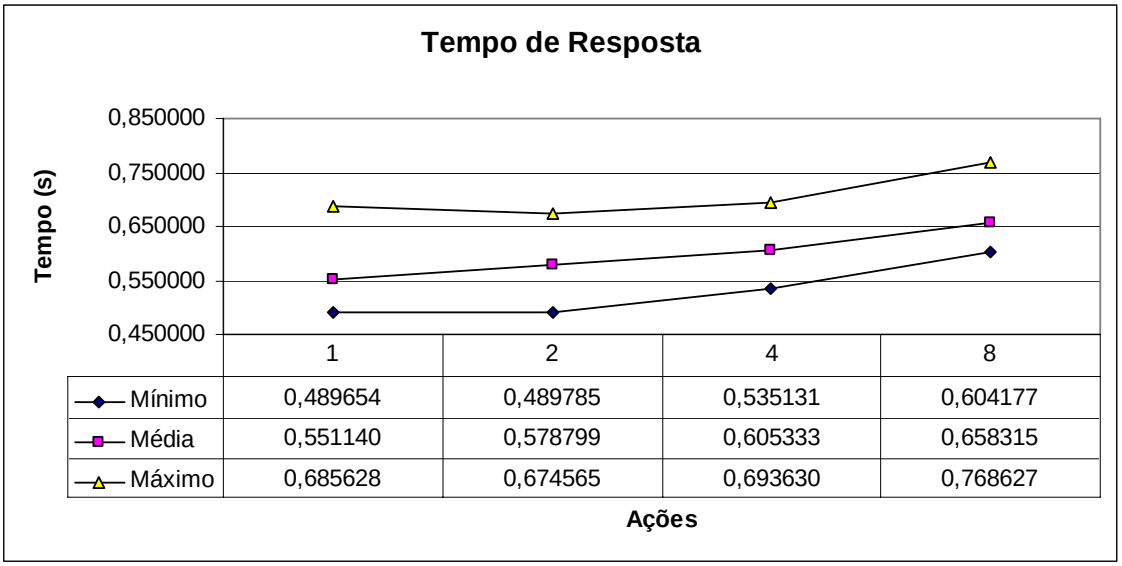


Figura 7.9: Número de mensagens SNMP em relação ao número de ações

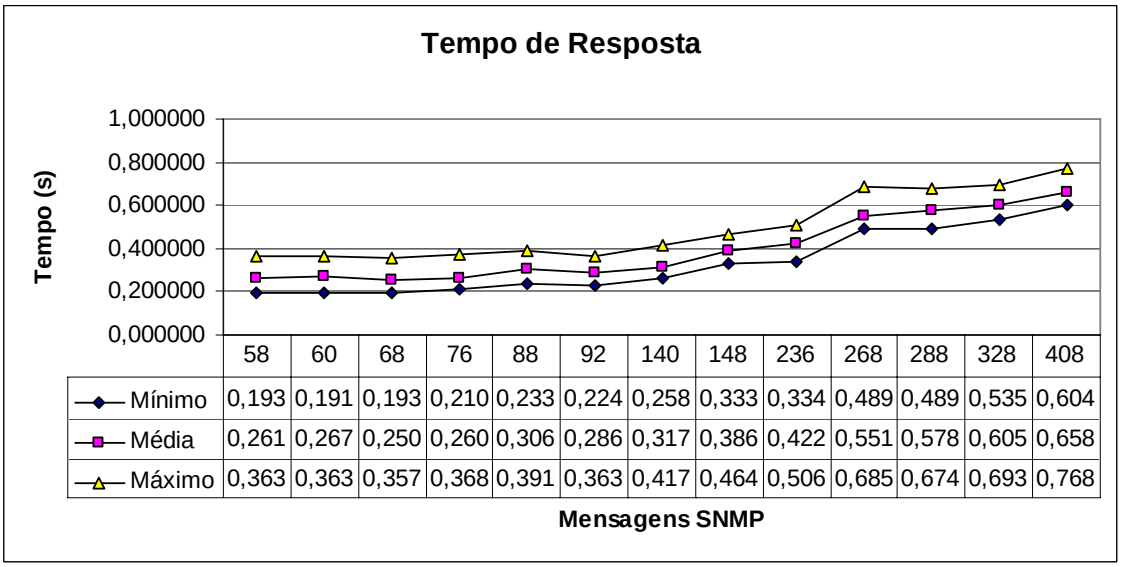
A Figura 7.10 apresenta a variação no tempo de resposta em função do número de ações a serem configuradas. A influência de uma ação adicional no tempo de resposta é semelhante à de uma *classifier rule* adicional nos exemplos considerados. Cada ação necessita de 20 mensagens adicionais, ao passo que cada *classifier rule* necessita de 24 mensagens adicionais. A configuração de política 13 apresenta o pico mais alto de todos no tempo de resposta (cerca de 0,77s.). Isto certamente está relacionado à utilização de oito políticas distintas, cada uma com sua própria definição de ação.



	1	2	4	8
Mínimo	0,489654	0,489785	0,535131	0,604177
Média	0,551140	0,578799	0,605333	0,658315
Máximo	0,685628	0,674565	0,693630	0,768627

Figura 7.10: Tempo de resposta em relação ao número de ações

Dado que o tempo de resposta está fortemente relacionado ao número de requisições SNMP necessárias para realizar a configuração, torna-se interessante traçar um gráfico relacionando estes dois elementos. A Figura 7.11 apresenta o tempo de resposta em função do número de mensagens SNMP necessárias para realizar a configuração.



	58	60	68	76	88	92	140	148	236	268	288	328	408
Mínimo	0,193	0,191	0,193	0,210	0,233	0,224	0,258	0,333	0,334	0,489	0,489	0,535	0,604
Média	0,261	0,267	0,250	0,260	0,306	0,286	0,317	0,386	0,422	0,551	0,578	0,605	0,658
Máximo	0,363	0,363	0,357	0,368	0,391	0,363	0,417	0,464	0,506	0,685	0,674	0,693	0,768

Figura 7.11: Tempo de resposta em relação ao número de mensagens SNMP

Visando uma melhor compreensão daquilo que compõe o tempo de resposta, analisou-se mais detalhadamente uma configuração típica realizada pelo PDP para a configuração de política 1. Por configuração típica entenda-se uma configuração cujo tempo de resposta ficou bastante próximo da média para a configuração de política 1. A Figura 7.12 ilustra a composição do tempo de resposta analisado. Os valores absolutos são dados em segundos.

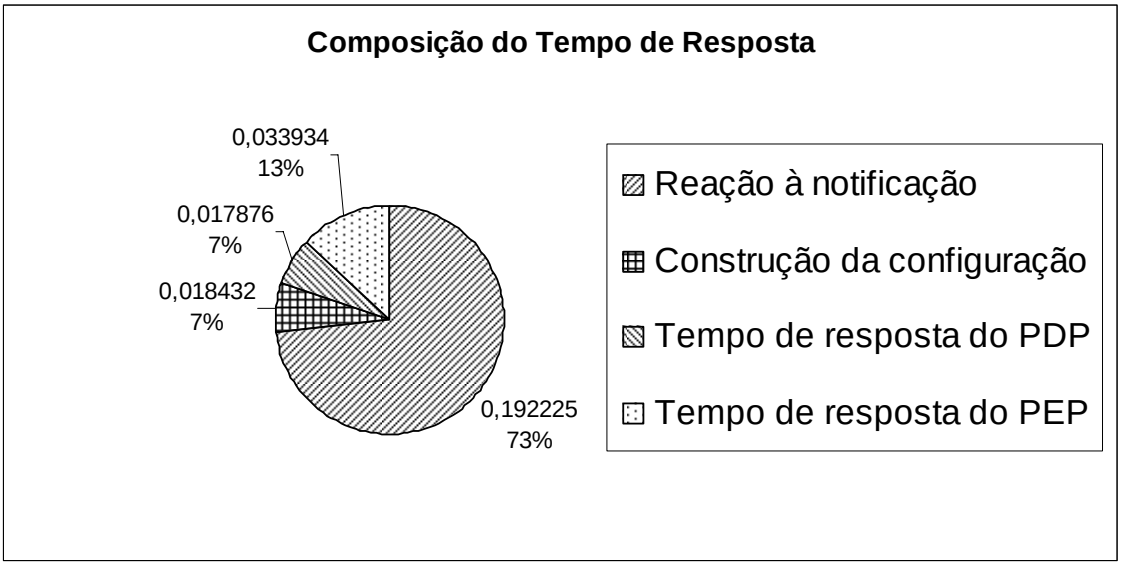


Figura 7.12: Composição do tempo de resposta em uma configuração típica para a configuração de política 1

Observa-se na Figura 7.12 que a maior parte do tempo de resposta considerado ocorre para que o PDP reaja à notificação de conexão da estação móvel. Este tempo, que é altamente dependente da implementação do PDP, foi medido entre a emissão da notificação de conexão e o início da busca de interfaces que realizada anteriormente à configuração da política. A implementação realizada realmente não é muito eficiente neste quesito: a recepção da notificação ocorre por um processo externo ao PDP; este dispara um segundo processo que repassa a notificação ao processo do PDP utilizando mecanismos de IPC; por fim, o processo do PDP somente verifica a recepção de notificações a cada décimo de segundo. Cabe notar que o tempo para reação à notificação é relativamente constante, não dependendo da configuração de política considerada.

O segundo maior tempo na Figura 7.12 é o “Tempo de resposta do PEP”. Este é a soma dos tempos medidos entre a chegada de uma mensagem SNMP e a emissão da respectiva resposta. Em média, o tempo de resposta do PEP para uma mensagem SNMP é de 1,17ms no caso considerado.

O “Tempo de resposta do PDP” na Figura 7.12 representa a soma dos tempos medidos entre a resposta de uma requisição SNMP e a requisição seguinte. Ele captura o tempo que o PDP leva para enviar a próxima requisição, dado que a requisição anterior foi concluída. Como a captura de pacotes ocorreu na máquina das estações base, este tempo também inclui os tempos de RTT (Round-trip Time). Cabe notar que este tempo não inclui a decisão de quais objetos SNMP devem ser configurados, pois a implementação do PDP primeiro decide o conjunto de configurações, e depois configura todas em lote. O tempo para decidir os objetos SNMP a serem configurados está representado no tempo de “Construção da configuração” na Figura 7.12, e foi medido entre o final da busca de interfaces e a primeira mensagem de configuração.

Embora isto não tenha ficado muito evidente nos experimentos, o RTT pode apresentar forte influência no tempo de resposta. Como, no cenário experimental utilizado, PDP e PEP estão na mesma rede local, o RTT acaba ficando muito pequeno. Medições com o utilitário ping indicaram uma média de 0,466ms para 21 mensagens,

quando medido do PEP para o PDP. Considerando 29 RTTs com este tempo médio na composição da Figura 7.12 teríamos cerca de 5% do tempo de resposta total. Se subtrairmos 27 RTTs do "Tempo de resposta do PDP", este ficaria somente em 5,294ms, representando somente 2,02% do tempo de resposta total.

Considerando o RTT médio informado no parágrafo anterior para cada duas mensagens da Figura 7.11, podemos estimar a curva média obtida para outros valores de RTT. Esta informação está representada na Figura 7.13, onde a legenda indica o RTT considerado. A mesma informação é detalhada para os RTTs até 16ms na Figura 7.14.

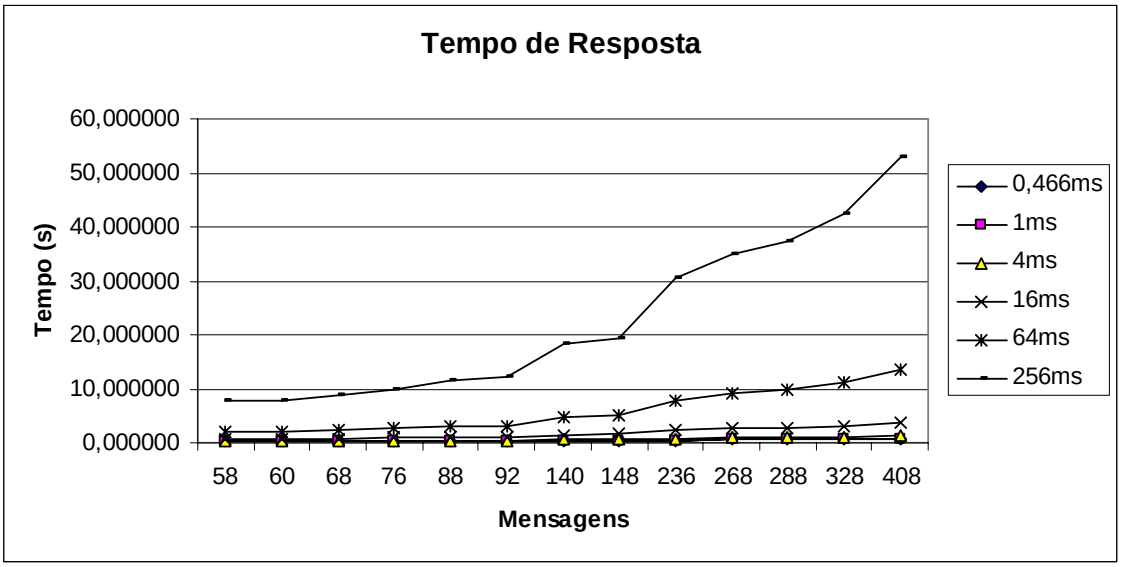


Figura 7.13: Estimativa do tempo de resposta em relação ao número de mensagens SNMP para diferentes valores de RTT (ver legenda)

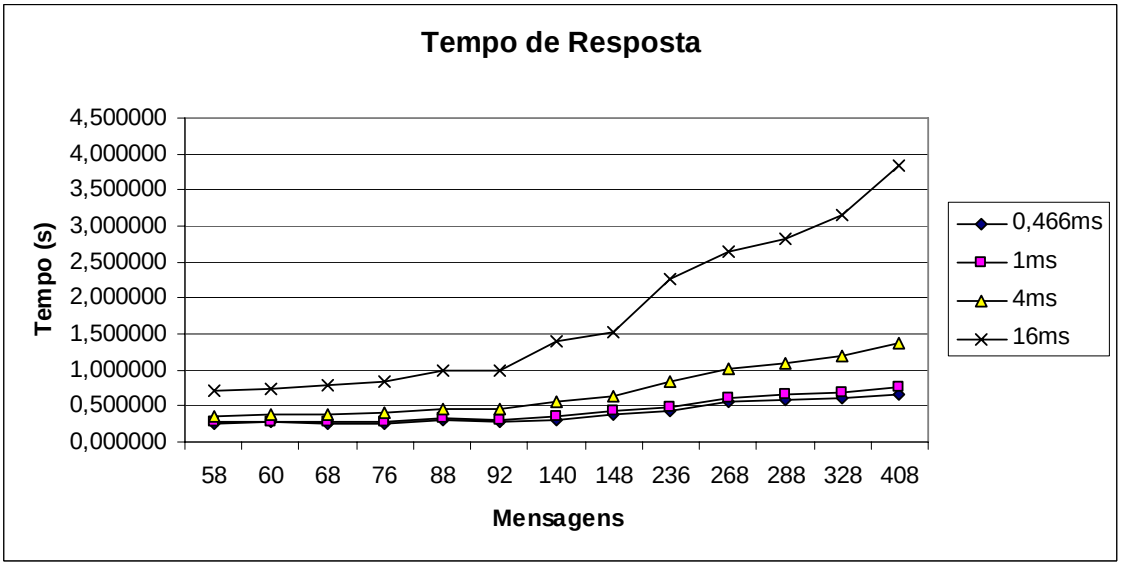


Figura 7.14: Estimativa do tempo de resposta em relação ao número de mensagens SNMP para diferentes valores de RTT (ver legenda) – detalhe

Observando a Figura 7.13 e a Figura 7.14, chega-se a conclusão que é necessário ter muita cautela e um bom planejamento ao utilizar o modelo de gerenciamento baseado na informação de conexão das estações móveis. Supondo que se deseje poder configurar políticas como a da configuração de política 13, que exige 408 mensagens para ser

configurada no PEP, e supondo-se, ainda, que o tempo de resposta do PDP após uma conexão de estação móvel não possa ser superior a 1s, temos o fato de que seria inviável que o PDP e o PEP apresentassem um RTT superior a 2,14ms no pior caso, o que impõe uma séria restrição. Se considerarmos a diferença de tempo entre o pior caso e o caso médio para 408 mensagens da Figura 7.11, o RTT médio teria que ser de, no máximo, 1,6ms. Para apresentar este RTT, o PDP e o PEP teriam que estar, praticamente, na mesma rede local. A Figura 7.15 faz uma análise do valor de RTT no pior caso para diferentes tempos de resposta limites, considerando a configuração de política 13.

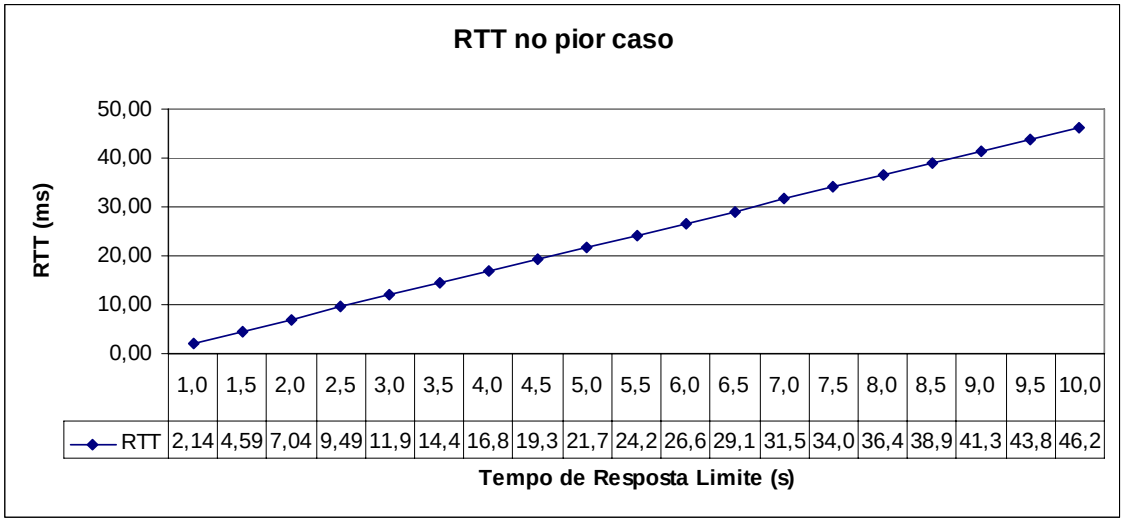


Figura 7.15: RTT no pior caso em função do tempo de resposta limite

Há alternativas, entretanto, para possibilitar um afastamento maior entre o PDP e o PEP. Embora o protocolo SNMP permita a configuração de mais de um objeto na mesma requisição, a implementação de PDP realizada somente envia uma configuração por mensagem (devido a limitações do PHP). Cada objeto SNMP a ser configurado exige que o PDP envie uma requisição *snmpset* e aguarde até receber a resposta antes de enviar a próxima. Uma implementação de PDP mais eficiente poderia enviar todo um registro de tabela SNMP por requisição e diminuir drasticamente o número de mensagens.

Enviando um registro por requisição SNMP, no caso da configuração de política 13, o número de mensagens necessárias à configuração cairia de 408 para 72, viabilizando RTTs da ordem de 16ms nas condições estabelecidas anteriormente. Esta conclusão foi obtida observando o tempo de resposta para 76 mensagens na Figura 7.14. Certamente que esta conclusão precisaria ser validada através de experimentação com um PDP que possua a implementação sugerida.

Para fechar a avaliação sobre tempo de resposta e número de mensagens, a Tabela 7.3 apresenta os valores medidos para as configurações de política 14, 15 e 16, que representam configurações de política mais usuais. Os tempos de resposta não diferem muito daqueles obtidos em configurações de política com número de mensagens próximo aos apresentados, embora os valores para as configurações de política 15 e 16 tenham ficado um pouco acima do esperado.

Tabela 7.3: Medições para as configurações de política 14, 15 e 16

Configuração	Número de Mensagens	Tempo de Resposta (s)		
		Mínimo	Média	Máximo
14	88	0,236733	0,309386	0,409429
15	138	0,292939	0,367811	0,501816
16	206	0,383818	0,449715	0,554215

O consumo de banda será apresentado apenas para as configurações de política 13 e 16, visto que estas representam as configurações mais estressantes para a rede em suas categorias.

Assim como a definição dos *scripts* de simulação, o padrão de consumo de banda nos experimentos realizados é cíclico, sendo caracterizado por rajadas nos momentos em que ocorrem eventos relevantes na rede. A Figura 7.16 apresenta o padrão de consumo de banda no experimento com a configuração de política 13.

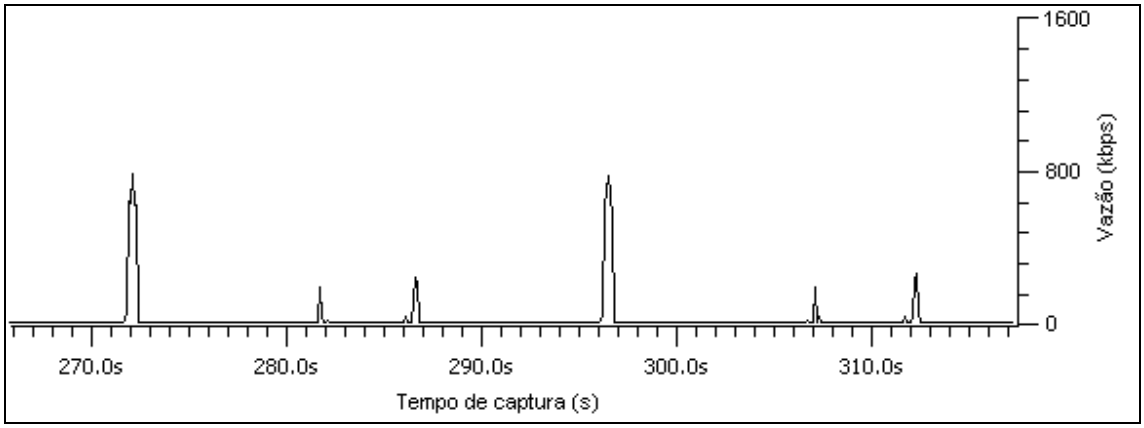


Figura 7.16: Padrão de consumo de banda no experimento com a configuração de política 13

Observa-se na Figura 7.16 um padrão cíclico em que há um pico maior seguido por dois picos menores. O pico maior representa o momento em que o PDP realiza a configuração das políticas para a estação móvel, ocorrendo após a conexão da mesma em uma estação base. O primeiro pico menor é o momento em que o *handover* ocorre. Neste momento, o tráfego na rede é causado por notificações enviadas pela estação base para o PDP. Observa-se que, no *script* de simulação criado, configurou-se o *handover* para que ocorresse 10s após a conexão, sendo este o limite de tempo de resposta aceitável nos experimentos. O segundo pico menor é o momento em que a estação móvel desconecta-se de sua estação base original, isto é, a estação base original descarta o estado mantido para a estação móvel. Neste momento, ocorre a desconfiguração das políticas da estação móvel pelo PDP.

O maior consumo de banda ocorre nos momentos em que o PDP realiza a configuração das políticas. Nestes momentos, observa-se uma média de aproximadamente 320kbps em cada sentido durante cerca de meio segundo (totalizando os cerca de 640kbps nos dois sentidos observáveis na Figura 7.16). Isto decorre dos certa de 40000 bytes que necessitam ser transmitidos nos dois sentidos ao longo do tempo de resposta do PDP.

Mais uma vez, é necessário um bom planejamento para garantir a correta operação do sistema de gerenciamento. Seria possível diminuir a banda consumida aumentando-se o tempo de resposta. Entretanto, isto pode não ser aceitável, como analisado

anteriormente. Devem-se projetar os enlaces entre o PDP e o PEP de modo que a banda necessária ao sistema de gerenciamento seja suprida. A exigência de banda poderia ser ainda maior se várias configurações pudessem ser realizadas em paralelo (o que não é o caso do PDP implementado).

A Figura 7.17 apresenta o padrão de consumo de banda no experimento com a configuração de política 16. Não há grande diferença do padrão observado para a configuração de política 13. O mais importante a ser reparado é que, como o tempo de resposta para esta configuração de política é menor, a taxa de 320kbps em cada sentido é mantida por apenas cerca de 0,2s. Isto decorre da menor quantidade de dados a serem transmitidos.

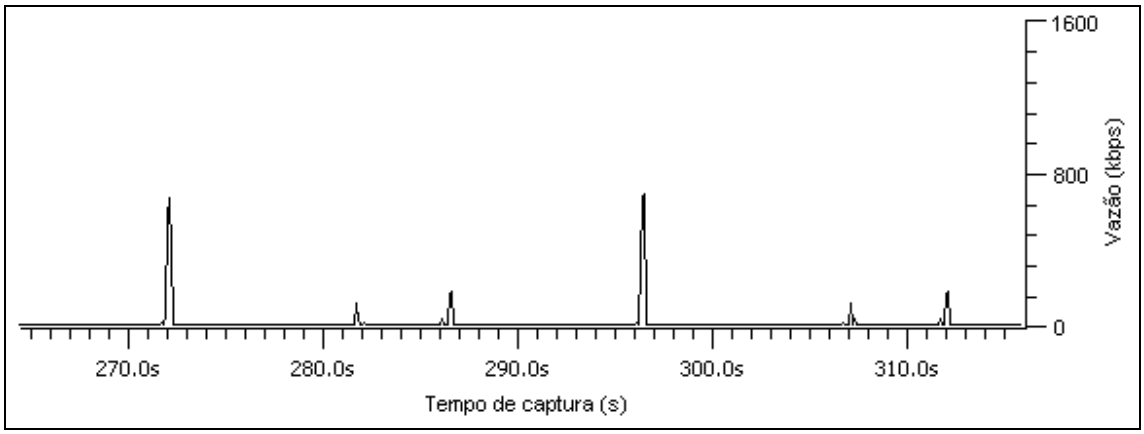


Figura 7.17: Padrão de consumo de banda no experimento com a configuração de política 16

7.2.2 Avaliação do tempo de reação à inversão de prioridade

Esta subseção visa responder à quarta questão listada no início da seção corrente: quanto tempo o PDP leva para reconfigurar os Service Flows em função da rejeição de um fluxo associado a uma política de alta prioridade? Novamente, somente duas estações base foram necessárias para a avaliação.

Os *scripts* de simulação são praticamente os mesmos da subseção anterior. A diferença é que, além da estação móvel que fica realizando *handover* constantemente, duas outras estações cliente (fixas) entram na rede ao início da simulação, uma em cada estação base, e permanecem registradas nestas estações até o final da simulação.

As estações fixas têm configuradas para si políticas de baixa prioridade, enquanto a estação móvel tem configurada para si uma política de alta prioridade. A simulação ocorre de tal forma que é impossível uma estação base atender simultaneamente às políticas da estação fixa e da estação móvel. Desta forma, sempre que a estação móvel realiza *handover*, um processo de adaptação da configuração na rede deve de ocorrer.

As configurações de política utilizadas estão descritas a seguir. As taxas de transmissão foram simuladas propositalmente baixas para causar o efeito desejado.

- Para as estações fixas, foi preparada uma política de *download* de FTP com banda mínima e máxima de 600kbps e prioridade de escalonamento 5. Filtro realizado pelo MAC das estações de destino e pela porta de destino 20; A política foi aplicada em ambas as estações base e foi marcada com prioridade 0. As taxas de transmissão simuladas para as estações fixas foram de 1Mbps em

cada sentido, o que exigirá pelo menos 60% de utilização dos recursos da rede para garantir a QoS;

- Para a estação móvel, foi utilizada a configuração de política de VoIP listada sob o número 14 na subseção anterior. Adicionalmente, as políticas foram marcadas com prioridade 1. As taxas de transmissão simuladas para a estação móvel foram de 200kbps em cada sentido, o que exigirá pelo menos 50% de utilização dos recursos da rede para garantir a QoS.

Os eventos da simulação relativos ao experimento podem ser vistos no Quadro 7.2. A notação é a mesma utilizada na subseção anterior.

```
#Inicio da execução dos simuladores
coldStart BS1: (1) 0.000000
coldStart BS2: (2) 1.478075

#Configuração da política pelo sistema de gerenciamento
Busca de interfaces na BS1: (3) 12.054051 -> (42)
    46.894768: 40 mensagens
Inicialização BS1: (43) 64.578611 -> (62) 64.612843: 20
    mensagens
Busca de interfaces na BS2: (63) 71.992635 -> (102)
    90.341424: 40 mensagens
Inicialização BS2: (103) 96.456542 -> (122) 96.491639: 20
    mensagens

#Entrada da estação 1 (fixa) na rede pela BS1
Conexão SS1<->BS1: (123) 200.160536
Registro SS1<->BS1: (124) 200.161175
Configuração dos fluxos da SS1 na BS1: (125) 200.339139 ->
    (181) 200.415141: 56 mensagens
Admissão de serviço da SS1 na BS1: (180) 200.414810
Busca das taxas de serviço da SS1 na BS1: (182) 200.581780
    -> (185) 200.584060: 4 mensagens

#Entrada da estação 2 (fixa) na rede pela BS2
Conexão SS2<->BS2: (186) 201.635278
Registro SS2<->BS2: (187) 201.635927
Configuração dos fluxos da SS2 na BS2: (188) 201.793226 ->
    (244) 201.872243: 56 mensagens
Admissão de serviço da SS2 na BS2: (243) 201.871898
Busca das taxas de serviço da SS2 na BS2: (245) 202.027928
    -> (248) 202.029982: 4 mensagens

#Entrada da estação 3 (móvel) na rede pela BS1
Conexão SS3<->BS1: (249) 260.212336
Registro SS3<->BS1: (250) 260.212930
Configuração dos fluxos da SS3 na BS1: (251) 260.414275 ->
    (340) 260.560630: 88 mensagens
Admissão de serviço de download da SS3 na BS1: (308)
```

```
    260.507584
Rejeição de serviço de upload da SS3 na BS1: (339)
    260.560299
Busca das taxas de serviço da SS3 na BS1: (341) 260.669018
    -> (344) 260.671233: 4 mensagens
Cancelamento de serviço da SS1 na BS1: (345) 260.897654 ->
    (347) 260.900204: 2 mensagens
Admissão de serviço de upload da SS3 na BS1: (346)
    260.899909

#Handover da SS3 para a BS2
Conexão SS3<->BS2: (348) 271.694576
Configuração dos fluxos da SS3 na BS2: (349) 271.978870 ->
    (436) 272.115012: 88 mensagens
Registro SS3<->BS2: (437) 281.703153
Admissão de serviço de download da SS3 na BS2: (438)
    281.703782
Rejeição de serviço de upload da SS3 na BS2: (439)
    281.704469
Busca das taxas de serviço da SS3 na BS2: (440) 281.834078
    -> (443) 281.836800: 4 mensagens
Cancelamento de serviço da SS2 na BS2: (444) 282.173039 ->
    (446) 282.175027: 2 mensagens
Admissão de serviço de upload da SS3 na BS2: (445)
    282.174743
Indicação de handover da SS3 na BS1: (447) 282.230818
Desregistro SS3<->BS1: (448) 286.234429
Desconexão SS3<->BS1: (449) 286.235029
Normalização de serviço da SS1 na BS1: (450) 286.413292 ->
    (452) 286.415603: 2 mensagens
Admissão de serviço da SS1 na BS1: (451) 286.415307
Desconfiguração dos fluxos da SS3 na BS1: (453) 286.541709
    -> (474) 286.608498: 22 mensagens

#Handover da SS3 para a BS1
Conexão SS3<->BS1: (475) 296.242840
Configuração dos fluxos da SS3 na BS1: (476) 296.358142 ->
    (563) 296.486599: 88 mensagens
Indicação de handover da SS3 na BS2: (564) 306.724271
Registro SS3<->BS1: (565) 307.252215
Admissão de serviço de download da SS3 na BS1: (566)
    307.252887
Rejeição de serviço de upload da SS3 na BS1: (567)
    307.253569
Busca das taxas de serviço da SS3 na BS1: (568) 307.464873
    -> (571) 307.467064: 4 mensagens
Cancelamento de serviço da SS1 na BS1: (572) 307.732119 ->
    (574) 307.734292: 2 mensagens
Admissão de serviço de upload da SS3 na BS1: (573)
```

```
    307.733975
Desregistro SS3<->BS2: (575) 311.728501
Desconexão SS3<->BS2: (576) 311.729060
Normalização de serviço da SS2 na BS2: (577) 311.906318 ->
    (579) 311.908713: 2 mensagens
Admissão de serviço da SS2 na BS2: (578) 311.908448
Desconfiguração dos fluxos da SS3 na BS2: (580) 311.908713
    -> (601) 312.095944: 22 mensagens

# Uma série de handovers ocorre até o final da simulação
# (...)
```

Quadro 7.2: Eventos SNMP para a avaliação do tempo de reação à inversão de prioridade

Alguns eventos são novos em relação aos vistos na subseção anterior. Estes eventos estão descritos na Tabela 7.4.

Tabela 7.4: Descrição dos eventos SNMP para a avaliação do tempo de reação à inversão de prioridade

Evento	**Descrição**
Rejeição de serviço	*Trap* que indica que um *Service Flow* foi rejeitado na rede
Cancelamento de serviço	Mensagem *snmpset* e resposta para marcar um *Service Flow* como *provisioned*. A admissão do serviço é cancelada administrativamente com esta ação.
Normalização de serviço	Mensagem *snmpset* e resposta para marcar um *Service Flow* como *active*. A admissão do serviço é reativada administrativamente com esta ação.

As medições para os 21 casos de inversão de prioridade apontaram uma média de 464,545ms entre a emissão da notificação que indica a rejeição de um *Service Flow* de alta prioridade e a emissão da notificação de que o mesmo foi admitido em função da atuação do PDP. O menor tempo ficou em 339,610ms e o maior tempo ficou em 559,323ms. A Figura 7.18 faz uma análise da composição do tempo de resposta para um caso típico, cujo tempo de resposta fica próximo à média. Os valores absolutos são dados em segundos.

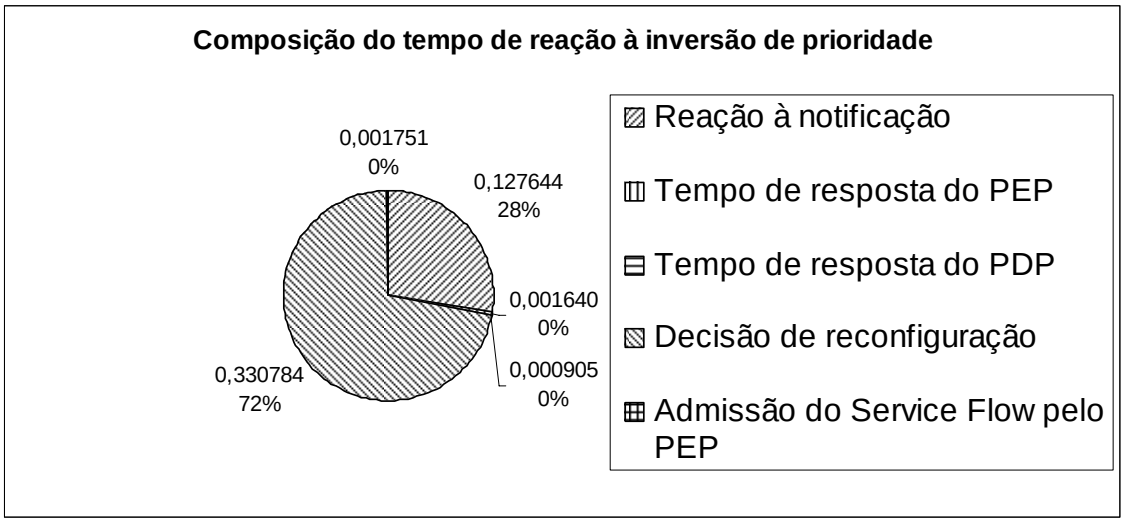


Figura 7.18: Composição do tempo de reação à inversão de prioridade para um caso típico

Os três primeiros tempos constantes na legenda da Figura 7.18 já foram apresentados na subseção anterior, inclusive o motivo pelo qual a "Reação à notificação" é relativamente demorada. Os tempos de resposta estão associados à busca das taxas de serviço pelo PDP, decorrente da notificação de registro da estação móvel que ocorre imediatamente antes das notificações sobre o estado de admissão dos *Service Flows*.

O tempo de "Admissão do *Service Flow* pelo PEP" é o tempo que o PEP demora a admitir o *Service Flow* de alta prioridade após a solicitação de cancelamento da admissão de *Service Flows* ativos de baixa prioridade pelo PDP. Este tempo não é muito significativo, como se pode ver na Figura 7.18. O tempo exigido para que o PDP cancele administrativamente a admissão de *Service Flows* é relativamente pequeno, visto que basta um *snmpset* para cada *Service Flow*. Em muitos casos, um *Service Flow* pode ser suficiente para liberar os recursos necessários ao fluxo de alta prioridade.

A maior parcela do tempo de reação à inversão de prioridade é dada pela "Decisão de reconfiguração". É um tanto surpreendente que o PDP esteja levando 330ms para decidir como liberar os recursos para os fluxos de uma política de alta prioridade. Certamente este tempo pode ser contornado com uma implementação mais eficiente do PDP.

A conclusão a que se chega após a análise do tempo de reação à inversão de prioridade é que ele é altamente dependente da implementação do PDP. Implementações mais eficientes podem levar este tempo ao mínimo, evitando a rejeição prolongada de políticas de alta prioridade.

7.2.3 Avaliação do tempo de resposta com várias estações

Esta subseção visa responder à última questão listada no início da seção corrente: qual seria o tempo de resposta em um cenário em que há várias estações se movimentando aleatoriamente pela rede? As três estações base foram utilizadas neste experimento.

Criar três *scripts* de simulação coerentes para as estações base, simulando múltiplas estações móveis, e ainda introduzindo um certo grau de aleatoriedade é uma tarefa bastante complexa. Para tanto, foi criado um utilitário em Java que, dado um conjunto de parâmetros, gera *scripts* de simulação automaticamente. Os parâmetros considerados no utilitário são apresentados na Tabela 7.5. O escopo de randomização, nos casos em que um parâmetro é descrito por valor mínimo e valor máximo, também é apresentado na tabela.

Tabela 7.5: Parâmetros para a criação dos *scripts* de simulação

Parâmetro	Valor	Randomização
Número de estações base	3	-
Número de estações móveis	5	-
Tempo total de simulação	800s	-
Taxa mínima de upload	20000kbps	por estação móvel
Taxa máxima de upload	30000kbps	
Taxa mínima de download	20000kbps	por estação móvel
Taxa máxima de download	30000kbps	

Tempo de simulação mínimo em que as estações se conectarão à rede	200s	por estação móvel
Tempo de simulação máximo em que as estações se conectarão à rede	210s	
Atraso entre a primeira conexão à rede e o respectivo registro da estação	0s	-
Tempo mínimo de permanência da estação móvel em uma estação base	15s	por entrada em estação base
Tempo máximo de permanência da estação móvel em uma estação base	30s	
Tempo mínimo entre a conexão na estação base destino e a ocorrência do *handover*	5s	por *handover*
Tempo máximo entre a conexão na estação base destino e a ocorrência do *handover*	10s	
Tempo de manutenção do estado da estação móvel pela estação base de origem de um *handover*	10s	-
Tempo entre o desregistro na estação base de origem de um *handover* e a respectiva desconexão	0s	-

As taxas de transmissão foram indicadas com valores maiores desta vez visando evitar problemas com rejeição de fluxos. Note que os valores indicados não estão fora do razoável para uma rede IEEE802.16, dada a estimativa apresentada na seção “Visão Geral de IEEE802.16” de até 70Mbps em condições favoráveis. Não se tem uma estimativa do caso médio para as taxas de transmissão, sendo que este valor somente pode ser obtido com experimentos em ambientes reais. O tempo de simulação em que as estações se conectam à rede é propositalmente alto para permitir a configuração das políticas antes de iniciarem os eventos relativos às estações. O tempo total de simulação de eventos de estações é de aproximadamente 600s.

A mesma política foi configurada para todas as estações. Trata-se de uma política de VoD igual à listada sob o número 1 na subseção “Avaliação de tempo de resposta, número de mensagens e banda consumida”. A única diferença com relação aos parâmetros originais é que não foi incluído filtro pelo MAC das estações de destino, indicando que a política deveria ser aplicada a qualquer estação que se conectasse à rede. A política foi configurada nas três estações base.

Foram escolhidas aleatoriamente 21 notificações de conexão ao longo da simulação para realizar a avaliação. Os tempos de resposta obtidos neste experimento são apresentados na Tabela 7.6 em comparação a seus tempos correspondentes obtidos para uma única estação na rede, conforme apresentados para a configuração de política 1 na subseção “Avaliação de tempo de resposta, número de mensagens e banda consumida”.

Tabela 7.6: Tempo de resposta para uma e várias estações móveis na rede

Experimento	**Tempo de Resposta (s)**		
	Mínimo	**Média**	**Máximo**
VoD com uma estação	0,193029	0,261388	0,363750
VoD com múltiplas estações	0,179105	0,264505	0,531123

Em relação aos resultados obtidos para uma estação, observa-se uma maior dispersão do tempo de resposta. Dois efeitos contribuíram para que isto ocorresse: diminuição das configurações de ações e possibilidade de espera para configuração.

O efeito de diminuição das configurações de ações provocou uma redução no tempo de resposta (daí o mínimo ser menor do que para uma estação). Em muitos casos, a *Service Class* já estava configurada no PEP por conta de outra estação no momento em que uma nova estação móvel se conectou à estação base. Deste efeito pode-se deduzir que uma forma de reduzir o tempo de resposta é deixar pré-configuradas no PEP todas as *Service Classes* relevantes, de modo que somente a configuração das políticas seja feita no momento da conexão.

O efeito da possibilidade de espera para configuração já era previsto, e provocou um aumento no tempo de resposta. A implementação que se fez do PDP realiza as configurações seqüencialmente. Desta forma, se dois eventos ocorrem na rede ao mesmo tempo, um deles terá de esperar o termino da reação do PDP ao outro para que a reação a si comece a ser processada. Este é o caso quando duas estações móveis conectam-se em estações base (distintas ou não) ao mesmo tempo. Descontando-se o tempo de reação à notificação do PDP, o efeito da possibilidade de espera para configuração tem o potencial de dobrar, triplicar ou multiplicar por um “n” qualquer o tempo de resposta, dependendo de quantas configurações anteriores terão de ser esperadas.

O efeito da possibilidade de espera para configuração pode ser tratado de duas maneiras: com uma implementação mais eficiente do PDP ou estatisticamente. Se a abordagem de implementar um PDP mais eficiente for escolhida, deve-se incluir no PDP a capacidade de realizar duas ou mais configurações simultaneamente, eliminando o risco de espera para configuração.

Na abordagem estatística, deve-se determinar qual seria o maior número de eventos simultâneos esperados para a rede dimensionada, considerando um percentual definido do tempo (ex. 99,99% do tempo). Para este número de eventos simultâneos, ajusta-se o sistema de gerenciamento de modo que o tempo de resposta decorrente seja aceitável.

8 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo apresenta as principais conclusões obtidas ao longo da realização deste trabalho e sugere trabalhos futuros. Algumas conclusões são resgatadas da seção “Conclusões obtidas após as implementações”, visto que ela as apresentou antecipadamente.

Como mencionado anteriormente, apesar de o gerenciamento baseado em políticas facilitar o gerenciamento de uma rede IEEE802.16, o gerenciamento de redes IEEE802.16 apresenta uma série de particularidades que devem ser tratadas e que podem forçar o sistema de gerenciamento baseado em políticas a perder um pouco de sua generalidade.

A perda de generalidade de um sistema de gerenciamento baseado em políticas ocorre de dois modos: políticas genéricas que não podem ser implementadas em todas as tecnologias e falta de suporte nas políticas a exigências específicas de determinadas tecnologias. De um modo geral, ao tentar manter a generalidade, o gerenciamento baseado em políticas pode impedir que se explore toda a capacidade de uma dada tecnologia; ao tratar as extensões específicas, o sistema de gerenciamento perde a generalidade. Como mencionado, talvez o melhor seja permitir a utilização de extensões específicas de cada tecnologia, sempre tentando ser o mais genérico possível, e valer-se principalmente da maior facilidade de expressão e gerenciamento integrado existentes em sistemas de gerenciamento baseado em políticas.

Observa-se que a exigência de um tratamento adequado às situações de *handover* em redes IEEE802.16 é uma das características que mais afeta o sistema de gerenciamento. A utilização do mecanismo baseado no *ranging* simplifica muito este tratamento ao permitir que o PDP preocupe-se com cada estação base isoladamente. Por outro lado, o mecanismo baseado em *ranging* também impõe suas limitações, que decorrem principalmente de restrições de tempo real no momento em que as estações móveis conectam-se a uma estação base. É necessário um bom planejamento do sistema de gerenciamento para garantir que o mesmo funcione adequadamente na prática.

A criação de um PDP mais eficiente é algo que poderia contribuir para a utilização do mecanismo baseado em *ranging*. Idéias como a configuração de vários objetos SNMP através de uma única mensagem, a realização de diversas configurações em paralelo e a pré-configuração das *Service Classes* nas estações base podem contribuir para reduzir o tempo de resposta do PDP. Sugere-se explorar estas alternativas em trabalhos futuros, bem como as melhorias na MIB sugeridas na seção “Conclusões obtidas após as implementações”. Também se poderia explorar a utilização de outros protocolos na comunicação entre o PDP e o PEP.

Algo importante a ser feito, mas que exigiria o uso de uma instalação real de rede IEEE802.16, é a caracterização estatística do tempo entre o *ranging* e o *registering* de uma estação móvel em processo de *handover*. A determinação deste tempo na prática é que permitiria validar ou invalidar a utilização do gerenciamento baseado no *ranging* como foi proposto. Este é o tempo que impõe o limite de tempo de resposta do PDP. Infelizmente não se pôde medi-lo neste trabalho.

Também sugere-se como trabalho futuro a implementação da alternativa de gerenciamento baseada no *registering*, que foi apresentada na seção “Arquitetura de gerenciamento proposta”. Esta abordagem alternativa não apresenta as mesmas restrições de tempo real que são encontradas na abordagem baseada no *ranging*.

Por questões de prazo, não foi possível validar a abordagem de tratamento para situações de degradação sugerida na seção “Gerenciamento de QoS baseado em políticas”. Trata-se de uma abordagem que pode facilitar o gerenciamento destas situações, de modo que pode ser um ótimo assunto para trabalhos futuros.

Por fim, observa-se que o QAME apresenta bastante espaço para evoluir. Uma idéia para evolução seria implementar uma hierarquia entre os fluxos, permitindo que fluxos fossem compostos por outros fluxos. Na versão atual do sistema, fluxos somente podem ser agrupados nas próprias políticas, o que se traduz em maior dificuldade de gerenciamento.

REFERÊNCIAS

ALAVI, H.; MOJDEH, M.; YAZDANI, N. A Quality of Service Architecture for IEEE 802.16 Standards. In: ASIA-PACIFIC CONFERENCE ON COMMUNICATIONS, APCC, 2005. **Proceedings…** [S.l.:s.n.], 2005.

BLAKE, K. et al. **An architecture for Differentiated Services**: RFC 2475. [S.l.]: Internet Engineering Task Force, Network Working Group, 1998.

BRADEN, R.; CLARK, D.; SHENKER, S. **Integrated Services in the Internet Architecture**: an Overview: RFC 1633. [S.l.]: Internet Engineering Task Force, Network Working Group, 1994.

BRADEN, R. et al. **Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification**: RFC 2205. [S.l.]: Internet Engineering Task Force, Network Working Group, 1997.

BIESZCZAD, A.; PAGUREK, B.; WHITE, T. Mobile Agents for Network Management. **IEEE Communications Surveys**, [S.l.], v. 1, n. 1, 1998.

CASE, J. et al. **A Simple Network Management Protocol (SNMP)**: RFC 1157. [S.l.]: Internet Engineering Task Force, Network Working Group, 1990.

CHAN, K. et al. **COPS usage for Policy Provisioning (COPS-PR)**: RFC 3084. [S.l.]: Internet Engineering Task Force, Network Working Group, 2001.

CHEN, J.; CHI, C.; GUO, Q. A Bandwidth Allocation Model with High Concurrence Rate in IEEE802.16 Mesh Mode. In: ASIA-PACIFIC CONFERENCE ON COMMUNICATIONS, APCC, 2005. **Proceedings…** [S.l.:s.n.], 2005.

CHEN, J.; JIAO, W.; GUO, Q. An Integrated QoS Control Architecture for IEEE 802.16 Broadband Wireless Access Systems. In: IEEE Globecom, 2005. **Proceedings...** [S.l.:s.n.], 2005.

CHEN, J.; JIAO, W.; WANG, H. A Service Fow Management Strategy for IEEE 802.16 Broadband Wireless Access Systems in TDD Mode. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, ICC, 2005. **Proceedings…** [S.l.:s.n.], 2005.

CHOI, S. et al. Fast Handover Scheme for Real-Time Downlink Services in IEEE 802.16e BWA System. In: IEEE VEHICULAR TECHNOLOGY CONFERENCE, VTC, 61., 2005. **Proceedings…** [S.l.:s.n.], 2005.

CHU, G.; WANG, D.; MEI, S. A QoS Architecture for the MAC Protocol of IEEE 802.16 BWA System. In: IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, CIRCUITS AND SYSTEMS AND WEST SINO EXPOSITIONS, 2002. **Proceedings…** [S.l.:s.n.], 2002.

CICCONETTI, C. et al. Quality of Service Support in IEEE 802.16 Networks. **IEEE Network**, [S.l.], v. 20, n. 2, p. 50-55, Apr. 2006.

CISCO SYSTEMS. **Quality of Service for Voice over IP**. [S.l.], 2001. Disponível em: <http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/qossol/qosvoip.pdf>. Acesso em: out. 2006.

CURBERA, F. et al. Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. **IEEE Internet Computing**, [S.l.], v. 6, n. 2, p. 86-93, Mar./Apr. 2002.

DMTF: Distributed Management Task Force. Disponível em: < http://www.dmtf.org/home>. Acesso em: set. 2007.

DURHAM, E. et al. **The COPS (Common Open Policy Service) Protocol**: RFC 2748. [S.l.]: Internet Engineering Task Force, Network Working Group, 2000.

FESTOR, O. et al. Management of Mobile Ad-hoc Networks: Evaluating the Network Behavior. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM IN INTEGRATED NETWORK MANAGEMENT, IM, 9., 2005. **Proceedings…** [S.l.:s.n.], 2005a.

FESTOR, O. et al. A hierarchical architecture for a distributed management of P2P networks and services. In: IFIP/IEEE DISTRIBUTED SYSTEMS: OPERATION AND MANAGEMENT, DSOM, 16., 2005. **Proceedings…** Spain: [s.n.], 2005b.

FIELDING, R. et al. **Hypertext Transfer Protocol – HTTP/1.1**: RFC 2616. [S.l.]: Internet Engineering Task Force, Network Working Group, 1999.

GINEVAN, S. Can WiMax go the distance? **Planet Analog**, [S.l.], Mar. 2008. Disponível em: < http://www.planetanalog.com/news/showArticle.jhtml?articleID=206904742&pgno=2> . Acesso em: mar. 2008.

GOLDSZMIDT, G.; YEMINI, Y. Distributed Management by Delegation. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS, ICDCS, 15., 1995. **Proceedings…** New York: [s.n.], 1995.

GOSH, A. et al. Broadband Wireless Access with WiMax/802.16: Current Performance Benchmarks and Future Potential. **IEEE Communications Magazine**, [S.l.], v. 43, n. 2, p. 129-136, Feb. 2005.

GRANVILLE, L. **Gerenciamento Integrado de QoS em Redes de Computadores**. 2001. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.

GRANVILLE, L. et al. An Approach for Integrated management of Networks with Quality of Service Support Using QAME. In: INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS AND MANAGEMENT, DSOM, 12., 2001. **Proceedings...** Nancy, France: [s.n.], 2001a.

GRANVILLE, L. et al. QAME – QoS-aware management environment. In: COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, COMPSAC, 25., 2001. **Proceedings...** [S.l.:s.n.], 2001b.

GRANVILLE, L. et al. An architecture for Automated Replacement of QoS Policies. In: INTERNATIONAL SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS, ISCC, 7., 2002. **Proceedings…** [S.l.:s.n.], 2002

GRANVILLE, L. et al. Managing computer networks using peer-to-peer technologies. **IEEE Communications Magazine**, [S.l.], v. 43, n. 10, p. 62-68, Oct. 2005.

GUDGIN, M. et al. **SOAP Version 1.2 Part 1**: Messaging Framework (Second Edition): W3C Recommendation. [S.l.:s.n.], 2007. Disponível em: <http://www.w3.org/TR/soap12-part1>. Acesso em: set. 2007.

HERZOG, E. et al. **COPS usage for RSVP**: RFC 2749. [S.l.]: Internet Engineering Task Force, Network Working Group, 2000.

HONG, S.; KWON, O. Considerations for VoIP Services in IEEE 802.16 Broadband Wireless Access Systems. In: IEEE VEHICULAR TECHNOLOGY CONFERENCE, VTC, 63., 2006. **Proceedings…** [S.l.:s.n.], 2006.

HWANG, Y.; KIM, E. An architecture of SNMP-based network management of the broadband wireless access system. In: ASIA-PACIFIC CONFERENCE ON COMMUNICATIONS, APCC, 9., 2003. **Proceedings…** [S.l.:s.n.], 2003.

INSTITUTE OF ELETRICAL AND ELETRONIC ENGINEERING. **IEEE 802.16-2001**: IEEE Standard for Local and metropolitan area networks – Part 16: Air Interface for Fixed Bradband Wireless Access Systems. New York, 2002.

INSTITUTE OF ELETRICAL AND ELETRONIC ENGINEERING. **IEEE 802.16d-2004**: IEEE Standard for Local and metropolitan area networks – Part 16: Air Interface for Fixed Bradband Wireless Access Systems. New York, 2004.

INSTITUTE OF ELETRICAL AND ELETRONIC ENGINEERING. **IEEE 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. New York, 2005a.

INSTITUTE OF ELETRICAL AND ELETRONIC ENGINEERING. **IEEE 802.16f-2005**: IEEE Standard for Local and metropolitan area networks – Part 16: Air Interface for Fixed Bradband Wireless Access Systems – Amendment 1: Management Information Base. New York, 2005b.

INSTITUTE OF ELETRICAL AND ELETRONIC ENGINEERING. **IEEE 802.16e-2005**: IEEE Standard for Local and metropolitan area networks – Part 16: Air Interface for Fixed Bradband Wireless Access Systems – Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands. New York, 2006.

JONES, D. WiMAX: A Spec Divided. **Light Reading**, [S.l.], July 2005. Disponível em: <http://www.lightreading.com/document.asp?doc_id=76864&WT.svl=news1_2>. Acesso em: mar. 2008.

LABARRE, L. **Management By Exception**: OSI Event Generation, Reporting, and Logging. Burlington Road: The MITRE Corporation, 1991.

LEE, D.; KYAMAKYA, K.; UMONDI, J. Fast Handover Algorithm for IEEE 802.16e Broadband Wireless Access Systems. In: INTERNATIONAL SYMPOSIUM ON WIRELESS PERVASIVE COMPUTING, 1., 2006. **Proceedings…** [S.l.:s.n.], 2006.

LEINWAND, A. **Network Management: A Practical Perspective**. 2nd ed. Reading, MA: Addison Wesley, 1996.

LIU, N. et al. Delay Character of a Novel Architecture for IEEE 802.16 Systems. In: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED COMPUTING, APPLICATIONS AND TECHNOLOGIES, PDCAT, 6., 2005. **Proceedings…** [S.l.:s.n.], 2005.

LUPU, E.; SLOMAN, M. Conflicts in Policy-Based Distributed Systems Management. **IEEE Transactions on Software Engineering**, [S.l.], v. 25, n. 6, p. 852-869, Nov./Dec. 1999.

LYMAN, J. WiMAX Mobile Standard Ratified. **TechNewsWorld**, [S.l.], Sept. 2005. Disponível em: <http://www.technewsworld.com/story/47766.html>. Acesso em: mar. 2008.

MARQUEZAN, C. et al. QAME Support for Policy-Based Management of Country-wide Networks. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, IM, 9., 2005. **Application session proceedings...** Nice, France: [s.n.], 2005.

MEER, H. et al. Programmable Agents for Flexible QoS Management in IP Networks. **IEEE Journal on Selected Areas in Communications**, [S.l.], v. 18, n. 2, p. 256-267, Feb. 2000.

MOFFETT, J.; SLOMAN, M. Policy Conflict Analysis in Distributed System Management. **Journal of Organizational Computing**, USA, v. 4, n. 1, 1994.

MOORE, B. et al. **Policy Core Information Model – Version 1 Specification**: RFC 3060. [S.l.]: Internet Engineering Task Force, Network Working Group, 2001.

MOORE, B. **Policy Core Information Model (PCIM) Extensions**: RFC 3460. [S.l.]: Internet Engineering Task Force, Network Working Group, 2003.

NET-SNMP. Disponível em: <http://net-snmp.sourceforge.net/>. Acesso em: set. 2007.

PANA, M. et al. **Policy Core Extension Lightweight Directory Access Protocol Schema (PCELS)**: RFC 4104. [S.l.]: Internet Engineering task Force, Network Working Group, 2005.

PONNAPPAN, A. et al. A Policy Based QoS Management Systemfor the IntServ/DiffServ Based Internet. In: INTERNATIONAL WORKSHOP ON POLICIES FOR DISTRIBUTED SYSTEMS AND NETWORKS, POLICY, 3., 2002. **Proceedings…** [S.l.:s.n.], 2002.

PRADO, E.; LIMA, F. Dimensionamento de Redes WiMAX. **Revista de WiMAX**, [S.l.], maio 2006. Disponível em: <http://www.revistadewimax.com.br/AdminSite/Revista/DimensionamentoRedes/tabid/92/Default.aspx>. Acesso em: set. 2006.

PRADO, E. Os desafios do WiMAX móvel. **Revista de WiMAX**, [S.l.], fev. 2007. Disponível em: <http://www.revistadewimax.com.br/Revista/WiMAXMóvel/tabid/89/Default.aspx>. Acesso em: mar. 2008.

SAMAAN, N.; KARMOUCH, A. Prediction-Based Policy Adaptation for QoS Management in Wireless Networks. In: INTERNATIONAL WORKSHOP ON POLICIES FOR DISTRIBUTED SYSTEMS AND NETWORKS, POLICY, 4., 2003. **Proceedings…** [S.l.:s.n.], 2003.

STONE, G.; LUNDY, B.; XIE, G. Network Policy Languages: A Survey and a New Approach. **IEEE Network**, [S.l.], v. 15, n. 1, p. 10-21, Jan. 2001.

STRASSNER, J. **Policy-Based Network Management**: Solutions for the Next Generation. [S.l.]: Morgan Kaufmann, 2003.

WANG, H.; HE, B.; AGRAWAL, D. Admission Control and Bandwidth Allocation above Packet Level for IEEE 802.16 Wireless MAN. In: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED SYSTEMS, ICPADS, 12., 2006. **Proceedings…** [S.l.:s.n.], 2006.

WANG, H.; LI, W.; AGRAWAL, D. Dynamic Admission Control and QoS for 802.16 Wireless MAN. In: WIRELESS TELECOMMUNICATIONS SYMPOSIUM, 2005. **Proceedings…** [S.l.:s.n.], 2005.

WEI, H. et al. Interference-Aware IEEE 802.16 WiMax Mesh Networks. In: IEEE VEHICULAR TECHNOLOGY CONFERENCE, VTC, 61., 2005, Stockholm, SW. **Proceedings…** [S.l.]: IEEE, 2005.

WIKIPEDIA. **IEEE 802.16**. Disponível em: <http://en.wikipedia.org/wiki/IEEE_802.16>. Acesso em: set. 2006.

WIMAX FORUM. Disponível em: <http://www.wimaxforum.org>. Acesso em: set. 2006.

WONGTHAVARAWAT, K.; GANZ, A. IEEE 802.16 Based Last Mile Broadband Wireless Military Networks with Quality of Service Support. In: IEEE MILITARY COMMUNICATIONS CONFERENCE, MILCOM, 2003. **Proceedings…** [S.l.:s.n.], 2003.

YAVATKAR, R.; PENDARAKIS, D.; GUERIN, R. **A Framework for Policy-based Admission Control**: RFC 2753. [S.l.]: Internet Engineering Task Force, Network Working Group, 2000.

ZEILENGA, K. **Lightweight Directory Access Protocol (LDAP)**: Technical Specification Road Map: RFC 4510. [S.l.]: Internet Engineering Task Force, Network Working Group, 2006.

APÊNDICE A MODIFICAÇÕES NA MIB PARA AJUSTE NOS IDENTIFICADORES DE FLUXOS

(. . .)

```
wmanIfBsProvisionedSfTable OBJECT-TYPE
SYNTAX SEQUENCE OF WmanIfBsProvisionedSfEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table contains service flow profiles provisioned by
NMS. The service flow should be created with SS(s)
following instruction given by wmanIfBsSfState object.
1. The QoS parameters of the service flow are provisioned
in wmanIfBsServiceClassTable and referenced by
wmanIfBsServiceClassIndex.
2. The classifier rules of the service flow are provisioned
in wmanIfBsClassifierRuleTable, where they refer to SF
via wmanIfBsSfMacAddress and wmanIfBsSfId.

The wmanIfBsSfMacAddress field must be the SS MAC address in
case of unicast flows and the broadcast MAC address
(ff:ff:ff:ff:ff:ff) in case of multicast flows.
The MAC addresses of SSs the service flow is created with
are provisioned in wmanIfBsSsProvisionedForSfTable, where
they refer to SF via wmanIfBsSfId.
This table represents only the desire of the management system.
The table wmanIfCmnCpsServiceFlowTable is what actually
represents the current network state"
REFERENCE
"Subclause 6.3.13 and 6.3.14 in IEEE Std 802.16-2004"
::= { wmanIfBsPacketCs 1 }

wmanIfBsProvisionedSfEntry OBJECT-TYPE
SYNTAX WmanIfBsProvisionedSfEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table provides one row for each service flow
provisioned by NMS. The table is indexed by ifIndex,
wmanIfBsSfMacAddress and wmanIfBsSfId.
ifIndex is associated with the BS sector."
INDEX { ifIndex, wmanIfBsSfMacAddress, wmanIfBsSfId }
::= { wmanIfBsProvisionedSfTable 1 }
```

WmanIfBsProvisionedSfEntry ::= SEQUENCE {
wmanIfBsSfMacAddress MacAddress,
wmanIfBsSfId Unsigned32,
wmanIfBsSfDirection INTEGER,
wmanIfBsServiceClassIndex INTEGER,
wmanIfBsSfState WmanIfSfState,
wmanIfBsSfProvisionedTime TimeStamp,
wmanIfBsSfCsSpecification WmanIfCsSpecification,
wmanIfBsProvisionedSfRowStatus RowStatus}

wmanIfBsSfMacAddress OBJECT-TYPE
SYNTAX **MacAddress**
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The MAC address of the SS the service flow is created with or the broadcast MAC address (ff:ff:ff:ff:ff:ff) in case of multicast flows."
::= { wmanIfBsProvisionedSfEntry 1 }

(. . .)

wmanIfBsClassifierRuleEntry OBJECT-TYPE
SYNTAX **WmanIfBsClassifierRuleEntry**
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table provides one row for each packet classifier rule, and is indexed by ifIndex, wmanIfBsSfMacAddress, wmanIfBsSfId, and wmanIfBsClassifierRuleIndex.
IfIdex is associated with the BS sector.
wmanIfBsSfMacAddress and wmanIfBsSfId identify the service flow, while wmanIfBsClassifierRuleIndex identifies the packet classifier rule."
INDEX { ifIndex, wmanIfBsSfMacAddress, wmanIfBsSfId, wmanIfBsClassifierRuleIndex }
::= { wmanIfBsClassifierRuleTable 1 }

(. . .)

wmanIfBsPkmTekTable OBJECT-TYPE
SYNTAX **SEQUENCE OF WmanIfBsPkmTekEntry**
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table describes the attributes of each Traffic Encryption Key (TEK) association. The BS maintains one TEK association per SAID/SS on each BS wireless interface."
::= { wmanIfBsPkmObjects 3 }
wmanIfBsPkmTekEntry OBJECT-TYPE
SYNTAX **WmanIfBsPkmTekEntry**
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Each entry contains objects describing attributes of one TEK association on a particular BS wireless interface. The BS MUST create one entry per SAID/SS per wireless interface,

```
based on the receipt of a Key Request message, and MUST not
delete the entry before the SS authorization for the SAID
permanently expires."
INDEX { ifIndex, wmanIfBsPkmTekMacAddress, wmanIfBsPkmTekSAId }
::= { wmanIfBsPkmTekTable 1 }

WmanIfBsPkmTekEntry ::= SEQUENCE {
wmanIfBsPkmTekMacAddress MacAddress,
wmanIfBsPkmTekSAId INTEGER,
wmanIfBsPkmTekSAType INTEGER,
wmanIfBsPkmTekDataEncryptAlg WmanIfDataEncryptAlgId,
wmanIfBsPkmTekDataAuthentAlg WmanIfDataAuthAlgId,
wmanIfBsPkmTekEncryptAlg WmanIfTekEncryptAlgId,
wmanIfBsPkmTekLifetime Integer32,
wmanIfBsPkmTekKeySequenceNumber Integer32,
wmanIfBsPkmTekExpiresOld DateAndTime,
wmanIfBsPkmTekExpiresNew DateAndTime,
wmanIfBsPkmTekReset TruthValue,
wmanIfBsPkmKeyRequests Counter32,
wmanIfBsPkmKeyReplies Counter32,
wmanIfBsPkmKeyRejects Counter32,
wmanIfBsPkmTekInvalids Counter32,
wmanIfBsPkmKeyRejectErrorCode INTEGER,
wmanIfBsPkmKeyRejectErrorString SnmpAdminString,
wmanIfBsPkmTekInvalidErrorCode INTEGER,
wmanIfBsPkmTekInvalidErrorString SnmpAdminString}

wmanIfBsPkmTekMacAddress OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The MAC address of the SS the security association
is created with. If the security association is used
with all subscriber stations of a multicast service
flow, there will be one entry for each subscriber
station."
::= { wmanIfBsPkmTekEntry 1 }
```

(. . .)

```
wmanIfCmnClassifierRuleEntry OBJECT-TYPE
SYNTAX WmanIfCmnClassifierRuleEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table provides one row for each packet classifier
rule, and is indexed by ifIndex, wmanIfCmnCpsSfMacAddress,
wmanIfCmnCpsSfId, and wmanIfCmnClassifierRuleIndex.
ifIndex is associated with the BS sector.
wmanIfCmnCpsSfMacAddress and wmanIfCmnCpsSfId identify the
service flow, and wmanIfCmnClassifierRuleIndex identifies the
packet classifier rule."
INDEX { ifIndex, wmanIfCmnCpsSfMacAddress, wmanIfCmnCpsSfId,
wmanIfCmnClassifierRuleIndex }
::= { wmanIfCmnClassifierRuleTable 1 }
```

(. . .)

wmanIfCmnPhsRuleEntry OBJECT-TYPE
SYNTAX **WmanIfCmnPhsRuleEntry**
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table provides one row for each PHS rule created dynamically by the BS and SS on a given service flow. The PHS rule is defined by the pair (PHSS, PHSM) for each distinct header data. It is indexed by IfIndex, wmanIfCmnCpsSfMacAddress, wmanIfCmnCpsSfId, and wmanIfCmnPhsIndex. The table is read-only for NMS. "
INDEX { ifIndex, wmanIfCmnCpsSfMacAddress, wmanIfCmnCpsSfId, wmanIfCmnPhsRulePhsIndex }
::= { wmanIfCmnPhsRuleTable 1 }

(. . .)

APÊNDICE B MODIFICAÇÕES NA MIB PARA MAIOR CONTROLE SOBRE SERVICE FLOWS

(. . .)

```
WmanIfBsConfigurationEntry ::= SEQUENCE {
wmanIfBsDcdInterval INTEGER,
wmanIfBsUcdInterval INTEGER,
wmanIfBsUcdTransition INTEGER,
wmanIfBsDcdTransition INTEGER,
wmanIfBsInitialRangingInterval INTEGER,
wmanIfBsSsULMapProcTime Unsigned32,
wmanIfBsSsRangRespProcTime Unsigned32,
wmanIfBsT5Timeout INTEGER,
wmanIfBsT9Timeout INTEGER,
wmanIfBsT13Timeout INTEGER,
wmanIfBsT15Timeout INTEGER,
wmanIfBsT17Timeout INTEGER,
wmanIfBsT27IdleTimer Unsigned32,
wmanIfBsT27ActiveTimer Unsigned32,
wmanIfBs2ndMgmtDlQoSProfileIndex INTEGER,
wmanIfBs2ndMgmtUlQoSProfileIndex INTEGER,
wmanIfBsSsSfMgmtEnabled INTEGER,
wmanIfBsAutoSfidEnabled INTEGER,
wmanIfBsAutoSfidRangeMin Unsigned32,
wmanIfBsAutoSfidRangeMax Unsigned32,
wmanIfBsMinBestEffortResources INTEGER,
wmanIfBsAasChanFbckReqFreq INTEGER,
wmanIfBsAasBeamSelectFreq INTEGER,
wmanIfBsAasChanFbckReqResolution INTEGER,
wmanIfBsAasBeamReqResolution INTEGER,
wmanIfBsAasNumOptDiversityZones INTEGER,
wmanIfBsResetSector INTEGER}

(...)

wmanIfBsSsSfMgmtEnabled OBJECT-TYPE
SYNTAX INTEGER {ssSfMgmtDisabled(0),
ssSfMgmtEnabled(1)}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This object defines weather SSs are allowed to
manage service flows related to it dynamically through
DSA, DSC or DSD messages.
```

When set to ssSfMgmtDisabled, the BS must automatically reject any DSA, DSC or DSD messages coming from SSs."
REFERENCE
"Subclause 6.3.2.3 in IEEE Std 802.16-2004"
::= { wmanIfBsConfigurationEntry 17 }

(...)

wmanIfBsMinBestEffortResources OBJECT-TYPE
SYNTAX **INTEGER (0 .. 10000)**
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"This object defines the minimum percentage of the network resources to be used with Best Effort service flows.
The objective of this parameter is to avoid service starvation caused by the admission of an excessive number of non Best Effort service flows.
This value is expressed with two digits fixed point in a range from 0.00% to 100.00%"
REFERENCE
"Subclause 11.13.1 in IEEE Std 802.16-2004"
::= { wmanIfBsConfigurationEntry 21 }

(. . .)

WmanIfCmnCpsServiceFlowEntry::= SEQUENCE {
wmanIfCmnCpsSfMacAddress MacAddress,
wmanIfCmnCpsSfId Unsigned32,
wmanIfCmnCpsSfProvisionStatus INTEGER,
wmanIfCmnCpsSfCid WmanIfCidType,
wmanIfCmnCpsSfType INTEGER,
wmanIfCmnCpsSfDirection INTEGER,
wmanIfCmnCpsSfState WmanIfSfState,
wmanIfCmnCpsTrafficPriority INTEGER,
wmanIfCmnCpsMaxSustainedRate Unsigned32,
wmanIfCmnCpsMaxTrafficBurst Unsigned32,
wmanIfCmnCpsMinReservedRate Unsigned32,
wmanIfCmnCpsToleratedJitter Unsigned32,
wmanIfCmnCpsMaxLatency Unsigned32,
wmanIfCmnCpsFixedVsVariableSduInd INTEGER,
wmanIfCmnCpsSduSize Unsigned32,
wmanIfCmnCpsSfSchedulingType WmanIfSfSchedulingType,
wmanIfCmnCpsArqEnable TruthValue,
wmanIfCmnCpsArqWindowSize INTEGER,
wmanIfCmnCpsArqBlockLifetime INTEGER,
wmanIfCmnCpsArqSyncLossTimeout INTEGER,
wmanIfCmnCpsArqDeliverInOrder TruthValue,
wmanIfCmnCpsArqRxPurgeTimeout INTEGER,
wmanIfCmnCpsArqBlockSize INTEGER,
wmanIfCmnCpsMinRsvdTolerableRate Unsigned32,
wmanIfCmnCpsReqTxPolicy BITS,
wmanIfCmnSfCsSpecification WmanIfCsSpecification,
wmanIfCmnCpsTargetSaid INTEGER}

(...)

wmanIfCmnCpsSfProvisionStatus OBJECT-TYPE

SYNTAX **INTEGER {provisioned(1),**
automaticallyAdded(2),
destroy(3)}
MAX-ACCESS read-write
STATUS current
DESCRIPTION "Indicates weather the service flow has been
provisioned or automatically added.
In a BS, setting an automatically added flow to provisioned will
create the corresponding objects bellow wmanIfBsPacketCs as if
the flow has been provisioned. A consequence of that is that the
service flow will become manageable.
For automatically added service flows, setting this field to destroy
will cause the destruction of the service flow without the need to
make it manageable.
Setting this field in other circunstances not described above will
have no effect."
::= { wmanIfCmnCpsServiceFlowEntry 4 }

wmanIfCmnCpsSfType OBJECT-TYPE
SYNTAX **INTEGER {unicast(1),**
multicast(2)}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The type of a service flow with respect to
multiplexing among different subscriber stations"
::= { wmanIfCmnCpsServiceFlowEntry 5 }

(. . .)

APÊNDICE C MODIFICAÇÕES NA MIB PARA MAIOR CONTROLE SOBRE AS ESTAÇÕES CLIENTE

(. . .)

```
wmanIfBsConnectedSsTable OBJECT-TYPE
SYNTAX SEQUENCE OF WmanIfBsConnectedSsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table contains basic information of SSs that have
at least one connection to this BS. An entry in this table
indicates the SS has at least its basic and primary
management connection identifiers assigned. The assignment
occurs during the ranging process."
REFERENCE
"Subclause 6.3.2.3.5 in IEEE Std 802.16-2004"
::= { wmanIfBsCps 1 }

wmanIfBsConnectedSsEntry OBJECT-TYPE
SYNTAX WmanIfBsConnectedSsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table provides one row for each SS that is connected
to the BS, and is indexed by ifIndex and
wmanIfBsConnectedSsMacAddress.
The primary index is the ifIndex with an ifType of
propBWAp2Mp, indicating the BS sector with which the SS
is associated. wmanIfBsConnectedSsMacAddress identifies the
connected SS."
INDEX { ifIndex, wmanIfBsConnectedSsMacAddress }
::= { wmanIfBsConnectedSsTable 1 }

WmanIfBsConnectedSsEntry ::= SEQUENCE {
wmanIfBsConnectedSsMacAddress MacAddress,
wmanIfBsConnectedSsBasicCid WmanIfCidType,
wmanIfBsConnectedSsPrimaryCid WmanIfCidType,
wmanIfBsConnectedSsAasBroadcastPermission INTEGER,
wmanIfBsConnectedSsMacVersion WmanIfMacVersion}

wmanIfBsConnectedSsMacAddress OBJECT-TYPE
SYNTAX MacAddress
```

```
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The MAC address of SS is received from the RNG-REQ
message."
REFERENCE
"Subclause 6.3.2.3.5 in IEEE Std 802.16-2004"
::= { wmanIfBsConnectedSsEntry 1 }

wmanIfBsConnectedSsBasicCid OBJECT-TYPE
SYNTAX WmanIfCidType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object indicates the SS's basic CID
that was sent in the RNG-RSP message."
REFERENCE
"Subclause 6.3.2.3.6 in IEEE Std 802.16-2004"
::= { wmanIfBsConnectedSsEntry 2 }

wmanIfBsConnectedSsPrimaryCid OBJECT-TYPE
SYNTAX WmanIfCidType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object indicates the primary CID of the
SS received from the RNG-RSP message."
REFERENCE
"Subclause 6.3.2.3.6 in IEEE Std 802.16-2004"
::= { wmanIfBsConnectedSsEntry 3 }

wmanIfBsConnectedSsAasBroadcastPermission OBJECT-TYPE
SYNTAX INTEGER {contBasedBwReqPermitted(0),
contBasedBwReqNotPermitted(1)}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This parameter specifies if SS can issue contention-based
bandwidth request or not."
REFERENCE
"Subclause 11.6 in IEEE Std 802.16-2004"
::= { wmanIfBsConnectedSsEntry 4 }

wmanIfBsConnectedSsMacVersion OBJECT-TYPE
SYNTAX WmanIfMacVersion
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This parameter specifies the version of 802.16 to which the
message originator conforms."
REFERENCE
"Subclause 11.1.3 in IEEE Std 802.16-2004"
::= { wmanIfBsConnectedSsEntry 5 }
```

(. . .)

```
WmanIfBsRegisteredSsEntry ::= SEQUENCE {
wmanIfBsSsMacAddress MacAddress,
```

```
wmanIfBsSsBasicCid WmanIfCidType,
wmanIfBsSsPrimaryCid WmanIfCidType,
wmanIfBsSsSecondaryCid WmanIfCidType,
wmanIfBsSsManagementSupport INTEGER,
wmanIfBsSsIpManagementMode INTEGER,
wmanIfBsSs2ndMgmtArqEnable TruthValue,
wmanIfBsSs2ndMgmtArqWindowSize INTEGER,
wmanIfBsSs2ndMgmtArqDnLinkTxDelay INTEGER,
wmanIfBsSs2ndMgmtArqUpLinkTxDelay INTEGER,
wmanIfBsSs2ndMgmtArqDnLinkRxDelay INTEGER,
wmanIfBsSs2ndMgmtArqUpLinkRxDelay INTEGER,
wmanIfBsSs2ndMgmtArqBlockLifetime INTEGER,
wmanIfBsSs2ndMgmtArqSyncLossTimeout INTEGER,
wmanIfBsSs2ndMgmtArqDeliverInOrder TruthValue,
wmanIfBsSs2ndMgmtArqRxPurgeTimeout INTEGER,
wmanIfBsSs2ndMgmtArqBlockSize INTEGER,
wmanIfBsSsVendorIdEncoding OCTET STRING,
wmanIfBsSsAasBroadcastPermission INTEGER,
wmanIfBsSsMaxTxPowerBpsk WmanIfMaxTxPowerType,
wmanIfBsSsMaxTxPowerQpsk WmanIfMaxTxPowerType,
wmanIfBsSsMaxTxPower16Qam WmanIfMaxTxPowerType,
wmanIfBsSsMaxTxPower64Qam WmanIfMaxTxPowerType,
wmanIfBsSsMacVersion WmanIfMacVersion,
wmanIfBsSsMaxUplinkTransmission Gauge,
wmanIfBsSsMaxDownlinkTransmission Gauge}

(...)

wmanIfBsSsMaxUplinkTransmission OBJECT-TYPE
SYNTAX Gauge
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This parameter specifies the current maximum transmission rate
of the SS in kbit/s. It may be changed toghether with the SS burst
profile."
::= { wmanIfBsRegisteredSsEntry 25 }

wmanIfBsSsMaxDownlinkTransmission OBJECT-TYPE
SYNTAX Gauge
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This parameter specifies the current maximum reception rate
of the SS in kbit/s. It may be changed toghether with the SS burst
profile."
::= { wmanIfBsRegisteredSsEntry 26 }
```

(...)

APÊNDICE D MODIFICAÇÕES NA MIB PARA NOVAS NOTIFICAÇÕES

(. . .)

```
wmanIfBsTrapControlRegister OBJECT-TYPE
SYNTAX BITS {wmanIfBsSsStatusNotification (0),
wmanIfBsSsDynamicServiceSuccess (1),
wmanIfBsSsDynamicServiceFail (2),
wmanIfBsSsServiceAdmissionSuccess (3),
wmanIfBsSsServiceAdmissionFail (4),
wmanIfBsSsRssiStatusChange (5),
wmanIfBsSsConnect (6),
wmanIfBsSsRegister (7),
wmanIfBsSsHandOver (8),
wmanIfBsSsPkmFail (9),
wmanIfBsSsBurstProfileChange (10),
wmanIfBsSsServiceDegradation (11),
wmanIfBsNetDegradation (12)}
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"The object is used to enable or disable Base Station traps.
From left to right, the set bit indicates the corresponding
Base Station trap is enabled."
::= { wmanIfBsTrapControl 1 }
```

(. . .)

```
WmanIfBsThresholdConfigEntry ::= SEQUENCE {
wmanIfBsRssiLowThreshold Integer32,
wmanIfBsRssiHighThreshold Integer32,
wmanIfBsServiceDegradationStartThreshold Unsigned32,
wmanIfBsServiceDegradationStepThreshold Unsigned32,
wmanIfBsServiceDegradationHysteresisTime Unsigned32,
wmanIfBsNetDegradationStartThreshold Unsigned32,
wmanIfBsNetDegradationStepThreshold Unsigned32,
wmanIfBsNetDegradationHysteresisTime Unsigned32}

(...)

wmanIfBsServiceDegradationStartThreshold OBJECT-TYPE
SYNTAX Unsigned32 (1 .. 1000000)
MAX-ACCESS read-write
```

```
STATUS current
DESCRIPTION
"Threshold for wmanIfBsServiceDegradationUsefulPercentage that
specifies when to first notify about a service flow degradation.
If wmanIfBsServiceDegradationUsefulPercentage for a service flow
is greater than or equal to this threshold, the service flow is
not considered as being suffering from degradation.
This value is expressed with four digits fixed point in a range
from 0.0001% to 100.0000%"
::= { wmanIfBsThresholdConfigEntry 3 }

wmanIfBsServiceDegradationStepThreshold OBJECT-TYPE
SYNTAX Unsigned32 (1 .. 1000000)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Step in wmanIfBsServiceDegradationUsefulPercentage
considered to notify about a new service flow degradation
for a service flow already suffering from service degradation.
This value is expressed with four digits fixed point in a range
from 0.0001% to 100.0000%"
::= { wmanIfBsThresholdConfigEntry 4 }

wmanIfBsServiceDegradationHysteresisTime OBJECT-TYPE
SYNTAX Unsigned32
UNITS "seconds"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Number of seconds a service flow degradation situation must
stay solved before an event reporting the end of degradation can
be sent"
::= { wmanIfBsThresholdConfigEntry 5 }

wmanIfBsNetDegradationStartThreshold OBJECT-TYPE
SYNTAX Unsigned32 (1 .. 1000000)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Threshold for wmanIfBsNetDegradationUsefulPercentage that
specifies when to first notify about a network degradation.
If wmanIfBsNetDegradationUsefulPercentage is greater than
or equal to this threshold, the network is not considered as being
suffering from degradation.
This value is expressed with four digits fixed point in a range
from 0.0001% to 100.0000%"
::= { wmanIfBsThresholdConfigEntry 6 }

wmanIfBsNetDegradationStepThreshold OBJECT-TYPE
SYNTAX Unsigned32 (1 .. 1000000)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Step in wmanIfBsNetDegradationUsefulPercentage considered
to notify about a new network degradation when the management
system already knows the network is suffering from degradation.
This value is expressed with four digits fixed point in a range
from 0.0001% to 100.0000%"
::= { wmanIfBsThresholdConfigEntry 7 }
```

wmanIfBsNetDegradationHysteresisTime OBJECT-TYPE
SYNTAX **Unsigned32**
UNITS "seconds"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"Number of seconds a network degradation situation must stay solved before an event reporting the end of degradation can be sent"
::= { wmanIfBsThresholdConfigEntry 8 }

(. . .)

WmanIfBsSsNotificationObjectsEntry ::= SEQUENCE {
wmanIfBsSsNotificationMacAddr MacAddress,
wmanIfBsSsStatusValue INTEGER,
wmanIfBsSsStatusInfo OCTET STRING,
wmanIfBsDynamicServiceSfId Unsigned32,
wmanIfBsDynamicServiceType INTEGER,
wmanIfBsDynamicServiceFailReason OCTET STRING,
wmanIfBsServiceAdmissionSfId Unsigned32,
wmanIfBsServiceAdmissionStatus INTEGER,
wmanIfBsServiceAdmissionFailReason OCTET STRING,
wmanIfBsSsRssiStatus INTEGER,
wmanIfBsSsRssiStatusInfo OCTET STRING,
wmanIfBsSsConnectStatus INTEGER,
wmanIfBsSsRegisterStatus INTEGER,
wmanIfBsSsHandoverStatus INTEGER,
wmanIfBsServiceDegradationSfId Unsigned32,
wmanIfBsServiceDegradationStatus INTEGER,
wmanIfBsServiceDegradationUsefulPercentage Unsigned32}

(...)

wmanIfBsDynamicServiceSfId OBJECT-TYPE
SYNTAX **Unsigned32 (1 .. 4294967295)**
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"A 32 bit quantity that uniquely identifies the last created dynamic service flow to both the subscriber station and base station (BS)."
::= { wmanIfBsSsNotificationObjectsEntry 4 }

(...)

wmanIfBsServiceAdmissionSfId OBJECT-TYPE
SYNTAX **Unsigned32 (1 .. 4294967295)**
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"A 32 bit quantity that uniquely identifies the last provisioned service flow submitted to the admission control engine to both the subscriber station and base station (BS)."
::= { wmanIfBsSsNotificationObjectsEntry 7 }

wmanIfBsServiceAdmissionStatus OBJECT-TYPE

SYNTAX **INTEGER {bsSfAdmissionSuccess(1),**
bsSfAdmissionDegraded(2),
bsSfAdmissionFail(3)}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates the admission status of the last provisioned service flow submitted to the admission control engine.

bsSfAdmissionSuccess indicates the service flow has been admitted in the network

bsSfAdmissionDegraded indicates the service flow has been admitted in a
degraded mode - some of its requirements couldn't be completelly filled

bsSfAdmissionFail indicates the service flow has not been admitted in the network"
::= { wmanIfBsSsNotificationObjectsEntry 8 }

wmanIfBsServiceAdmissionFailReason OBJECT-TYPE
SYNTAX **OCTET STRING (SIZE(0..255))**
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates the reason why the last provisioned service flow submitted to the admission control engine hasn't been admited."
::= { wmanIfBsSsNotificationObjectsEntry 9 }

(...)

wmanIfBsSsConnectStatus OBJECT-TYPE
SYNTAX **INTEGER {ssConnect(1),**
ssDisconnect(2)}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates the status of SS connection."
::= { wmanIfBsSsNotificationObjectsEntry 12 }

(...)

wmanIfBsSsHandoverStatus OBJECT-TYPE
SYNTAX **INTEGER {ssHandoverAttempt(1),**
ssHandoverCancel(2)}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates the status of SS handover.

ssHandoverAttempt indicates the BS has received a
MOD_HO-IND message with mode 0b00 (HO) and
HO_IND_type 0b00 (serving BS release)

ssHandOverCancel indicates the BS has received a
MOD_HO-IND message with mode 0b00 (HO) and
HO_IND_type 0b01 (HO cancel)"
::= { wmanIfBsSsNotificationObjectsEntry 14 }

wmanIfBsServiceDegradationSfId OBJECT-TYPE
SYNTAX **Unsigned32 (1 .. 4294967295)**
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"A 32 bit quantity that uniquely identifies the last degraded
or restablished service flow to both the subscriber station
and base station (BS)."
::= { wmanIfBsSsNotificationObjectsEntry 15 }

wmanIfBsServiceDegradationStatus OBJECT-TYPE
SYNTAX **INTEGER {bsSfDegradationStarted(1),**
bsSfDegradationEnded(2)}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates the last degradation status change of
the last degraded or restablished service flow."
::= { wmanIfBsSsNotificationObjectsEntry 16 }

wmanIfBsServiceDegradationUsefulPercentage OBJECT-TYPE
SYNTAX **Unsigned32 (1 .. 1000000)**
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The percentage of the reserved network utilization that can
still be provided by the network during the service flow
degradation period.
It can be mathematically expressed as the ratio between the time
the network is able to use to assure bandwidth reservation
constraints and the time the network would need to be using to
assure bandwidth reservation constraints without any degradation.
The complement of this value can be viewed as the percentage of
service flow degradation.
This value is expressed with four digits fixed point in a range
from 0.0001% to 100.0000%
The idea is not to have an exact value, but to give a hint to
the management system about the criticity of the degradation"
::= { wmanIfBsSsNotificationObjectsEntry 17 }

(. . .)

wmanIfBsNetNotificationObjectsTable OBJECT-TYPE
SYNTAX **SEQUENCE OF WmanIfBsNetNotificationObjectsEntry**
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table contains notification objects that have been
reported by the trap and affect the BS sector as a whole."
::= { wmanIfBsTrapDefinitions 2 }

wmanIfBsNetNotificationObjectsEntry OBJECT-TYPE
SYNTAX **WmanIfBsNetNotificationObjectsEntry**
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table provides one row for each BS sector that has

```
generated traps, and is indexed by ifIndex for BS sector."
INDEX { ifIndex }
::= { wmanIfBsNetNotificationObjectsTable 1 }

WmanIfBsNetNotificationObjectsEntry ::= SEQUENCE {
wmanIfBsNetDegradationStatus INTEGER,
wmanIfBsNetDegradationUsefulPercentage Unsigned32}

wmanIfBsNetDegradationStatus OBJECT-TYPE
SYNTAX INTEGER {bsNetDegradationStarted(1),
bsNetDegradationEnded(2)}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object indicates the status of a BS sector with respect
to QoS degradation."
::= { wmanIfBsNetNotificationObjectsEntry 1 }

wmanIfBsNetDegradationUsefulPercentage OBJECT-TYPE
SYNTAX Unsigned32 (1 .. 1000000)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The percentage of the reserved network utilization that can
still be provided by the network during the degradation period.
It can be mathematically expressed as the ratio between the time
the network is able to use to assure bandwidth reservation
constraints and the time the network would need to be using to
assure bandwidth reservation constraints without any degradation.
The complement of this value can be viewed as the percentage of
network degradation.
This value is expressed with four digits fixed point in a range
from 0.0001% to 100.0000%
The idea is not to have an exact value, but to give a hint to
the management system about the criticity of the degradation"
::= { wmanIfBsNetNotificationObjectsEntry 2 }
```

(. . .)

APÊNDICE E NOVAS NOTIFICAÇÕES NA MIB

(. . .)

```
wmanIfBsSsDynamicServiceSuccessTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsSsNotificationMacAddr,
wmanIfBsDynamicServiceSfId}
STATUS current
DESCRIPTION
"An event to report the success of a dynamic service
operation (DSA, DSC or DSD).
This event is generated only for not provisioned flows or
operations started at SS side."
::= { wmanIfBsTrapPrefix 2 }

(...)

wmanIfBsSsServiceAdmissionTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsSsNotificationMacAddr,
wmanIfBsServiceAdmissionSfId,
wmanIfBsServiceAdmissionStatus,
wmanIfBsServiceAdmissionFailReason}
STATUS current
DESCRIPTION
"An event to report the result of an admission attempt
for a provisioned service flow.

If wmanIfBsSsServiceAdmissionSuccess is set, this trap is
sent for bsSfAdmissionSuccess and bsSfAdmissionDegraded
as they both indicate a service flow has been admited in
the network.

If wmanIfBsSsServiceAdmissionFail is set, this trap is sent
for bsSfAdmissionDegraded and bsSfAdmissionFail statuses
as they both indicate a service flow couldn't be admited the
way it has been provisioned or that a provisioned service flow
couldn't be admited at all.

wmanIfBsServiceAdmissionFailReason only makes sense if
wmanIfBsServiceAdmissionStatus is different from
bsSfAdmissionSuccess.

In case of degraded admission, the admited Service Flow
```

```
parameters can be obtained in the wmanIfCmnCpsServiceFlowTable
table."
::= { wmanIfBsTrapPrefix 4 }

(...)

wmanIfBsSsConnectTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsSsNotificationMacAddr,
wmanIfBsSsConnectStatus}
STATUS current
DESCRIPTION
"An event to report that the connection status
of a SS has changed - it entered or leaved the
wmanIfBsConnectedSsTable."
::= { wmanIfBsTrapPrefix 6 }

wmanIfBsSsRegisterTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsSsNotificationMacAddr,
wmanIfBsSsRegisterStatus}
STATUS current
DESCRIPTION
"An event to report SS registration status."
::= { wmanIfBsTrapPrefix 7 }

wmanIfBsSsHandoverTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsSsNotificationMacAddr,
wmanIfBsSsHandoverStatus}
STATUS current
DESCRIPTION
"An event to report SS handover status."
::= { wmanIfBsTrapPrefix 8 }

wmanIfBsSsPkmFailTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsSsNotificationMacAddr}
STATUS current
DESCRIPTION
"An event to report the failure of a Pkm operation."
::= { wmanIfBsTrapPrefix 9 }

wmanIfBsSsBurstProfileChangeTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsSsNotificationMacAddr,
wmanIfBsSsMaxUplinkTransmission,
wmanIfBsSsMaxDownlinkTransmission}
STATUS current
DESCRIPTION
"An event to report that the burst profile for a
registered subscriber station has changed.
It is expected that the upload and download
transmission rates change as well."
::= { wmanIfBsTrapPrefix 10 }

wmanIfBsSsServiceDegradationTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsSsNotificationMacAddr,
```

```
wmanIfBsServiceDegradationSfId,
wmanIfBsServiceDegradationStatus,
wmanIfBsServiceDegradationUsefulPercentage}
STATUS current
DESCRIPTION
"An event to report that a service flow started or
stoped being degraded.

The first bsSfDegradationStarted notification must be
sent if wmanIfBsServiceDegradationUsefulPercentage
for a specific service flow drops bellow
wmanIfBsServiceDegradationStartThreshold.

A new bsSfDegradationStarted event shall be sent if
wmanIfBsServiceDegradationUsefulPercentage drops
down by wmanIfBsServiceDegradationStepThreshold
after the last event was sent for a specific
service flow.

A bsSfDegradationEnded event must be sent if
wmanIfBsServiceDegradationUsefulPercentage stays
consistently above or equal to
wmanIfBsServiceDegradationStartThreshold during the time
specified by wmanIfBsServiceDegradationHysteresisTime."
::= { wmanIfBsTrapPrefix 11 }

wmanIfBsNetDegradationTrap NOTIFICATION-TYPE
OBJECTS {ifIndex,
wmanIfBsNetDegradationStatus,
wmanIfBsNetDegradationUsefulPercentage}
STATUS current
DESCRIPTION
"An event to report the BS sector as a whole is
suffering degradation. That means that at least some
of the service flows being served by the BS in the
sector are suffering from degradation.

The criteria to distinguish the cases when the BS sector
is suffering from degradation from the cases when the
problem is just with a few SSs won't be stated here.
The idea of this notification is to indicate that
the delivered service level of some degraded service flows
could be restored to the expected level (or at least made
better) if service flows of other SSs were intentionally
degraded. That means the management system might try to solve
the degradation for the critical flows by turning the
requirements for non critical flows less strict (even if the
non critical flows are of a subscriber station distinct than
that of the critical flows).

The first bsNetDegradationStarted notification must be
sent if wmanIfBsNetDegradationUsefulPercentage drops bellow
wmanIfBsNetDegradationStartThreshold.

A new bsNetDegradationStarted event shall be sent if
wmanIfBsNetDegradationUsefulPercentage drops down by
wmanIfBsNetDegradationStepThreshold after the last event
was sent.
```

```
A bsNetDegradationEnded event must be sent if
wmanIfBsNetDegradationUsefulPercentage stays
consistently above or equal to
wmanIfBsNetDegradationStartThreshold during the time
specified by wmanIfBsNetDegradationHysteresisTime."
::= { wmanIfBsTrapPrefix 12 }
```

(. . .)

APÊNDICE F DETALHES DE IMPLEMENTAÇÃO DO SIMULADOR DA MIB

O simulador da MIB proposta para IEEE802.16 foi implementado como uma extensão do agente SNMP disponível no pacote NET-SNMP (NET-SNMP, s.d.). Ele simula o comportamento de uma estação base com relação à MIB proposta, sendo possível especificar ações das estações cliente sob esta estação base, incluindo processos de *handover* realizados de/para estações bases vizinhas, representadas por outras instâncias do simulador.

A extensão do agente SNMP foi feita como um módulo de carga dinâmica, o que permite que o simulador seja compilado e carregado de forma independente do executável principal (*snmpd*). É necessário, entretanto, que o agente SNMP principal tenha sido compilado com suporte a carga dinâmica de módulos.

Dois arquivos devem ser configurados para que o simulador torne-se funcional: o *snmpd.conf* e o *sim.txt*. O primeiro arquivo (*snmpd.conf*) é o arquivo geral de configuração do agente *snmpd*. Neste arquivo devem-se incluir configurações que indiquem a carga do módulo do simulador e a descrição do ambiente de simulação. O segundo arquivo (*sim.txt*) é o arquivo de simulação propriamente dito. O simulador atua como um simulador discreto de eventos, cuja unidade de tempo é o segundo (a implementação do *snmpd* impede a simulação com eventos de granularidade mais fina). O arquivo *sim.txt* descreve os eventos que ocorrem em cada segundo a partir do início da simulação (do inicio do programa *snmpd*).

Exemplos das configurações válidas no arquivo *snmpd.conf* podem ser vistas no Quadro F.1 (*baseStation.so* é o nome do arquivo que representa o módulo do simulador):

```
#Loads the BS module
dlmod baseStation /path/to/baseStation.so
bsId 2
bsPort 5002
bsConnect 127.0.0.1 5001
bsSimulationFile sim.txt

# Specify trap destinations
trapcommunity public
#trap2sink 10.0.3.2
informsink 10.0.3.2
```

Quadro F.1: Exemplo de configuração do simulador (*snmpd.conf*)

A configuração *dlmod* instrui o agente *snmpd* a carregar o módulo do simulador (baseStation) no caminho indicado. A configuração *bsId* indica ao módulo do simulador qual será o ID da estação base sendo simulada (este ID poderá ser utilizado no arquivo *sim.txt*, posteriormente). A configuração *bsPort* indica ao simulador em que porta TCP da máquina local este deverá esperar conexões de simuladores que simulam estações bases vizinhas (as estações móveis somente podem efetuar *handover* entre estações bases vizinhas). A configuração *bsConnect* instrui o simulador a conectar-se a uma estação base vizinha (outra instância de simulador), representada por seu par IP e porta. Ressalta-se que, para conectar duas estações bases vizinhas, basta que uma conecte-se a outra. Por fim, a configuração *bsSimulationFile* instrui o simulador a carregar o arquivo de simulação com o nome *sim.txt*.

As configurações *trapcommunity*, *trap2sink* e *informsink* são configurações padrão do agente *snmpd* relativas ao envio de notificações. Sua configuração é relevante à medida em que o simulador envia notificações da MIB definida para IEEE802.16. Ressalta-se que a utilização de *informsink* é mais apropriada do que *trap2sink*, visto que as mensagens InformRequest possuem confirmação, o que é algo desejável para evitar perda de eventos na rede.

Todos os comandos de simulação que podem constar no arquivo *sim.txt* também podem ser utilizados através de uma conexão de rede com a porta configurada para comunicação entre estações base (simuladores). O simulador não distingue entre comandos no arquivo de simulação e comandos recebidos via rede, embora tipicamente as estações base troquem comandos distintos dos utilizados em arquivos de simulação (como será visto adiante). A utilização de um protocolo baseado em texto permite que o aplicativo *telnet* seja utilizado para enviar comandos de simulação em tempo de execução para o simulador.

A sintaxe dos comandos de simulação (arquivo *sim.txt*) pode ser vista no Quadro F.2. Antes de cada comando deve haver a indicação do tempo (*time*) em que o comando deve ser executado pelo simulador (em segundos).

```
<time> connect <mac>
<time> disconnect <mac>
<time> register <mac> <maxUp> <maxDown>
<time> unregister <mac>
<time> handover <mac> <toBsId>
<time> waitHandover <mac> <fromBsId>
<time> changeProfile <mac> <maxUp> <maxDown>
<time> degradeNet <servicePercentage>
```

Quadro F.2: Comandos de simulação

Os comandos *connect* e *disconnect* simulam a conexão e desconexão de uma estação cliente. O formato a ser utilizado para o MAC é um número hexadecimal comum, não separado por ":" como na notação padrão de MAC. Embora não existam tais MACs na prática, o simulador pode trabalhar com MACs de menos de 48bits para facilitar a visualização (basta utilizarem-se menos dígitos).

Os comandos *register* e *unregister* simulam o registro e desregistro de uma estação cliente na estação base. A simulação do registro somente pode ser feita com a estação cliente conectada, e os parãmetros *maxUp* e *maxDown* indicam as taxas de transmissão (em kbps) da estação cliente em *upload* e *download* respectivamente. De modo similar,

a simulação de uma desconexão somente pode ser feita com a estação cliente desregistrada. Eventos que não podem ocorrer na prática são ignorados pelo simulador, que emite um aviso sobre o problema.

Os comandos *handover* e *waitHandover* são utilizados em conjunto. Seu efeito é o de uma sincronização entre as estações base (simuladores) origem e destino do *handover* de uma estação móvel (o tempo de simulação do simulador que estiver adiantado pára até que ambas as simulações estejam sincronizadas no momento do *handover*). O comando *handover* indica para a estação base de origem que a estação móvel com o MAC indicado irá efetuar *handover* para a estação base que possui o BSID informado. A estação móvel deverá estar registrada na estação base de origem para que possa efetuar o *handover*. O comando *waitHandover* indica para a estação base de destino que a estação móvel com o MAC indicado irá efetuar *handover* a partir da estação base que possui o BSID informado. A estação móvel deverá estar conectada (e não registrada) na estação base de destino para que possa efetuar o *handover*. Os comandos *handover* e *waitHandover* promovem apenas a sincronização entre as estações base. Outros comandos de simulação deverão ser incluídos para provocar o registro da estação móvel na estação base destino e o desregistro e desconexão da estação móvel na estação base de origem. O comando *handover* também pode causar o envio da notificação wmanIfBsSsHandOver, dependendo da configuração feita via MIB na estação base.

Conforme a especificação IEEE802.16e, após um *handover*, o estado de uma estação móvel deve ser mantido em sua estação base de origem por um período ditado pelo temporizador *Resource Retain Time*. Este temporizador, que visa possibilitar o cancelamento do *handover*, pode ser configurado para um tempo entre 0,1s e 25,5s. O desregistro e desconexão de uma estação móvel na estação base de origem devem ser simulados para ocorrer somente após passado este tempo do momento do *handover*, visando uma simulação mais realista.

O comando *changeProfile* visa simular a alteração do perfil de transmissão ou recepção de uma estação cliente registrada. Tipicamente, perfis mais robustos apresentam taxas de transmissão menores, ao passo que perfis mais eficientes apresentam maior taxa de transmissão e são menos robustos. O comando permite alterar diretamente as taxas de transmissão em *upload* (*maxUP*) e *download* (*maxDown*) da estação cliente que possui o MAC informado (em kbps). Se a alteração das taxas de transmissão provocar degradação de algum serviço da estação cliente, a notificação wmanIfBsSsServiceDegradationTrap poderá vir a ser enviada, dependendo da configuração feita via MIB na estação base.

Por fim, o comando *degradeNet* visa simular uma condição de rede que cause a alteração concomitante dos perfis de transmissão de todas as estações cliente na rede. O parâmetro *servicePercentage* é um inteiro que indica percentualmente o nível dos perfis de transmissão na rede comparado com àquele que estaria sendo simulado para as estações. Supondo uma estação que teria um perfil que indicasse a taxa de 100kbps, um *servicePercentage* igual a 75 faria com que a taxa simulada para a estação fosse de 75kbps. Se, posteriormente, utilizasse-se o comando *changeProfile* para alterar a taxa de transmissão da estação para 200kbps, a estação teria sua taxa alterada para 150kbps até que o *servicePercentage* fosse configurado novamente para 100. Caso o serviço da rede degrade por conta da alteração do *servicePercentage*, a notificação

wmanIfBsNetDegradationTrap poderá vir a ser enviada, dependendo da configuração feita via MIB na estação base.

Além dos comandos utilizados para simulação, outros comandos são utilizados especificamente na comunicação entre estações base (simuladores). Embora se possa marcar estes comandos com um tempo qualquer, tipicamente o tempo 0 (zero) é utilizado para forçar a interpretação imediata dos mesmos. Se um comando é enviado ao simulador informando um tempo inferior ao tempo atual de simulação, o comando é processado imediatamente. O Quadro F.3 apresenta os comandos do protocolo entre estações base.

```
0 end
0 bsId <bsId>
0 handoverConfirm <mac> <toBsId>
0 waitHandoverConfirm <mac> <fromBsId>
```

Quadro F.3: Comandos do protocolo entre estações base

O comando *end* é o mais simples. Serve apenas para sinalizar que a comunicação se encerrou, e que a conexão deve ser fechada.

Logo que a conexão entre duas estações base (simuladores) é estabelecida, cada estação base se identifica enviando o comando *bsId* e informando o seu BSID, conforme configurado no arquivo *snmpd.conf*. Isto permite que cada simulador saiba através de qual conexão deverá comunicar-se quando estiver processando os comandos de *handover*, que indicam explicitamente o BSID da outra estação base envolvida no processo. Não é obrigatória a execução do comando *bsId*, sendo que o mesmo não precisa ser executado quando a conexão é realizada via *telnet*, por exemplo. Neste caso, a conexão não representa outra estação base para o simulador. A execução do comando duas vezes fará ambos os BSIDs informados serem vinculados à conexão em questão.

Os comandos *handoverConfirm* e *waitHandoverConfirm* são, de fato, notificações utilizadas no mecanismo de sincronismo que simula um *handover*. Quando o comando *handover* é executado, o simulador envia a notificação *waitHandoverConfirm* informando o MAC da estação móvel e o seu BSID para a estação base destino do *handover*, e pára o tempo de simulação até receber a notificação *handoverConfirm* relacionada. Quando o comando *waitHandover* é executado, o simulador envia a notificação *handoverConfirm* informando o MAC da estação móvel e o seu BSID para a estação base de origem do *handover*, e pára o tempo de simulação até receber a notificação *waitHandoverConfirm* relacionada. Se notificações *handoverConfirm* ou *waitHandoverConfirm* são recebidas antes do momento em que são necessárias, elas são armazenadas até a execução do comando de *handover* que as consome, que, neste caso, não causará a parada do tempo de simulação.

Com relação à implementação da MIB pelo simulador, ele utiliza o índice de interface 1 para simular a interface IEEE802.16 a ser gerenciada. Tipicamente esta interface é a interface de *loopback*, mas o simulador não interage propriamente com a mesma. As tabelas implementadas são apresentadas abaixo. A descrição das mesmas é mencionada para auxiliar na compreensão.

- *wmanIfBsProvisionedSfTable*: *Service Flows* provisionados. É através desta tabela que o sistema de gerenciamento cria os *Service Flows* na rede.

- *wmanIfBsSsProvisionedForSfTable*: associação de *Service Flows* provisionados a estações cliente. Quando a estação cliente se registra, os *Service Flows* associados a ela são negociados e passam a existir na rede (passam a constar na tabela *wmanIfCmnCpsServiceFlowTable*).
- *wmanIfBsServiceClassTable*: classes de QoS. Uma classe de QoS mantém um conjunto de requisitos que devem ser fornecidos aos *Service Flows* associados. A tabela *wmanIfBsProvisionedSfTable* possui um campo para indicar a classe de QoS de cada *Service Flow* provisionado. O campo *wmanIfBsQoSMinReservedRate* é utilizado, em conjunto com as taxas de transmissão das estações cliente, como base para medição do consumo de recursos na rede pelo simulador (recursos já reservados). Os demais requisitos de QoS são ignorados pelo simulador.
- *wmanIfBsClassifierRuleTable*: classificadores de pacotes nos *Service Flows* provisionados. Classificadores não possuem efeito prático no simulador, mas são copiados para a tabela *wmanIfCmnClassifierRuleTable* quando o *Service Flow* para o qual estão realizando classificação passa a existir na rede.
- *wmanIfBsConnectedSsTable*: estações cliente conectadas na estação base. As entradas nesta tabela são diretamente afetadas pelos comandos de simulação *connect* e *disconnect*. Apenas o índice (que inclui o MAC) e os campos de identificadores de conexão são preenchidos (estes últimos com zero).
- *wmanIfBsRegisteredSsTable*: estações cliente registradas na estação base. As entradas nesta tabela são diretamente afetadas pelos comandos de simulação *register* e *unregister*. Apenas o índice (que inclui o MAC), os campos de identificadores de conexão e os campos de taxa de transmissão são preenchidos. Os identificadores de conexão são preenchidos com zero.
- *wmanIfBsConfigurationTable*: configurações da estação base. Apenas a configuração de reserva mínima de recursos para fluxos do tipo melhor esforço (*wmanIfBsMinBestEffortResources*) foi implementada (o valor *default* é 10%). As outras configurações são ignoradas pelo simulador. Na linha de configuração criada pelo simulador, outros campos foram preenchidos com o valor assumido pelo simulador apenas para documentação: *wmanIfBsSsSfMgmtEnabled* (o gerenciamento de *Service Flows* pelas estações cliente é desabilitado), *wmanIfBsAutoSfIdEnabled* (a geração de SFIDs pela estação base é desabilitada), *wmanIfBsAutoSfidRangeMin* (1), *wmanIfBsAutoSfidRangeMax* (10000). Os dois últimos apenas complementam o segundo campo.
- *wmanIfBsThresholdConfigTable*: limiares para emissão de notificações. Os campos associados às notificações emitidas foram implementados.
- *wmanIfBsSsNotificationObjectsTable*: objetos enviados em notificações associadas a estações cliente.
- *wmanIfBsNetNotificationObjectsTable*: objetos enviados em notificações associadas à rede.
- *wmanIfCmnCpsServiceFlowTable*: *Service Flows* existentes na rede. Reúne informações tanto da tabela *wmanIfBsProvisionedSfTable*, quanto da tabela *wmanIfBsServiceClassTable*. Os *Service Flows* passam a existir nesta tabela

quando a estação cliente associada aos mesmos encontra-se registrada. Somente fluxos que estejam nos estados *admitted* ou *active* e possuam reserva de banda (campo *wmanIfCmnCpsMinReservedRate*) consomem recursos de rede. Fluxos *unicast* consomem recursos individualmente por estação cliente, e fluxos *multicast* consomem recursos de forma única para todas as estações cliente associadas (baseado na taxa de transmissão da estação mais lenta admitida para o fluxo). Vale notar que, se um fluxo estiver em situação de degradação, a rede pode não estar provendo os recursos que constam como negociados no campo *wmanIfCmnCpsMinReservedRate*. O valor negociado pode ser menor que o valor provisionado em função de uma admissão parcial, que pode ocorrer no simulador quando há alteração na reserva provisionada quando o fluxo já se encontrava admitido com uma reserva menor.

- *wmanIfCmnClassifierRuleTable*: classificadores de pacotes nos *ServiceFlows* existentes na rede. Copia das informações em *wmanIfBsClassifierRuleTable* para os fluxos que tiverem sido negociados.

O objeto *wmanIfBsTrapControlRegister* foi implementado para permitir a marcação das notificações que deverão ser emitidas pela estação base. As notificações implementadas estão listadas abaixo. A descrição das mesmas é mencionada para auxiliar na compreensão.

- *wmanIfBsSsServiceAdmissionTrap*: notificação enviada para indicar o resultado do controle de admissão sobre um *Service Flow* provisionado. Vale ressaltar que um *Service Flow* inicialmente não admitido pode, posteriormente, vir a ser admitido quando recursos forem liberados. Neste caso, a notificação será enviada duas vezes (uma para cada situação).
- *wmanIfBsSsConnectTrap*: notificação enviada para indicar a conexão ou desconexão de uma estação cliente.
- *wmanIfBsSsRegisterTrap*: notificação enviada para indicar o registro ou desregistro de uma estação cliente.
- *wmanIfBsSsHandoverTrap*: notificação enviada para indicar a tentativa de um *handover* ou o cancelamento do mesmo. Como o simulador não simula cancelamento de *handover*, somente a tentativa de *handover* é notificada.
- *wmanIfBsSsBurstProfileChangeTrap*: notificação enviada para indicar a alteração do perfil de transmissão ou recepção de uma estação cliente, causando alteração na taxa de transmissão. Esta notificação é diretamente associada ao comando *changeProfile* do simulador.
- *wmanIfBsSsServiceDegradationTrap*: notificação enviada para indicar inicio, fim ou alteração no estado de degradação de um *ServiceFlow*. Um *Service Flow* é degradado quando a rede não consegue atender aos requisitos de QoS que aceitou atender à princípio. No caso do simulador, a análise é baseada no requisito *wmanIfBsQoSMinReservedRate* negociado com todos os fluxos da rede. O início de uma degradação pode ocorrer em função do comando *changeProfile* do simulador.
- *wmanIfBsNetDegradationTrap*: notificação enviada para indicar início, fim ou alteração no estado de degradação generalizado entre os *Service Flows* na rede.

A análise é semelhante à realizada para a notificação *wmanIfBsSsServiceDegradationTrap*, mas uma degradação de rede somente pode ocorrer em função do comando de simulação *degradeNet* do simulador.

A implementação das tabelas da MIB foi feita utilizando-se tabelas gerenciadas pelo próprio NET-SNMP. Isto evitou que se tivesse que construir e gerenciar estruturas de dados específicas para cada tabela. Os dados são acessados diretamente nas próprias tabelas, exceto em alguns casos em que estruturas otimizadas redundantes foram mantidas por questões de performance.

Ao executar o simulador, recomenda-se ativar a depuração sobre as classes “baseStation” e “bsSimulator”. A primeira captura mensagens gerais do módulo que simula uma estação base (incluindo mensagens de erro), enquanto a segunda captura a passagem do tempo de simulação e os eventos que ocorrem. Embora o simulador tome as devidas precauções quanto à ocorrência de erros, por uma questão de tempo, não foram implementados mecanismos de recuperação, de modo que a simulação pode tornar-se inválida na ocorrência de erros. Deve-se estar atento às mensagens emitidas pelo simulador. A linha de execução recomendada para utilizar um arquivo de configuração no diretório local é esta:

- `snmpd -f -L -DbaseStation,bsSimulator,dlmod -C -c./snmpd.conf`

Um detalhe a ser mencionado é que houve uma confusão durante a implementação do simulador quanto à unidade de vazão adotada na tabela de classes de QoS da MIB. Por este motivo, a unidade utilizada para o campo *wmanIfBsQoSMinReservedRate* acabou sendo byte/segundo, ao invés de bit/segundo, como deveria ser.

APÊNDICE G EXTENSÃO AO ESQUEMA LDAP DO QAME

```
#-- qame ldap map
#=========== Attribute =================
attributetype   (  1.3.6.1.3.12619.1.1
     NAME 'qameResourcePriority'
     DESC 'Policy priority for resources on policy conflict.'
     EQUALITY numericStringMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.36
     SINGLE-VALUE
   )
attributetype   (  1.3.6.1.3.12619.1.2
     NAME 'qameEquivalenceSetDN'
     DESC 'DN reference to a qameEquivalenceSet entry.'
     EQUALITY distinguishedNameMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
     SINGLE-VALUE
   )

#========= class ============
objectClass    (  1.3.6.1.3.12619.2.1
     NAME 'qamePolicyRuleParameters'
     DESC 'Auxiliary class that contains parameters used for
           rules in QAME that are not available in the standard
           PCIM or PCIME classes'
     SUP top
     AUXILIARY
     MUST ( qameResourcePriority )
     MAY ( qameEquivalenceSetDN )
   )
#=========== Attribute =================
attributetype   (  1.3.6.1.3.12619.1.3
     NAME 'qameEquivalenceSetName'
     DESC 'The user-friendly name of this equivalence set.'
     EQUALITY caseIgnoreMatch
     ORDERING caseIgnoreOrderingMatch
     SUBSTR caseIgnoreSubstringsMatch
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
     SINGLE-VALUE
   )
```

```
#========= class ============
objectClass    (  1.3.6.1.3.12619.2.2
     NAME 'qameEquivalenceSet'
     DESC 'Structural class that represents a set of Rules
           written for the same purpose, but which must be
           used in different PEPs'
     SUP pcimPolicy
     STRUCTURAL
     MUST ( qameEquivalenceSetName )
   )
```