UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE ENGENHARIA ELÉTRICA

PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PROPOSTA DE ALGORITMO DE CONTROLE DE ADMISSÃO DE CONEXÕES BASEADO EM *THRESHOLD* PARA AS REDES IEEE 802.16

Claiton Luiz Soares

Maio

2009

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE ENGENHARIA ELÉTRICA

PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PROPOSTA DE ALGORITMO DE CONTROLE DE ADMISSÃO DE CONEXÕES BASEADO EM *THRESHOLD* PARA AS REDES IEEE 802.16

Claiton Luiz Soares

Dissertação apresentada à Universidade Federal de Uberlândia para obtenção do título de Mestre em Engenharia Elétrica, aprovada em 19 de junho de 2009 pela banca examinadora:

Gilberto Arantes Carrijo, Dr. (UFU)

Paulo Roberto Guardieiro, Dr. – Orientador (UFU)

Solange da Silva, Dra. (UCG)

Uberlândia, Maio de 2009.

Dados Internacionais de Catalogação na Publicação (CIP)

S676p Soares, Claiton Luiz, 1983-
Proposta de algoritmo de controle de admissão de conexões baseado em Threshold para as Redes IEEE 802.16 / Claiton Luiz Soares. - 2009.
125 f. : il.

Orientador: Paulo Roberto Guardieiro.
Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Redes de computação - Teses. 2. Sistemas de comunicação sem fio - Teses. I. Guardieiro, Paulo Roberto. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 681.3.02

Elaborado pelo Sistema de Bibliotecas da UFU / Setor de Catalogação e Classificação

PROPOSTA DE ALGORITMO DE CONTROLE DE ADMISSÃO DE CONEXÕES BASEADO EM *THRESHOLD* PARA AS REDES IEEE 802.16

Claiton Luiz Soares

Dissertação apresentada por Claiton Luiz Soares à Universidade Federal de Uberlândia como parte dos requisitos à obtenção do título de Mestre em Engenharia Elétrica.

Prof. Dr. Paulo Roberto Guardieiro Orientador	Prof. Dr. Alexandre Cardoso Coordenador do curso de Pós-Graduação

Dedicatória

Dedico este trabalho a meu pai.
Procurei por uma palavra, que tentasse representar o meu sentimento. Por ter mania de fazer cálculos: somar, subtrair, multiplicar e dividir. E pela razão de ter adquirido um raciocínio lógico e matemático, analisei e calculei a melhor palavra e a melhor forma de lhe expressar o meu sentimento. Deduzir por indução do coração junto com a alma, que a palavra é amor, e a frase é te amo. Sinto falta do seu carinho, do seu jeito, de tuas palavras, dos seus ensinamentos. Queria acordar e estar ao seu lado, mas o problema que não estou dormindo, luto por uma vida, por um futuro, para ser seu orgulho, onde quer que esteja. A única coisa que posso fazer agora é lhe presentear realizando um grande sonho teu. Por estes motivos dedico este trabalho a ele.

Agradecimentos

Agradeço primeiramente a Deus, pela oportunidade de fazer o mestrado e pela força e sabedoria que me auxiliou durante os meus estudos.

Ao Prof. Dr. Paulo Roberto Guardieiro, pela dedicação, paciência e ensinamentos durante todo o tempo, além disso, acreditando e confiando na minha pessoa.

Ao meu amigo e companheiro de laboratório, Ederson, que sempre me incentivou e me apoiou em todos os momentos em que precisei.

A todos os colegas do Mestrado que me apoiaram e a todos aqueles que me ajudaram de alguma forma para a realização deste trabalho.

À CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pelo apoio financeiro concedido.

À minha família que tanto amo...

Resumo

Soares, C. L., Proposta de Algoritmo de Controle de Admissão de Conexões Baseado em *Threshold* para as Redes IEEE 802.16, UFU, Uberlândia, Brasil, 2009, 125p.

O padrão IEEE 802.16, também conhecido como WiMAX (*Worldwide Interoperability for Microwave Access*), é uma das tecnologias mais promissoras para o acesso banda larga sem fio (BWA – *Broadband Wireless Access*). O padrão IEEE 802.16 fornece especificações das características da camada de acesso ao meio (MAC) e física. Agrega às redes BWA algumas vantagens em relação às outras tecnologias, tal como, ampla área de cobertura, mesmo em regiões de difícil acesso ou sem qualquer infra-estrutura de rede convencional, como é o caso de algumas regiões urbanas e rurais brasileiras. O padrão IEEE 802.16 foi desenvolvido com Qualidade de Serviço (QoS) em mente. Para conseguir tal objetivo, criou-se um padrão orientado a conexão, onde as várias aplicações são diferenciadas em múltiplas classes de serviços, de acordo com os parâmetros solicitados por cada aplicação. Porém, o padrão não define como deve ser implementado o algoritmo de Controle de Admissão de Conexões (CAC – *Connection Admission Control*), que é um requisito fundamental para obtenção de QoS. O algoritmo CAC é responsável por admitir ou rejeitar uma solicitação de uma nova conexão dependendo dos recursos já alocados da rede. Desta forma, o CAC deve rejeitar solicitações de conexões que poderiam comprometer a QoS das conexões admitidas. Em vista disto, neste trabalho apresenta-se uma proposta de um algoritmo de CAC baseado em *threshold* para as redes IEEE 802.16. Além disto, o desempenho do algoritmo de CAC proposto foi analisado através de modelagem e simulação, utilizando o simulador de redes NS-2 (*Network Simulator*). Para realizar esta etapa, o algoritmo de CAC proposto foi implementado no módulo WiMAX do NIST (*National Institute of Standards and Technology*). Os resultados obtidos demonstraram que o algoritmo de CAC é eficiente e apresentou-se capaz de prover QoS em termos de largura de banda e atraso, sob diferente cenários.

Palavras-chave: IEEE 802.16, WiMAX, BWA, QoS, CAC.

Abstract

Soares, C. L., Proposed Threshold-Based Connection Admission Control (CAC) Algorithm for IEEE 802.16 Networks, UFU, Uberlândia, Brazil, 2009, 125p.

IEEE 802.16 standard, also called WiMAX (Worldwide Interoperability for Microwave Access), is one of the most promising technologies for BWA (Broadband Wireless Access) networks. The IEEE 802.16 standard provides specification of MAC and physical layer. The standard adds to the BWA networks some advantages over other technologies, such as wide area of coverage, even in areas of difficult access or without conventional network infrastructure, as is the case of some urban and rural regions in Brazil. IEEE 802.16 standard is developed with Quality of Service (QoS) in mind. For that purpose it was created a connection oriented standard, where different applications are differentiated into multiple classes of service, in accordance with the parameters required by each application. However, the standard does not define how to implement the Connection Admission Control (CAC) algorithm, which is a fundamental requisite for obtaining QoS. The CAC algorithm is responsible for accepting or rejecting a new connection request depending on the resources already allocated in the network. Thus, the CAC must reject requests for connections that might compromise the QoS of admitted connections. This way, this work presents the proposal of a CAC algorithm based on threshold for the IEEE 802.16 networks. Moreover, the performance of the proposed CAC algorithm has been analyzed through modeling and simulation using the NS-2 (Network Simulator). To perform this step, the proposed CAC algorithm has been implemented in the WiMAX module of the NIST (National Institute of Standards and Technology). The results showed that the CAC algorithm is efficient and it is able to provide QoS in terms of bandwidth and delay, under different scenarios.

Keywords: IEEE 802.16, WiMAX, BWA, QoS, CAC.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AMC	*Adaptive Modulation and Coding*
ARQ	*Automatic Repeat Request*
ATM	*Asynchronous Transfer Mode*
AWK	*Aho, Weinberger and Kernighan*
BE	*Best Effort*
BPSK	*Binary Phase Shift Keying*
BR	*Bandwidth Request*
BRAN	*Broadband Radio Access Networks*
BS	*Base Station*
BWA	*Broadband Wireless Access*
CAC	*Connection Admission Control*
CBR	*Constant Bit Rate*
CI	CRC *Indicator*
CID	*Connection Identifier*
CRC	*Cyclic Redundancy Check*
CS	Subcamada de Convergência
DAMA	*Demand Assigned Multiple Access*
DCD	*Downlink Channel Descriptor*
DFS	*Dynamic Frequency Selection*
DHCP	*Dynamic Host Configuration Protocol*
DOCSIS	*Data Over Cable Service Interface Specification*
DRR	*Deficit Round Robin*

DSA	*Dynamic Service Addition*
DSA-ACK	*Dynamic Service Addition Acknowledgment*
DSA-REQ	*Dynamic Service Addition Request*
DSA-RSP	*Dynamic Service Addition Response*
DSC	*Dynamic Service Change*
DSD	*Dynamic Service Delete*
DSL	*Digital Subscriber Line*
EC	*Encriptation Control*
EDF	*Earliest Deadline First*
EKS	*Encriptation Key Sequence*
ertPS	*extended real-time Polling Service*
ETSI	*European Telecommunications Standards Institute*
FDD	*Frequency Division Duplexing*
FEBA	*Fair End-to-end Bandwidth Access*
FEC	*Forward Error Correction*
FFT	*Fast Fourier Transform*
FIFO	*First In First Out*
FTP	*File Transfer Protocol*
GloMoSim	*Global Mobile Information Systems Simulation Library*
GPC	*Grant Per Connection*
GPS	*Generalized Processor Sharing*
GPSS	*Grant Per Subscriber Station*
HCS	*Header Check Sequence*
HiperACCESS	*High Performance Radio Access*

HiperMAN	*High Performance Radio Metropolitan Area Network*
HT	*Header Type*
IEEE	*Institute of Electrical and Electronics*
ITU	*International Telecommunication Union*
LEN	*Length*
LoS	*Line of Sight*
LSB	*Least Significant Bit*
LWX	*Light WIMAX*
MAC	*Medium Access Control*
MBAC	*Measurement-Based Admission Control*
MIMO	*Multiple-Input Multiple-Output*
MMS	*Multimedia Messaging Service*
MPEG	*Motion Picture Expert Group*
MSB	*Most Significant Bit*
MUFSS	*Multi-class Uplink Fair Scheduling Structure*
MWFQ	*Modified* WFQ
MWRR	*Modified* WRR
NCTUns	*National Chiao Tung University Network Simulator*
NDSL	*Networks & Distributed Systems Laboratory*
NIST	*National Institute of Standards and Technology*
NLoS	*Non Line of Sight*
nrtPS	*Non-real-time Polling Service*
NS	*Network Simulator*
OFDM	*Orthogonal Frequency Division Multiplexing*

OMNeT++	*Objective Modular Network Testbed in* C++
PARSEC	*Parallel Simulation Environment for Complex Systems*
PBAC	*Parameter-Based Admission Control*
PDU	*Protocol Data Unit*
PGPS	*Packet Generalized Processor Sharing*
PHS	*Packet Header Suppression*
PKM	*Privacy Key Management*
PMP	*Point to Multi-Point*
QAM	*Quadrature Amplitude Modulation*
QoS	*Quality of Service*
QPSK	*Quadrature Phase Shift Keying*
RR	*Round Robin*
RSV	*Reserved*
rtPS	*real-time Polling Service*
SA	S*ecurity Association*
SAID	S*ecurity Association Identifier*
SAP	*Service Access Point*
SC	*Single Carrier*
SCFQ	*Self-Clocking Fair Queuing*
SDU	*Service Data Unit*
SFID	*Service Flow Identifier*
SLA	*Service Level Agreement*
SMS	*Short Message Service*
SNMP	*Simple Network Management Protocol*

SNR	*Signal-to-Noise Ratio*
SS	*Subscriber Station*
TDD	*Time Division Duplexing*
TDM	*Time Division Multiplexing*
TDMA	*Time Division Multiple Access*
TFTP	*Trivial File Transfer Protocol*
UCD	*Uplink Channel Descriptor*
UGS	*Unsolicited Grant Service*
VBR	*Variable Bit Rate*
VC	*Virtual Channel*
VCI	VC *Identifiers*
VINT	*Virtual InterNetwork Testbed*
VoIP	*Voice over* IP
VP	*Virtual Path*
VPI	VP *Identifiers*
WF^2Q	*Worst-case Fair Weighted Fair Queuing*
WFQ	*Weighted Fair Queuing*
WIMAX	*Worldwide Interoperability for Microwave Access*
WLAN	*Wireless Local Area Network*
WMAN	*Wireless Metropolitan Area Network*
WPAN	*Wireless Personal Area Network*
WRR	*Weighted Round Robin*
xDSL	*Various Digital Subscriber Line Technologies*

Capítulo 1

INTRODUÇÃO

As redes banda larga sem fio estão se tornando uma alternativa viável às redes banda larga tradicionais, permitindo aos seus usuários executarem as mesmas tarefas, bem como, proporcionando aos usuários mais flexibilidade e mobilidade, não apenas dentro de suas corporações, mas também fora delas. Neste contexto, as principais tecnologias de redes sem fio: WPAN (*Wireless Personal Area Network*), WLAN (*Wireless Local Area Network*) e WMAN (*Wireless Metropolitan Area Network*), têm proporcionado vantagens sobre as redes cabeadas, bem como algumas desvantagens e limitações. Dentre as vantagens das redes sem fio sobre as cabeadas, pode-se destacar a facilidade de implementação, a flexibilidade dentro da área de cobertura, redução do custo agregado e a topologia dinâmica. Dentre as desvantagens em relação às redes cabeadas podem-se citar: as dificuldades para provisão de QoS (*Quality of Service*), as questões de segurança, a limitação de largura de banda, etc. Diante disto, o padrão IEEE 802.16 foi desenvolvido com o objetivo de minimizar estes problemas.

O padrão IEEE 802.16, também conhecido como *WiMAX* (*Worldwide Interoperability for Microwave Access*), especifica as características da camada física (PHY) e da camada de acesso ao meio (MAC). Este padrão foi elaborado com suporte à Qualidade de Serviço.

A escolha do padrão IEEE 802.16 como tema fundamental deste trabalho se deve ao fato do padrão IEEE 802.16 apresentar alguns pontos em aberto, ser uma tecnologia nova e

ser uma das tecnologias mais promissoras para o acesso banda larga sem fio. Além disso, o desenvolvimento das redes banda larga sem fio proporcionou uma revolução na maneira como certos serviços são oferecidos.

As redes banda larga sem fio baseadas no padrão IEEE 802.16 apresentam algumas características particulares, tais como a capacidade de prover QoS, a oferta de altas taxas de dados e uma ampla área de cobertura num ambiente sem fio de rede metropolitana. Além disto, a área de cobertura do padrão IEEE 802.16 é mais ampla, se comparada com as outras tecnologias sem fio, oferecendo serviços em regiões suburbanas e rurais, sem qualquer infra-estrutura ou recursos de comunicação, ou seja, em regiões típicas de países subdesenvolvidos, como o Brasil.

Apesar do padrão IEEE 802.16 apresentar uma arquitetura que fornece suporte a QoS, deixa em aberto alguns pontos fundamentais. Em muitos aspectos, o padrão IEEE 802.16 propõe diretrizes sobre “o que fazer”, mas deixa a cargo do desenvolvedor “como fazer”. O padrão IEEE 802.16 apenas propõe diretrizes a serem seguidas, mas não especifica como serão implementados os mecanismos que garantam as funcionalidades especificadas. O padrão IEEE 802.16 deixa em aberto dois dos principais mecanismos na provisão de QoS: o algoritmo de escalonamento e o Controle de Admissão de Conexões (CAC). Assim, os fabricantes de equipamentos em conformidade com o padrão IEEE 802.16 devem implementar estes mecanismos, que são fundamentais na provisão de QoS.

Na literatura encontram-se alguns trabalhos que abordam a provisão de Qualidade de Serviço nas redes IEEE 802.16. Destes, a maioria trata da questão de escalonamento, com poucos visando mecanismos de CAC. Diante deste cenário, o estudo do CAC torna-se bastante motivador, atraente e empolgante.

O mecanismo de CAC é responsável por gerenciar a admissão ou rejeição de conexões na rede, sendo de fundamental importância para o bom desempenho da rede como um todo.

Em uma rede sem CAC todas as solicitações de conexões são admitidas. Assim o enlace pode ser saturado, pois a rede não tem nenhum mecanismo que gerencie a admissão de novas conexões, podendo ocasionar descartes de pacotes e atrasos não aceitáveis para aplicações multimídias.

Em outras palavras, o mecanismo de CAC pode ser definido como o conjunto de ações tomadas pela rede durante a fase de estabelecimento da conexão que define quando um pedido de conexão pode ser aceito ou rejeitado, dependendo das condições da rede. Caso o pedido de conexão seja aceito, o mecanismo de CAC deve garantir que a QoS das conexões admitidas não sejam prejudicadas com a admissão de uma nova conexão.

Tendo em vista o número escasso de trabalhos que abordam sobre os algoritmos de CAC para as redes de acesso IEEE 802.16, o objetivo deste trabalho é apresentar uma proposta de um algoritmo de CAC para as redes de acesso IEEE 802.16. O desempenho do algoritmo de CAC proposto é analisado através de modelagem e simulação.

Para desenvolver esta etapa, o algoritmo de CAC proposto foi implementado em um módulo WiMAX do NS-2 (*Network Simulator*), desenvolvido pelo NIST (*National Institute of Standards and Technology*). O módulo WiMAX do NIST foi escolhido por ser amplamente utilizado pela comunidade de pesquisa e não ter qualquer mecanismo de CAC implementado. Com base nos resultados obtidos, artigos foram produzidos e submetidos a diversos congressos [66-71].

Este trabalho está organizado da maneira descrita a seguir:

O Capítulo 2 aborda as principais definições do padrão IEEE 802.16 e a sua evolução seguindo uma ordem cronológica. A camada física e a camada MAC são descritas neste capítulo, bem como a apresentação do modelo de referência e as subcamadas da MAC. Além disso, aspectos relativos às interfaces aéreas da camada física também são brevemente descritos, bem como as topologias de rede PMP (*Point-to-Multipoint*) e *Mesh*.

No Capítulo 3 apresenta-se a questão da QoS, desde suas bases teóricas até a arquitetura QoS definida pelo padrão IEEE 802.16, além de enfatizar dois dos principais mecanismos deixados em aberto pelo padrão IEEE 802.16: o mecanismo de escalonamento e CAC. Além disto, apresenta-se as especificações da teoria do modelo de objetos, as classes e fluxos de serviço, bem como a classificação dos fluxos de serviço do padrão IEEE 802.16. Este capítulo também realiza um levantamento bibliográfico das principais disciplinas de escalonamento encontradas na literatura e destacam-se as principais características desejáveis de uma disciplina de escalonamento. E por fim, apresentam-se algumas das principais propostas de escalonamento para o padrão IEEE 802.16, seguindo a classificação de [17], em três grupos: disciplinas de escalonamento homogêneas, híbridas e oportunistas.

O Capítulo 4 aborda o mecanismo de CAC e apresenta algumas das principais propostas de CAC encontradas na literatura, bem como, a descrição da proposta de um algoritmo de CAC baseado em *threshold* para as redes IEEE 802.16.

O Capítulo 5 analisa a proposta do algoritmo de CAC baseado em *threshold* para as redes IEEE 802.16 através de modelagem e simulação. Ainda neste capítulo, são destacadas algumas das principais ferramentas de simulação e como o mecanismo de CAC foi implementado e incorporado ao simulador.

E, por fim, apresentam-se as conclusões finais e os trabalhos futuros relativos ao tema abordado.

Capítulo 2

REDES DE ACESSO BANDA LARGA SEM FIO BASEADAS NO PADRÃO IEEE 802.16

2.1. Introdução

As redes de acesso banda larga sem fio (BWA – *Broadband Wireless Access*) tornaram-se uma solução tecnicamente e economicamente viável para fornecer acesso a Internet para usuários em regiões urbanas e rurais. Suas principais vantagens são a facilidade de implantação, que resulta em economia de custos, bem como a capacidade de fornecer acesso a regiões remotas, regiões de difícil acesso e rurais, sem qualquer infra-estrutura cabeada. Neste capítulo, será apresentado um dos principais padrões para as redes BWA, o padrão IEEE 802.16, que é considerado uma solução atrativa para as redes banda larga sem fio.

Este capítulo será organizado da seguinte forma: A Seção 2.2 descreve sucintamente o padrão IEEE 802.16. A Seção 2.3 apresenta algumas recomendações do padrão IEEE 802.16. Em seguida, a Seção 2.4 aborda os dois tipos de topologias suportadas pelo padrão IEEE 802.16: PMP (*Point to Multipoint*) e *Mesh*. Na Seção 2.5 apresenta o modelo de referência especificado para o padrão IEEE 802.16, enfatizando alguns detalhes da camada de acesso ao meio (MAC) e da camada física. Finalmente, na Seção 2.6 serão apresentadas as considerações finais referente a este capítulo.

2.2. Padrão IEEE 802.16

O padrão IEEE 802.16 é uma das tecnologias mais promissora para acesso banda larga sem fio em redes metropolitanas, com desempenho comparável ao das tecnologias tradicionais, como a cabo, DSL (*Digital Subscriber Line*) ou serviço E1/T1. O padrão IEEE 802.16, também conhecido como WIMAX (*World Interoperability for Microwave Access*) ou IEEE WirelessMAN (WMAN - Wireless *Metropolitan Area Network*), foi desenvolvido pelo IEEE (*Institute of Electrical and Electronics Engineers)* no intuito de especificar formalmente as redes de acesso banda larga sem fio para áreas metropolitanas [1]. O advento desta nova tecnologia de acesso sem fio possibilitou atender aos anseios das WMANs que necessitam de altas taxas de transmissão e precisam atender uma grande quantidade de usuários em uma ampla área de cobertura.

O *WIMAX Fórum* [2] foi criado em julho de 2001, com o objetivo de promover a compatibilidade e interoperabilidade dos equipamentos de acesso em redes banda larga sem fio. É uma organização sem fins lucrativos, formada por importantes empresas de equipamentos e componentes, tais como, a AT&T, Fujitsu, Intel, Siemens Mobile, Nozema, dentre outras. Atualmente, o *WIMAX Forum* está com mais de 400 filiados. O *WIMAX Forum* é o equivalente, ao *Wi-Fi (Wireless Fidelity) Alliance*, responsável pelo grande desenvolvimento e sucesso do *Wi-Fi* em todo o mundo.

O padrão IEEE 802.16 teve a sua primeira versão aprovada no final do ano de 2001. No entanto, foi lançado um projeto de revisão do padrão “IEEE 802.16 REVd” focando maior conformidade com os aspectos do padrão HIPERMAN (ETSI) e maior detalhamento das especificações de teste [1]. Desta forma, ele pôde ser definido como um padrão global, pois foi desenvolvido de modo a ser compatível com os padrões do ITU (*International Telecommunication Union*) e do ETSI (*European Telecommunications Standards Institute*), mais especificamente com os padrões HiperACCESS (*High Performance Radio Access*) e

HiperMAN (*High Performance Radio Metropolitan Area Network*) do projeto BRAN (*Broadband Radio Access Networks*) do ETSI .

2.3. Evolução do Padrão IEEE 802.16

O IEEE 802.16 *Working Group* [3] é o grupo responsável pelo desenvolvimento e documentação do padrão IEEE 802.16 e suas recomendações. Devido às exigências que foram somadas ao padrão, várias recomendações foram elaboradas e documentadas. Desta forma, o padrão teve o seu projeto concluído em 2004 com o lançamento final do documento IEEE 802.16 e suas recomendações “a”, “b” e “c”. A seguir serão apresentadas sucintamente, seguindo a ordem cronológica, algumas das recomendações do padrão IEEE 802.16 [4] [5]:

IEEE 802.16a: Foi a primeira recomendação do padrão IEEE 802.16 que cobre as freqüências de operação de 2 a 11 GHz. Assim, utiliza-se uma faixa de freqüência baixa que permite que o sinal penetre nos obstáculos. Deste modo, não requer linha de visada direta (NLoS - *Non Line of Sight*). O padrão IEEE 802.16a tem o objetivo de competir com tecnologias que provêem acesso na “última milha”, tais como *cable modem* e xDSL (*Various Digital Subscriber Line Technologies*), e propõe oferecer taxas de transmissão teóricas de até 100 Mbps e alcance máximo teórico de 50 quilômetros.

IEEE 802.16b: Trata questões referentes à Qualidade de Serviço, permitindo o uso de freqüências entre 5 e 6 GHz não licenciadas, e utiliza antenas fixas sem linha de visada.

IEEE 802.16c: Projetado para garantir a interoperabilidade entre diferentes fabricantes, através de protocolos e especificação de testes. Aponta para um conjunto de perfis para a operação do sistema na faixa de 10 a 66 GHz.

IEEE 802.16d ou IEEE 802.16-2004: O IEEE 802.16d (ratificado em junho de 2004) é conhecido como WIMAX Fixo ou WIMAX Nomádico (pela facilidade de remanejamento). Corresponde à atualização do padrão IEEE 802.16, que consolida as revisões dos padrões IEEE 802.16a e IEEE 802.16c em um único padrão, substituindo o padrão IEEE 802.16a como o padrão base. Entre as alterações pode-se destacar a provisão de suporte para antenas MIMO (*Multiple-Input Multiple-Output*), o que aumenta a confiabilidade do alcance com multipercurso.

IEEE 802.16e ou IEEE 802.16-2005: O padrão IEEE 802.16e (ratificado em dezembro de 2005) é conhecido como WiMAX Móvel. Introduz suporte à mobilidade ao padrão, apresenta compatibilidade com as especificações do padrão IEEE 802.16d e as especificações de mobilidade em WMANs. Este padrão inclui mobilidade com LoS e NLoS, em freqüências de 10-66 GHz e 2-11 GHz, respectivamente.

No entanto, diversas recomendações foram autorizadas desde então, e algumas se encontram em fase *draft* (IEEE 802.16h, IEEE 802.16i, IEEE 802.16j, etc.) e *pre-draft* (IEEE 802.16m). Maiores informações sobre estas recomendações podem ser encontradas em [3].

2.4. Topologia do Padrão IEEE 802.16

O padrão IEEE 802.16 foi desenvolvido para acesso banda larga sem fio em grandes extensões. A dimensão da área de cobertura das redes IEEE 802.16 depende de alguns fatores, como codificação, freqüência, topologia, etc. Contudo, a topologia da rede tem papel fundamental na provisão de acesso à última milha. O padrão IEEE 802.16 define dois tipos de topologias [1]: a ponto-multiponto (PMP - *Point to Multipoint*) e a *Mesh*.

2.4.1. Topologia Ponto-Multiponto (PMP)

Na topologia PMP a estação base (BS - *Base Station*) tem total controle e gerencia todo tráfego de dados dentro de sua área de cobertura (célula). Assim, quando uma estação assinante (SS - *Subscriber Station*) quiser se comunicar com outra terá que, obrigatoriamente, transmitir seus dados para a BS e, então, a BS encaminhará os dados à SS de destino [1]. Portanto, as redes de acesso banda larga sem fio funcionam como as redes celulares, ou seja, as SSs comunicam-se diretamente com a BS.

A BS é responsável pela coordenação da comunicação com as SSs. Assim, a BS deve ser posicionada em um lugar estratégico, para fornecer alcance para várias SSs simultaneamente. Na topologia PMP, a direção do tráfego é usada para distinguir o tipo do canal de dados: canal *uplink* e canal *downlink* [1] [5]. No canal *uplink* o tráfego é enviado no sentido da SS para a BS, enquanto no canal *downlink* o tráfego é enviado no sentido inverso. A Figura 2.1 ilustra a topologia PMP.

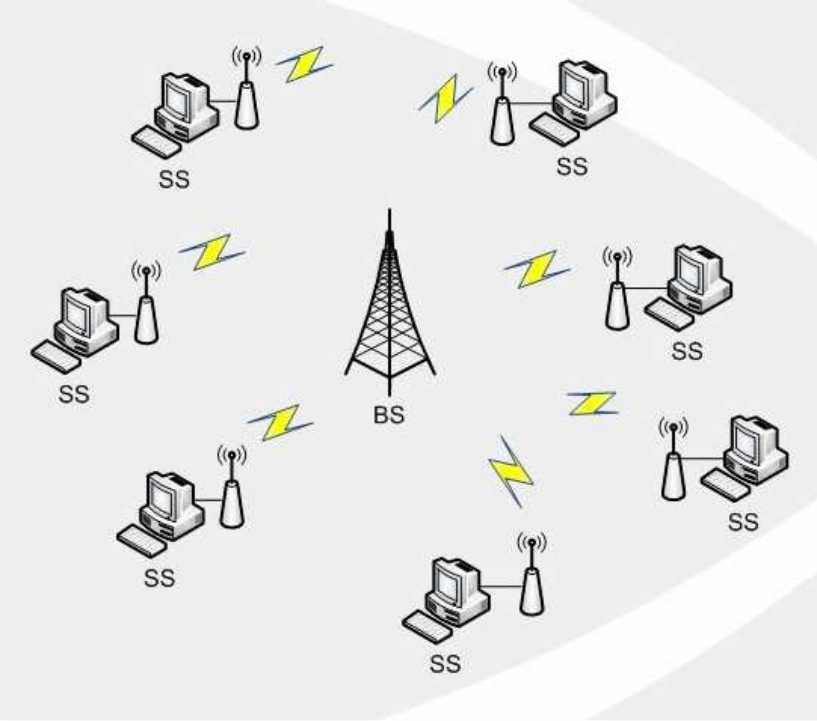

Figura 2.1: Topologia Ponto-Multiponto (PMP).

2.4.2. Topologia *Mesh*

Na topologia *Mesh* não existe a necessidade de comunicação direta entre a BS e as SSs, logo, as SSs podem comunicar com outras SSs diretamente [1]. Nesta topologia a BS não é responsável pela coordenação da comunicação entre as SSs. Assim, não há obrigatoriedade de transmitir os seus dados pela BS, desta forma, duas SSs podem trocar informações sem intermediários, como pode ser observado na Figura 2.2. Porém, a topologia *Mesh* exige algoritmos de roteamento complexos, pois operam no modo *ad hoc*, ou seja, operam sem a necessidade de um ponto central.

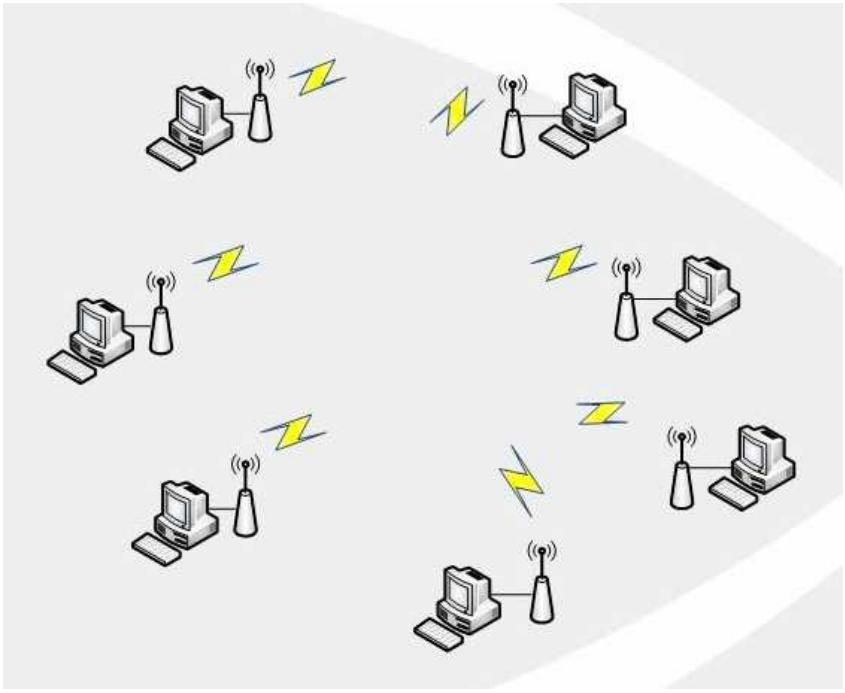

Figura 2.2: Topologia M*esh*.

Na topologia *Mesh* não se pode utilizar a direção do fluxo da BS para a SS, ou vice-versa, para distinguir o tipo do canal de dados, devido as SSs poderem se comunicar umas com as outras [5].

Uma vantagem da topologia *Mesh* em relação à PMP é que ela não possui um ponto único de falha, ou seja, caso a BS falhe na topologia PMP, todas as SSs de uma determinada célula ficarão impossibilitadas de se comunicar. Por outro lado, na topologia *Mesh*, se houver

algum problema que impossibilite a BS de rotear o tráfego, as SSs podem transmitir os dados, se tornando uma opção de roteamento de tráfego para a célula.

O padrão IEEE 802.16 especifica três importantes termos para topologia *Mesh:* vizinho (*neighbor*), vizinhança (*neighborhood*) e vizinhança estendida (*extended neighborhood*). As estações com as quais o nó possui enlaces diretos recebem a denominação de vizinhos. Vizinhos de um nó formam a vizinhança e são aqueles distantes de um salto do nó. A vizinhança estendida contém, adicionalmente, todos os vizinhos da vizinhança [1].

2.5. Modelo de Referência

O padrão IEEE 802.16 define um modelo de referência que é empregado tanto na BS quanto na SS e define as duas camadas especificadas pelo padrão IEEE 802.16: camada de acesso ao meio (MAC) e a camada física. A camada MAC é subdivida em três subcamadas: a Subcamada de Convergência Específica, a Subcamada da Parte Comum da MAC e a Subcamada de Segurança [1]. O modelo de referência do padrão IEEE 802.16 é ilustrado na Figura 2.3.

Figura 2.3: Modelo de Referência do Padrão IEEE 802.16.

2.5.1. Camada MAC

A camada MAC tem a função de gerenciar as conexões, garantir Qualidade de Serviço através de mecanismos de alocação dinâmica de recursos e atribuição de prioridades de tráfego. A camada MAC também é responsável pela multiplexação dos fluxos de tráfego em conexões, escalonamento, suporte a segurança da comunicação e suporte à topologia da rede [1]. A camada MAC do padrão IEEE 802.16 basicamente provê inteligência à camada física. A camada MAC também é responsável pelo controle de acesso ao meio e pela alocação de banda. A BS concede ou aloca largura de banda através de um dos seguintes mecanismos [1] [4] [9]:

- GPSS (*Grant Per Subscriber Station*): neste mecanismo a BS concede largura de banda por SS, e é responsabilidade da SS redistribuir a largura de banda entre as suas conexões mantendo a QoS de acordo com o nível de serviço negociado. Esse mecanismo é utilizado em cenários onde existem muitas conexões por terminal, o que possibilita ajustes mais sofisticados de acordo com as necessidades de QoS nas aplicações.
- GPC (*Grant Per Connection*): a BS concede largura de banda por conexão. Assim, este mecanismo é recomendado para cenários onde existem poucos usuários por SS.

Com a intenção de assegurar que as requisições das SSs sejam atendidas, a BS aloca largura de banda com o propósito específico de garantir essas requisições antes que as SSs possam efetuar suas requisições. Esse procedimento é designado de *polling*. Há duas formas de *polling* na BS [1] [5]:

- *Unicast*: Cada SS é interrogada individualmente pela BS para informar se deseja utilizar o meio para transmissão. Caso queira transmitir, a BS aloca largura de banda para o envio de mensagens de requisição.

- Baseado em Contenção: A técnica de requisição de largura de banda baseada em contenção é empregada quando uma quantidade insuficiente de largura de banda é disponibilizada individualmente pelo envio de mensagens da BS para muitas SSs inativas. A alocação é feita por *multicast* ou *broadcast* para um grupo de SSs que devem disputar por uma oportunidade de enviar suas requisições de largura de banda.

2.5.1.1. Formato da MAC PDU

A MAC PDU (*Protocol data Unit*) também conhecida como *frame* da MAC, é responsável pela troca de dados entre as camadas MAC da BS e da SS. A MAC PDU, cujo tamanho máximo é de 2048 bytes, é composta por um cabeçalho de tamanho fixo, um *payload* de tamanho variável e um CRC (*Cyclic Redundancy Check*) [1] [4] [6]. O *payload* e CRC são opcionais. A Figura 2.4 ilustra o formato geral da MAC PDU, as siglas MSB e LSB referem-se ao bit mais significativo (*Most Significant Bit)* e menos significativo (*Least Significant Bit*), respectivamente.

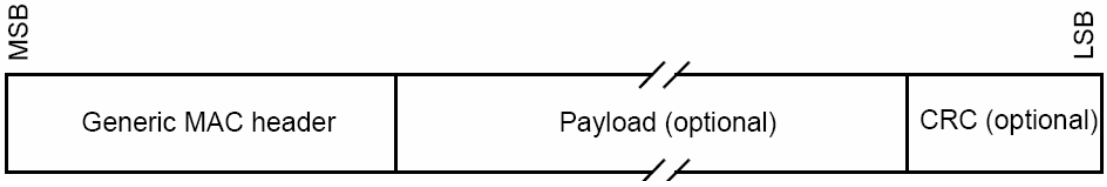

Figura 2.4: Formato do MAC PDU [1].

O padrão IEEE 802.16 especifica dois formatos de cabeçalhos: o cabeçalho genérico e cabeçalho de requisição de largura de banda. Os dois formatos são distinguidos pelo campo HT (*Header Type*), se o campo HT = 0 identifica que o formato do cabeçalho é genérico, caso o campo HT =1 indica o cabeçalho de requisição de largura de banda.

O cabeçalho genérico do padrão IEEE 802.16 é ilustrado na Figura 2.5. O cabeçalho genérico possui vários campos, a seguir serão apresentados os significados de cada um destes campos [1] [6]:

- EC (*Encriptation Control*) de 1 bit indica se o *payload* será criptografado.
- *Type* informa que tipo de carga contém o *payload*. Se *Type* = 0, indica que o *payload* é composto por um sub-cabeçalho de gerenciamento de concessão, utilizado para transportar o pedido de largura de banda à BS. As SSs informam à BS suas necessidades de gerenciamento da largura de banda no sentido *uplink*. Assim, evita-se a transmissão de um quadro completo para solicitar largura de banda, trata-se de uma requisição do tipo *piggyback*, em que um quadro de dados é aproveitado para fazer a requisição. Se *Type* = 1, implica que o *payload* contém um sub-cabeçalho de empacotamento que tem a função de empacotar várias MAC SDUs em uma única MAC PDU. Se *Type* = 2, informa que existe um sub-cabeçalho de fragmentação, utilizado para controlar o processo de fragmentação de MAC SDUs no *payload*. Assim, a MAC SDU pode ser transmitida e fragmentada independentemente. A fragmentação pode ocorrer tanto na BS, bem como na SS. Se *Type* = 3, compreende uma expansão do sub-cabeçalho de fragmentação ou de empacotamento no *payload*. Se *Type* = 4, indica que o *payload* detém informações referentes à retransmissão de quadros (ARQ – *Automatic Repeat Request*). E, finalmente, se *Type* = 5, haverá um subcabeçalho *Mesh* no *payload*.
- RSV (*Reserved*) este campo é reservado para uso futuro, contém 1 bit, no cabeçalho genérico contém dois campos RSV, como pode-se observar na Figura 2.5.
- CI (CRC *Indicator*) informa se há (CI = 1) ou não (CI = 0) um código CRC no final da MAC PDU.
- EKS (*Encriptation Key Sequence*) de 2 bits indica qual chave foi utilizada na criptografia. Sua ausência é denotada pelo campo EC = 0.

- LEN (*Lenght*) informa o tamanho total da MAC PDU, ou seja, comprimento do quadro, incluindo CRC.
- CID (*Connection Identifier*) (16 bits) é o identificador único de cada conexão atribuído pela BS.
- HCS (*Header Check Sequence*) de 8 bits para detectar erros presentes no cabeçalho.

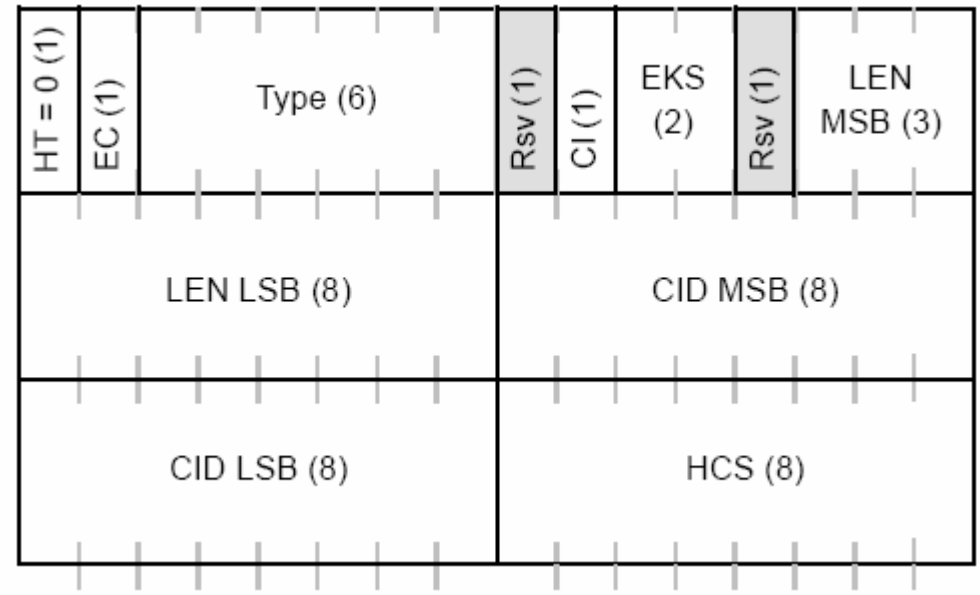

Figura 2.5: Formato do Cabeçalho Genérico [1].

As MAC PDUs que utilizam o cabeçalho de requisição de banda não contêm *payload* e são utilizadas exclusivamente para requisitar largura de banda *uplink* para uma determinada conexão. O cabeçalho de requisição de banda é representado na Figura 2.6. Observa-se que o campo HT = 1 indicando que é um cabeçalho de requisição de banda e o campo EC = 0 devido não se ter o *payload*. A seguir serão detalhados os campos do cabeçalho de requisição de banda [1] [6]:

- *Type* de 3 bits informa o tipo de requisição de banda. Se *Type* = 0, implica que a requisição de banda será incremental. Se *Type* = 1, a requisição será agregada.

- BR (*Bandwidth Request*) (19 bits) expressa a quantidade de largura de banda requerida pela SS (no sentido *uplink*) para transmitir um número específico de *bytes*.
- CID (*Connection Identifier*) (16 bits) identifica a que conexão pertence a MAC PDU.
- HCS (*Header Check Sequence*) (8 bits) detecta erros no cabeçalho.

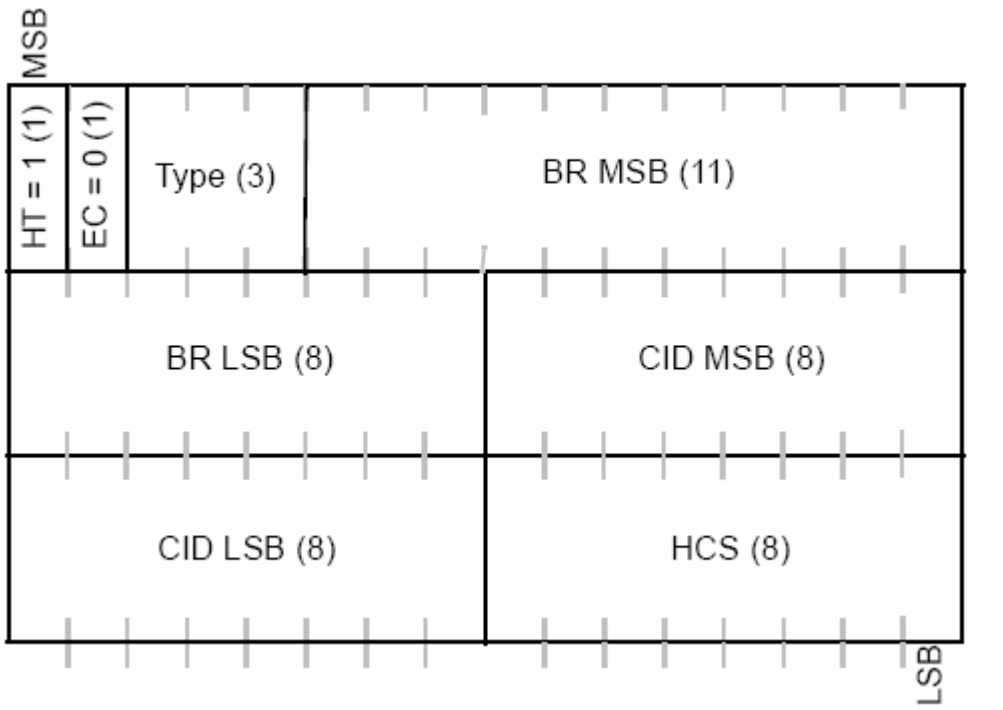

Figura 2.6: Formato do Cabeçalho de Requisição de Largura de Banda [1].

2.5.1.2. Subcamadas da MAC

A camada MAC é dividida em três subcamadas: a Subcamada de Convergência Específica, a Subcamada da Parte Comum da MAC e a Subcamada de Segurança. Nas próximas subseções serão apresentadas as especificações para cada uma das subcamadas MAC definidas pelo padrão IEEE 802.16.

2.5.1.2.1. Subcamada de Convergência Específica

A subcamada de Convergência (CS) é a subcamada superior na camada MAC, fazendo interface com a camada acima. A CS é responsável pelo mapeamento do tráfego proveniente das camadas superiores para a camada MAC [1] [5].

A subcamada de convergência classifica os dados que recebe da camada superior em fluxos de serviços e conexões, associando a eles um SFID (identificador de fluxo de serviço) e um CID (identificador de conexão), assim é possível fornecer QoS mais adequada de acordo com a necessidade de cada fluxo de serviço. Também é dever da subcamada de convergência remover informações redundantes do cabeçalho dos pacotes, tais pacotes são chamados SDUs (*Service Data Units*). Esta técnica é chamada de PHS (*Packet Header Supression*) [1] [6].

A CS necessita de múltiplas especificações para prover interface com diferentes protocolos da camada superior. Por este motivo, o padrão IEEE 802.16 apresenta duas especificações para esta subcamada [1]: A Convergência de Pacotes e a Convergência ATM (*Asynchronous Transfer Mode*).

2.5.1.2.1.1. Convergência de Pacotes

A Convergência de Pacote é definida para serviços de pacotes, tais como IPv4, IPv6, *Ethernet* e redes locais virtuais (VLAN) [5]. A Convergência de Pacotes é responsável pelo mapeamento dos pacotes das camadas superiores para um fluxo de serviço especificado pelo padrão IEEE 802.16 [1] [6]. Em outras palavras, a Convergência de Pacotes é responsável por classificar as PDUs do protocolo da camada superior na conexão apropriada. O processo de mapeamento relaciona uma MAC SDU com uma conexão, cria uma associação com as características do fluxo de serviço desta conexão. Desta forma, facilita a entrega da MAC SDU com as características apropriadas de QoS. Portanto, as PDUs recebidas da camada superior são encapsuladas no formato da MAC SDU.

2.5.1.2.1.2. Convergência ATM

A Convergência ATM é definida para serviços ATM. Nas redes ATM existem dois modos de comutação: a comutação por caminho virtual (VP – *Virtual Path*) e a comutação por canal virtual (VC – *Virtual Channel*). No modo de comutação VP, a conexão é identificada por uma VPI (VP *Identifiers*), enquanto que no modo de comutação VC, a conexão é identificada por um par de VPI e VCI (VC *Identifiers*) [1] [5].

Na Convergência ATM, diferentes mecanismos são aplicados às células ATM baseados em sua comutação. A Convergência ATM realiza um mapeamento, e este mapeamento é realizado durante a fase de estabelecimento da conexão. Depois da conexão estabelecida, as células ATM são mapeadas para o fluxo de serviço especificado pelo padrão IEEE 802.16 baseado nos seus valores VPI ou VPI e VCI dependendo do mecanismo de comutação utilizado.

2.5.1.2.2. Subcamada da Parte Comum da MAC

A Subcamada de Parte Comum da MAC é a principal subcamada da camada MAC do padrão IEEE 802.16. Dentre as principais funções desempenhadas, podem ser citadas [1]: o escalonamento e alocação dinâmica de recursos de transmissão, estabelecimento e manutenção das conexões, construção das MAC PDU, suporte a camada física, suporte a Qualidade de Serviço.

No padrão IEEE 802.16 as conexões são identificadas por identificadores de 16 bits denominado CID. Assim, podem existir no máximo 64000 conexões dentro de cada canal *uplink* e *downlink*. Na topologia PMP, durante o processo de inicialização de uma SS, três pares de conexões de gerência (*uplink* e *downlink*) devem ser estabelecidos entre as SS e BS [1] [4]:

- Conexão básica: é usada para enviar pequenas mensagens de gerência urgentes entre as SSs e a BS.
- Conexão primária: é usada para enviar mensagens de gerência não tão urgentes e maiores, que toleram atrasos maiores.
- Conexão secundária: é o terceiro tipo de conexão, e pode ser utilizada opcionalmente, sendo que foi desenvolvida para facilitar a gerência das SSs. A conexão secundária de gerência é usada para enviar mensagens de outros protocolos padronizados tolerantes ao atraso, tais como DHCP (*Dynamic Host Configuration Protocol*) e SNMP (*Simple Network Management Protocol*).

Além dessas três conexões de gerenciamento, também são alocadas conexões de transporte às SSs para os serviços contratados. As conexões de transporte são unidirecionais a fim de facilitar a distinção dos parâmetros de QoS nos tráfegos *uplink* e *downlink* [1].

2.5.1.2.3. Subcamada de Segurança

Segurança é um aspecto fundamental em redes de computadores, principalmente na área de redes sem fio, porque os dispositivos não estão fisicamente conectados e compartilham serviços e dados através do ar. Desta forma, são mais vulneráveis aos acessos indevidos dos dados que trafegam. Assim, as redes sem fio necessitam de mecanismos que garantam maior segurança [8]. Tendo em vista atender esse requisito, o padrão IEEE 802.16 especifica uma subcamada de segurança, localizada abaixo da subcamada de Parte Comum da MAC, definindo mecanismos de autenticação, protocolos, certificados e criptografia como forma de aumentar a segurança das redes metropolitanas sem fio [1].

Ela fornece privacidade às SSs, através da encriptação das conexões estabelecidas. Além disso, a subcamada protege a BS contra o acesso não autorizado a seus serviços, através

de um protocolo de administração de chaves, de métodos de autenticação baseados em certificados digitais, e de criptografia. Mesmo assim ainda há vulnerabilidades, devido principalmente ao enlace de radiofreqüência. Neste caso, a informação pode ser interceptada e modificada mais facilmente do que a informação que trafega por uma rede cabeada. Todavia, diversas vulnerabilidades existentes no IEEE 802.11 original foram eliminadas no IEEE 802.16.

A subcamada de Segurança emprega ainda um protocolo de gerenciamento de chave cliente/servidor autenticada, onde a BS é responsável por controlar a distribuição de chaves às SSs. Contudo, a autenticação das estações assinantes é baseada em certificados digitais adicionados ao seu protocolo de gerenciamento de chave (PKM – *Privacy Key Management*) [1]. O certificado digital contém a chave pública da SS e o seu endereço MAC.

2.5.1.2.3.1. Associações de Segurança

Associações de segurança (SA - *Security Association*) são informações de controle compartilhadas entre uma BS e uma ou várias SSs com a finalidade de proteger as conexões. O padrão IEEE 802.16 especifica dois tipos de SA [1] [8]: associações de segurança de autorização e associações de segurança de dados.

A função da SA de autorização é promover a autenticação dos dispositivos e com isso aumentar a segurança dos usuários da rede.

As associações de segurança de dados consistem de uma seqüência de informações trocadas com a finalidade de prover confidencialidade dos dados trafegados durante a conexão. O padrão define três tipos de SA de dados: primária, estática e dinâmica. A SA de dados é do tipo primária se ela é estabelecida durante a inicialização do enlace pelas estações assinantes. A SA é do tipo estática quando configurada diretamente na estação base. A SA é do tipo dinâmica se é estabelecida e eliminada de acordo com a demanda das conexões de transporte [1]. As associações de segurança estáticas e dinâmicas podem ser compartilhadas

por múltiplas estações assinantes, pois são estabelecidas pela BS e o padrão IEEE 802.16 tem suporte *multicast* permitindo que muitos CIDs compartilhem uma mesma associação de segurança.

O padrão define que cada SA de dados é identificada através de um SAID (*Security Association Identifier*), o que estabelece sua unicidade [1]. A estação base assegura a confidencialidade desses dados, desta forma, cada cliente tem acesso apenas a suas associações de segurança.

2.5.2. Camada Física

As principais funções da camada física são: transmissão das MAC PDUs, definição das técnicas de transmissão digital: modulação e codificação, definição de espectro, correção de erro, definição da técnica de *duplexing*, construção dos *frames* e *subframes* de transmissão. A camada física opera na faixa de freqüências de 10-66 GHz e 2-11 GHz (IEEE 802.16a) com taxas de transmissão teóricas entre 32 e 130 Mbps, dependendo do esquema de codificação e modulação utilizado [1].

Várias técnicas de modulação digital podem ser utilizadas em sistemas de telecomunicações, devido à intensidade do sinal e a relação sinal/ruído (SNR) diminuírem em função da distância relativa à BS. Assim, a camada física do padrão IEEE 802.16 suporta quatro modulações diferentes [6]:

- *Binary Phase Shift Keying* (BPSK): A modulação BPSK é uma modulação digital binária, ou seja, a modulação codifica um bit por símbolo.
- *Quadrature Phase Shift Keying* (QPSK): Diferentemente da BPSK a modulação QPSK codifica dois bits por símbolo.
- *Quadrature Amplitude Modulation* (QAM) 16-QAM : A modulação 16-QAM codifica quatro bits por símbolo.

- *Quadrature Amplitude Modulation* (QAM) 64-QAM : A modulação 64-QAM codifica seis bits por símbolo.

A camada física opera em um formato de *frames*, os quais são subdivididos em intervalos de tempo chamados *slots* físicos. Cada *frame* é divido em *subframe downlink* e *subframe uplink* (Figura 2.7). O *subframe downlink* é utilizado pela BS para a transmissão de dados e de informações de controle para as SSs. O *subframe uplink* é compartilhado entre todas as SSs para transmissões que têm como destino a BS [9]. Em outras palavras, a comunicação entre a BS e a SS ocorre em dois sentidos, da BS para SS e da SS para BS, respectivamente, *uplink* e *downlink* [1] [9].

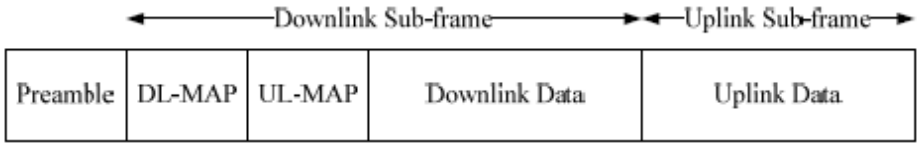

Figura 2.7: Frame do padrão IEEE 802.16 [1].

O padrão IEEE 802.16 permite dois modos de acesso ao meio físico: duplexação por divisão de freqüência (FDD – *Frequeny Division Duplexing*) e duplexação por divisão do tempo (TDD – *Time Division Duplexing*) [1] [4]. No modo FDD os canais *downlink* e *uplink* operam simultaneamente em freqüências diferentes. No modo TDD os *subframes uplink* e *downlink* compartilham a mesma freqüência, porém, não é possível realizar transmissões simultâneas nos dois sentidos. Assim, cada *frame* TDD tem um *subframe downlink* seguido por um *subframe uplink.*

2.5.2.1. Interfaces Aéreas

O padrão IEEE 802.16 especifica cinco interfaces aéreas para a camada física: *WirelessMAN*-SC, *WirelessMAN*-SCa, *WirelessMAN*-OFDM, *WirelessMAN*-OFDMA e *WirelessMAN*-HUMAN [1] [17]:

- *WirelessMAN*-SC : é baseada em uma portadora única (SC – *Single Carrier*), opera na faixa de freqüência de 10 – 66 GHz, projetada para suportar somente a topologia PMP. A *WirelessMAN*-SC provê suporte para TDD e FDD. O canal *uplink* é baseado em uma combinação do TDMA e DAMA (*Demand Assigned Multiple Acess*). O canal *downlink* é baseado no TDM (*Time Divison Multiplexing*).

- *WirelessMAN*-SCa: utiliza uma portadora única como a interface área *WirelessMAN*-SC, opera na faixa de freqüência de 2 – 11 GHz, sendo projetada para sistemas NLoS. O canal *uplink* é baseado no TDMA e o *downlink* em TDM ou TDMA. Acrescenta melhorias na estrutura dos quadros visando contornar as condições do meio de transmissão sem linha de visada direta, incluindo o esquema de codificação FEC (*Forward Error Correction*).
- *WirelessMAN*-OFDM: utiliza a modulação OFDM (*Ortoghonal Frequency Division Multiplexing*), projetada para sistema sem visada direta, com transformada de 256 sub-portadoras. A OFDM é uma técnica de modulação multiportadora que tem por idéia básica dividir os dados a serem transmitidos em diversos canais e transmití-los paralelamente, a taxas menores. Esta técnica é muito usada em sistemas modernos de telecomunicações e dentre as suas vantagens pode-se citar: facilidade para transmissão em caminhos múltiplos, maior resistência à interferência, ideal para condições NLoS. O uso desta interface aérea é obrigatório para bandas de freqüências não licenciadas. A sua especificação é definida tanto para o padrão IEEE 802.16 como para o HIPERMAN do ETSI, o que assegura a interoperabilidade global do padrão IEEE 802.16.
- *WirelessMAN*-OFDMA: Utiliza a modulação OFDM como a interface área *WirelessMAN*-OFDM, porém com um número maior de sub-portadoras, 2048 sub-portadoras. Portanto, a utilização de 2048 sub-portadoras torna a FFT (*Fast Fourier*

Transform) mais lenta e aumenta os requisitos de sincronização. Nesse sistema, o acesso múltiplo é oferecido através de um subconjunto de endereçamento de múltiplas portadoras para receptores individuais. Assim, as SSs podem utilizar mais de uma sub-portadora.

- *WirelessMAN*-HUMAN: opera nas faixas de freqüências não licenciadas 5 – 6 GHZ, diferencia da *WirelessMAN*-OFDM, por utilizar um esquema de seleção de freqüência dinâmico (DFS – *Dynamic Frequency Selection)* para detectar e evitar interferências.

2.5.3. Aquisição e Inicialização de um Canal

Uma estação assinante que deseja ter acesso a uma rede de comunicação deve passar por um processo de inicialização com a BS, o qual é ilustrado pela máquina de estados representada na Figura 2.8 [5] [7] [33]. Primeiramente, como pode ser observado no lado direito da Figura 2.8, a SS varre o espectro de freqüência *downlink*. Após decidir sobre que canal (ou pares de canais) irá se configurar uma conexão, a SS tenta sincronizar a transmissão *downlink* com a BS para detectar o preâmbulo do *frame*. Pelo preâmbulo a camada MAC procura por um DCD (*Downlink Channel Descriptor*) e um UCD (*Uplink Channel Descriptor*). Assim que a camada física é sincronizada e a SS obtém os parâmetros sobre o meio físico, a estação cliente pode iniciar o processo de *ranging*. Este processo testa o canal para definir correções de tempo e potência da transmissão. A SS deverá enviar uma rajada usando uma potência mínima e deverá tentar incrementar essa potência de transmissão se não receber a variação de resposta. Para finalizar a primeira etapa é iniciada a negociação da capacidade do canal, onde a SS a partir do DCD e UCD da mensagem *Capabilities Request Message* obtém informações sobre os seguintes parâmetros UL (*Uplink*) e DL (*Downlink*): tipos de modulação, esquemas de codificação, taxas suportadas e tipo de duplexagem. A BS aceita ou nega a admissão de uma SS com base nas suas capacidades [7].

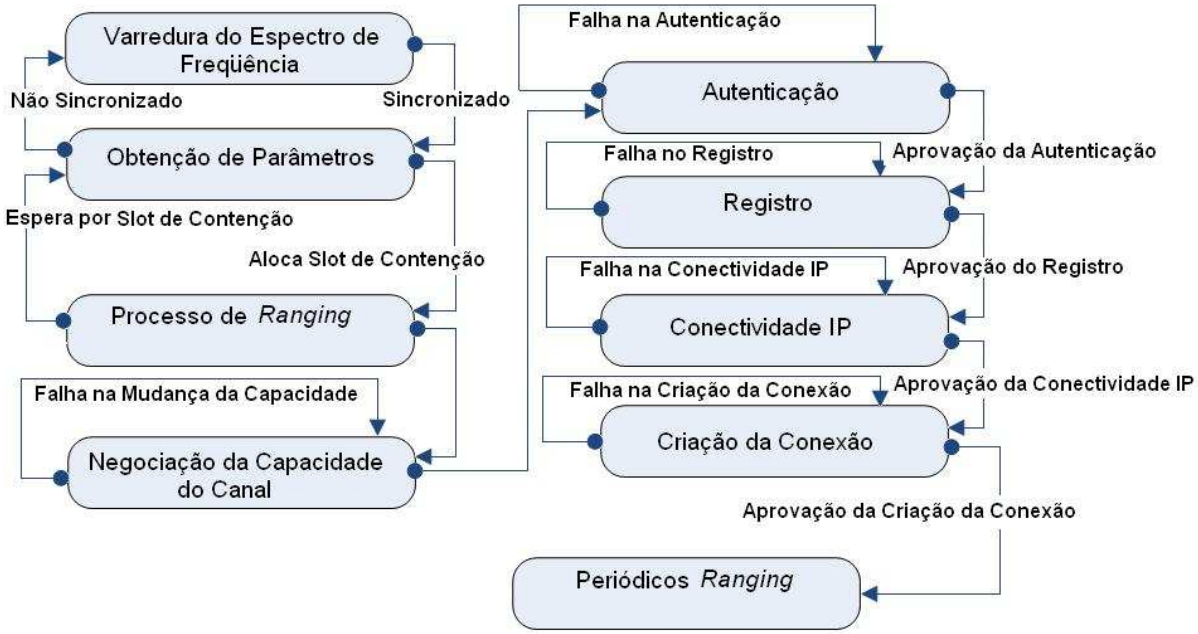

Figura 2.8: Processo de Entrada na Rede.

Na segunda etapa, representada pelo lado esquerdo da Figura 2.8, a BS conduz um processo de autorização da SS para que ela possa entrar na rede, o que inclui a troca de chaves de segurança, certificação e suporte à criptografia entre a BS e a SS [33]. Após o processo de autenticação, a SS envia uma mensagem de requisição de registro da conexão para a BS e a BS envia uma resposta à solicitação de registro da SS. Em seguida, a SS inicializa o DHCP para obter um endereço IP e outros parâmetros os quais viabilizam o estabelecimento de conectividade IP com a BS. A SS usa o protocolo TFTP (*Trivial File Transfer Protocol*) para obter os parâmetros operacionais. A BS envia informação adicional de configuração para a SS. Finalmente, as conexões são criadas após a finalização das etapas de registro e de transferência de parâmetros operacionais [5] [7].

2.6. Considerações Finais

Neste capítulo apresentou-se o padrão IEEE 802.16 que é uma das tecnologias mais promissoras para acesso banda larga sem fio (BWA). Além disto, descreveu-se, seguindo uma

ordem cronológica, algumas das recomendações do padrão IEEE 802.16. Os dois tipos de topologias especificadas para o padrão IEEE 802.16: Topologia PMP e *Mesh* foram abordados.

Alguns detalhes da camada MAC e da camada física especificados pelo padrão IEEE 802.16 foram descritos. As três subcamadas da camada MAC: a Subcamada de Convergência Específica, a Subcamada da Parte Comum da MAC e a Subcamada de Segurança foram enfatizadas. Apresentou-se de maneira sucinta as cinco interfaces aéreas especificadas para a camada física do padrão IEEE 802.16.

Capítulo 3

PROVISÃO DE QUALIDADE DE SERVIÇO (QoS) NO PADRÃO IEEE 802.16

3.1. Introdução

O avanço das redes de comunicação (*LAN's, WAN's, WLAN's, WMAN's, etc*) proporcionou um considerável aumento no número de usuários e aplicações. Junto a este crescimento veio a necessidade de satisfazer as exigências dos usuários. Com a intenção de prover garantias ao usuário surgiu o conceito de Qualidade de Serviço (*QoS – Quality of Service*) para rede de computadores.

Nas redes sem fio, os parâmetros de QoS são mais difíceis de serem mantidos do que nas redes cabeadas. Tal complexidade se deve a alguns fatores, como qualidade de transmissão do meio sem fio, recursos escassos de largura de banda, mobilidade das estações, etc. Diante deste cenário, o padrão IEEE 802.16 foi desenvolvido com QoS em mente, surgindo como uma solução para prover QoS nas redes de acesso banda larga sem fio.

Neste capítulo também será apresentado uma fundamentação teórica sobre as disciplinas de escalonamento. Devido ao fato do padrão IEEE 802.16 não especificar qual disciplina de escalonamento deve ser utilizada, a escolha de um mecanismo de escalonamento eficiente é essencial para garantir que os requisitos de QoS sejam atendidos e tem grande impacto no desempenho da rede.

O texto deste capítulo será organizado da seguinte maneira: a Seção 3.2 faz uma abordagem geral sobre QoS. A Seção 3.3 apresenta os desafios da provisão de QoS nas redes sem fio. A Seção 3.4 aborda a Qualidade de Serviço no padrão IEEE 802.16. Em seguida, a Seção 3.5 apresenta alguns conceitos referentes às disciplinas de escalonamento que são fundamentais na provisão de QoS nas redes de acesso IEEE 802.16. Finalmente, a Seção 3.6 apresenta as considerações finais deste capítulo.

3.2. Qualidade de Serviço (QoS)

Qualidade de Serviço é um conceito que expressa a qualidade de transmissão em redes de comunicação. Na definição da ISO [11], "QoS é o efeito coletivo do desempenho de um serviço, o qual determina o grau de satisfação de um usuário do serviço".

Qualidade de Serviço não faz mágica, não cria largura de banda inexistente, não corrige imperfeições físicas das redes e situações drásticas de congestionamentos causadas por projetos mal elaborados. Entretanto, fornecer garantias de QoS em uma rede é de grande importância para o sucesso de aplicações em tempo real, como videoconferência, VoIP (Voz sobre IP), etc. Estas aplicações demandam, além de grande largura de banda, um serviço diferenciado. Em algumas aplicações é preciso garantir que a transmissão de dados seja feita sem interrupção ou perda de pacotes [12].

QoS é caracterizada por um conjunto de parâmetros que traduz as expectativas dos usuários e cujos valores são estabelecidos nos contratos de nível de serviço ou SLAs (*Service Level Agreement*). Assim, a rede deve utilizar mecanismos para gerenciar seus recursos para que possa prover o nível de Qualidade de Serviço que o usuário deseja. Tais parâmetros são normalmente relacionados à capacidade de transmissão de dados, ao tempo consumido nas transmissões e à confiabilidade. Alguns parâmetros comumente empregados são descritos a seguir [12] [13]:

- Taxa de transmissão: quantidade de dados que podem ser transmitidos por unidade de tempo, normalmente expressa em bits por segundo (bits/s) ou em múltiplos dessa unidade (Kbps, Mbps);
- Vazão: o número de *bytes* de dados transmitidos com sucesso durante um determinado período de tempo, sendo medido separadamente em cada direção dos dados, também expresso em bits/s;
- Atraso: representa o tempo desde o envio da mensagem pelo usuário de origem até seu recebimento pelo usuário de destino. Em outras palavras, o atraso é o intervalo de tempo entre o envio e o recebimento de uma mensagem;
- Variação do Atraso (*jitter*): variação do atraso entre as unidades de dados consecutivas;
- Taxas de perdas de pacotes: é razão entre a quantidade de pacotes perdidos e a quantidade de pacotes enviados. Os pacotes podem ser perdidos na rede por descarte nas filas dos nós intermediários, ou podem ser corrompidos por colisão com outros pacotes em enlaces compartilhados e ainda por ruídos eletromagnéticos no meio físico.

3.3. Qualidade de Serviço em Redes Sem Fio

As redes de acesso sem fio e móveis revolucionaram a telefonia e também estão causando impacto cada vez mais profundo no mundo das redes de computadores. O avanço das redes sem fio irá proporcionar o acesso à infra-estrutura global a qualquer hora, desimpedido e em qualquer lugar, habilitando um novo conjunto muito interessante de serviços [12]. Porém, a provisão de QoS em redes de acesso sem fio impõe muitos desafios, devido à dinâmica do ambiente em função da mobilidade dos usuários, as interferências

externas provocam variações na capacidade do canal e na taxa de erros, e redução da força do sinal à medida que aumenta a distância entre emissor e receptor [13].

Entretanto, o ambiente sem fio requer alguns cuidados para garantir desempenho, segurança e disponibilidade: obstáculos reduzem a área de cobertura, vários usuários na mesma localidade influenciam no desempenho, e a proximidade com fontes de interferência podem inviabilizar a transmissão e recepção de sinais. Além disso, será necessário utilizar recursos de autenticação, criptografia e controle de endereço MAC para permitir a utilização da tecnologia com maior segurança.

3.4. Qualidade de Serviço no padrão IEEE 802.16

O projeto do padrão IEEE 802.16 foi desenvolvido com QoS como pauta, e para conseguir tal objetivo criou-se um padrão orientado a conexão, onde as diferentes aplicações são diferenciadas em múltiplas classes de serviços, de acordo, com os parâmetros solicitados por cada aplicação. Nesta Seção serão apresentadas algumas das características especificadas na documentação do IEEE 802.16 [1] referente à provisão de Qualidade de Serviço.

3.4.1. Arquitetura de Qualidade de Serviço do padrão IEEE 802.16

O padrão IEEE 802.16 foi projetado para suporte a QoS, ou seja, com a habilidade de suportar diferentes níveis de serviços para tipos distintos de tráfego, incorporada naturalmente na camada MAC. Porém, o padrão definiu apenas uma arquitetura capaz de suportar QoS e não especifica uma solução completa para fornecer garantias ao serviço oferecido.

A arquitetura de QoS especificada pelo padrão IEEE 802.16 é ilustrada na Figura 3.1, especificando-se as classes de serviço, as mensagens de sinalização, etc, porém deixa em aberto os mecanismos de escalonamento e de Controle de Admissão de Conexões (CAC).

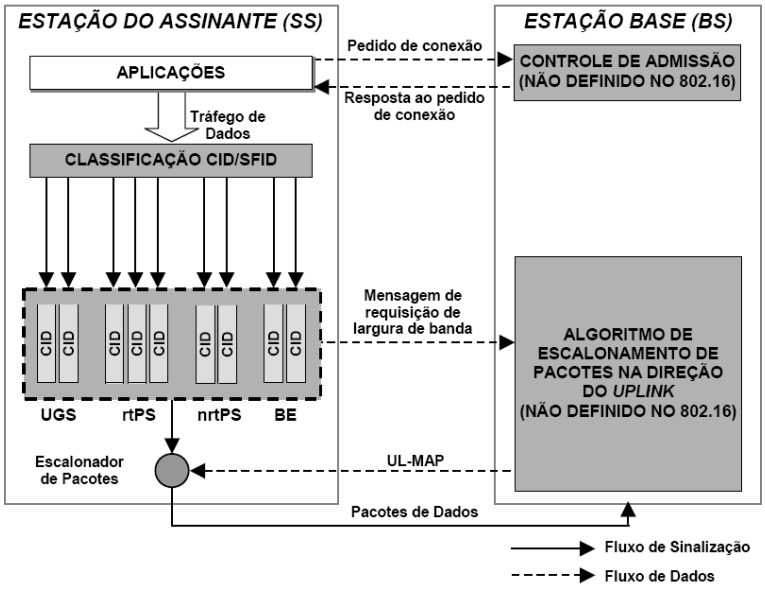

Figura 3.1: Arquitetura de QoS [5].

Uma disciplina de escalonamento eficiente é essencial para garantir que os requisitos de QoS sejam atendidos e tem grande influência no desempenho da rede. A Seção 3.5 apresentará alguns conceitos e propostas de disciplinas de escalonamento encontradas na literatura para o padrão IEEE 802.16.

O mecanismo de CAC restringe o número de usuários simultâneos presentes na rede de forma a evitar a saturação do enlace sem fio. Ele que decide se uma conexão é aceita ou rejeitada dependendo dos recursos já alocados na rede. No capítulo 4 será apresentada uma proposta de um mecanismo de CAC para as redes de acesso IEEE 802.16 e algumas propostas de CAC para o padrão IEEE 802.16 encontradas na literatura.

3.4.2. Teoria do Modelo de Objeto de Operação

A Figura 3.2 [1] [5] apresenta os principais objetos da arquitetura de provisão de QoS especificados pelo padrão IEEE 802.16. Cada objeto é representado por um retângulo que contém vários atributos. Os atributos sublinhados identificam de forma única os objetos ao qual pertencem. Os atributos opcionais são denotados por colchetes. O relacionamento entre o número de objetos é marcado ao final de cada linha de associação entre eles. Por exemplo, um

fluxo de serviço pode estar associado com *0* ou *N* (várias) PDUs, mas uma PDU é associada com exatamente um fluxo de serviço. O fluxo de serviço é o conceito central do protocolo MAC, sendo identificado unicamente através de um SFID (32 bits). Os fluxos de serviço podem estar tanto na direção *uplink* como *downlink*. Os fluxos de serviço ativos e admitidos são mapeados por um CID (16 bits) [1].

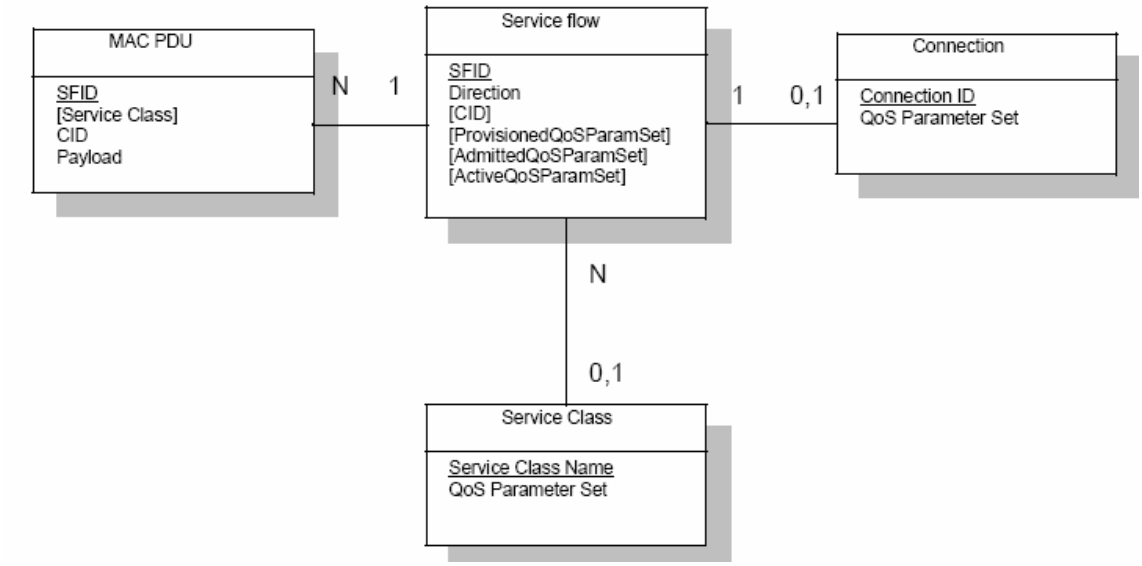

Figura 3.2: Teoria do Modelo de Objeto de Operação [1].

Os dados do usuário são classificados na subcamada de Convergência e submetidos ao MAC SAP **(***Service Access Point***)**. A informação entregue para a MAC SAP inclui o CID que identifica a conexão através da qual a informação é entregue. O fluxo de serviço para a conexão é mapeada para a conexão MAC identificada pelo CID [1].

A classe de serviço é um objeto opcional que pode ser implementada pela BS. Ela é referenciada por um nome ASCII, que é destinado para propósitos de aprovisionamento. A classe de serviço é definida na BS para possuir um conjunto de parâmetros de serviço particular (*QoS Parameter Sets*). O conjunto de parâmetros de QoS de um fluxo de serviço pode conter uma referência para o nome da classe de serviço (*Service Class Name*) como uma "macro"que seleciona todos os parâmetros de QoS da Classe de Serviço.

3.4.3. Classes de Serviço

O padrão IEEE 802.16 em seu projeto inicial especificou quatro classes de serviço, as quais devem ser tratadas de forma diferenciada pelo mecanismo de escalonamento da camada MAC. A seguir elas serão apresentadas [1] [6] [9]:

- UGS (*Unsolicited Grant Service*): suporta fluxos de serviço de tempo real que geram pacotes de dados com tamanho fixo periodicamente, tal como no tráfego CBR (*Constant Bit Rate*). O serviço oferece concessões de tamanho fixo periodicamente. Fluxos UGS não podem utilizar *slots* reservados para requisição de banda. Um fluxo UGS deve especificar os seguintes parâmetros de QoS: *Maximum Sustained Traffic Rate, Maximum Latency, Tolerated Jitter* e *Request/Transmission Policy*.
- rtPS (*Real-Time Polling Service*): projetada para o suporte aos fluxos de serviço de tempo real com pacotes de tamanho variável, gerados em intervalos periódicos, tais como tráfego multimídia no formato MPEG (*Motion Picture Experts Group*). O serviço oferece periodicamente oportunidades de requisição *unicast*, as quais devem satisfazer os requisitos de QoS do fluxo e permitem à SS especificar o tamanho da concessão desejada. Conexões rtPS não podem utilizar *slots* de contenção reservados para requisição de banda. Os parâmetros *Minimum Reserved Traffic Rate, Nominal Polling Interval* e *Tolerated Poll Jitter* são as principais especificações de QoS para esta classe de serviço. Para assegurar tais parâmetros, o escalonamento ideal é muito semelhante ao definido para os fluxos de serviço UGS.
- nrtPS (*Non-real-time Polling Service*): suporta tráfego não sensível ao atraso que requer concessões de largura de banda de tamanho variável regularmente, tais como as aplicações FTP (*File Transfer Protocol*), *e-mail*, SMS (*Short Message Service*), MMS (*Multimedia Messaging Service*), etc. O serviço é similar àquele oferecido pelo rtPS, porém o *polling unicast* ocorre com menor freqüência e o fluxo pode utilizar *slots* de

contenção reservados para requisição de banda. A oferta periódica de oportunidades de requisição *unicast* na classe nrtPS ocorre em intervalos de tempo mais espaçados do que na classe rtPS. Tal condição assegura que o fluxo seja recebido em oportunidades de requisição mesmo ocorrendo congestionamentos na rede. Um fluxo nrtPS deve informar os seguintes parâmetros de QoS: *Minimum Reserved Traffic Rate, Maximum Sustained Traffic Rate, Traffic Priority* e *Request/Transmission Policy*.

- BE (*Best Effort*): suporta tráfego de melhor esforço sem quaisquer garantias de QoS. A SS pode utilizar tanto *slots unicast* quanto *slots* de contenção para requisitar largura de banda. Os parâmetros *Maximum Sustained Traffic Rate*, *Traffic Priority* e *Request/Transmission Policy* são as principais especificações de QoS da classe BE.

O padrão IEEE 802.16e [15] incluiu uma nova classe de serviço, denominada ertPS (*extended real-time Polling Service)*. A classe ertPS é similar ao UGS, porém não há nenhum mecanismo de requisição de largura banda. Esta classe de serviço foi projetada para ser utilizada em fluxos de serviço de tempo real com pacotes de tamanho variável, como VoIP com supressão de silêncio. Os parâmetros de QoS dos fluxos de serviço ertPS são os mesmos da classe rtPS. Os parâmetros especificados pelo padrão IEEE 802.16 para as classes de serviço são apresentados na Tabela 3.1.

Tabela 3.1: Parâmetros Especificados pelo Padrão IEEE 802.16d [6].

Classe de Serviço	**Maximum sustained traffic rate**	**Minimum reserved traffic rate**	**Request/ transmission policy**	**Tolerated jitter**	**Maximum latency**	**Traffic priority**
UGS	x		x	x	x	
rtPS	x	x	x		x	
nrtPS	x	x	x			x
BE	x		x			x

3.4.4. Fluxos de Serviço

No padrão IEEE 802.16 toda a definição de parâmetros de QoS é feita com base no conceito de fluxo de serviço. Toda conexão em uma rede IEEE 802.16 possui um fluxo de serviço associado e especifica os parâmetros de QoS para tal conexão. Os fluxos de serviço são definidos como um fluxo unidirecional de pacotes para os quais é provido determinado nível QoS [1]. Em outras palavras, o fluxo de serviço é um serviço de transporte da camada MAC responsável pela transmissão unidirecional dos pacotes provenientes no sentido *uplink* e *downlink*.

Um fluxo de serviço é caracterizado por um conjunto de parâmetros de QoS, com o objetivo de padronizar a operação entre a SS e a BS. Esses parâmetros incluem detalhes de como a SS solicita *slots* no sentindo *uplink* e o comportamento esperado do escalonador implementado na BS [1] [2].

O padrão IEEE 802.16 define que os fluxos de serviços são parcialmente caracterizados pelos seguintes atributos [1] [5] [6] [8]:

- Identificador do Fluxo de Serviço (SFID): Um SFID é atribuído para todos os fluxos de serviço existentes. O SFID serve como o principal identificador na SS e na BS para o fluxo. Um fluxo de serviço tem no mínimo um SFID e uma direção associada.
- Identificador da Conexão (CID): O mapeamento para um SFID somente é realizado quando a conexão tem seu(s) fluxo(s) de serviço admitido(s).
- *ProvisionedQoSParamSet*: Um conjunto de parâmetros de QoS fornecido por mecanismos externos aos definidos no padrão IEEE 802.16, como, por exemplo, pelo sistema de gerenciamento de rede.
- *AdmittedQoSParamSet*: Define um conjunto de parâmetros de QoS para os quais a BS (e possivelmente a SS) reserva recursos. O principal recurso reservado é a largura de

banda, mas outros recursos também podem ser reservados (por exemplo, memória) para viabilizar a ativação do fluxo.

- *ActiveQoSParamSet*: Especifica um conjunto de parâmetros de QoS que define o serviço sendo atualmente provido para o fluxo de serviço. Somente um fluxo de serviço ativo pode encaminhar pacotes.
- *Authorization Module*: Uma função lógica dentro da BS que aprova ou rejeita cada mudança nos parâmetros de *QoS* e classificadores associados a um fluxo de serviço. Para tanto, define um "envelope" que limita os possíveis valores dos conjuntos de parâmetros *AdmittedQoSParamSet* e *ActiveQoSParamSet.*

O relacionamento entre os conjuntos de parâmetros de QoS é ilustrado na Figura 3.3 e na Figura 3.4. O *ActiveQoSParamSet* é sempre um subconjunto do A*dmittedQoSParamSet*, que é sempre um subconjunto do "envelope" de autorização. No modelo de autorização dinâmico (Figura 3.4), esse envelope é determinado pelo *Authorization Module* (rotulado como *AuthorizedQoSParamSet*). No modelo de autorização provisionado (Figura 3.3) esse envelope é determinado pelo *ProvisionedQoSParamSet* [1].

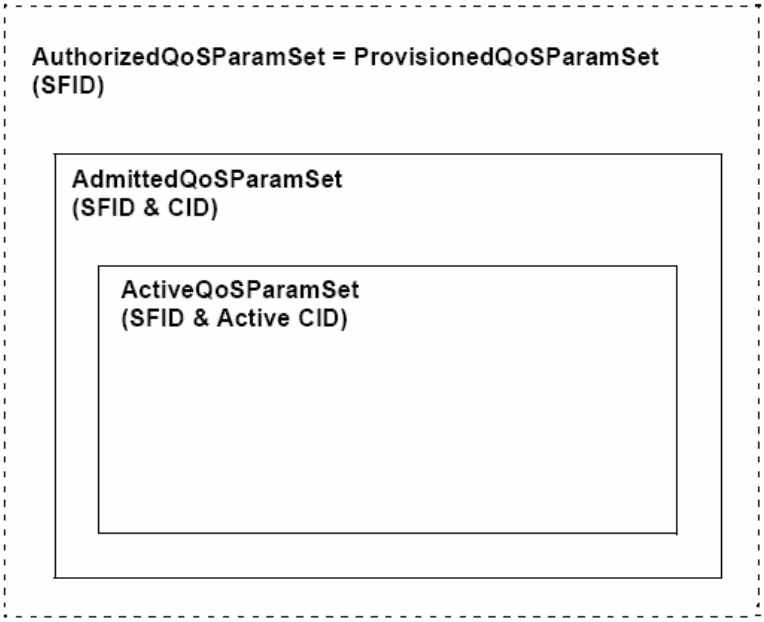

Figura 3.3: "Envelope" do Modelo de Autorização Provisionado [1].

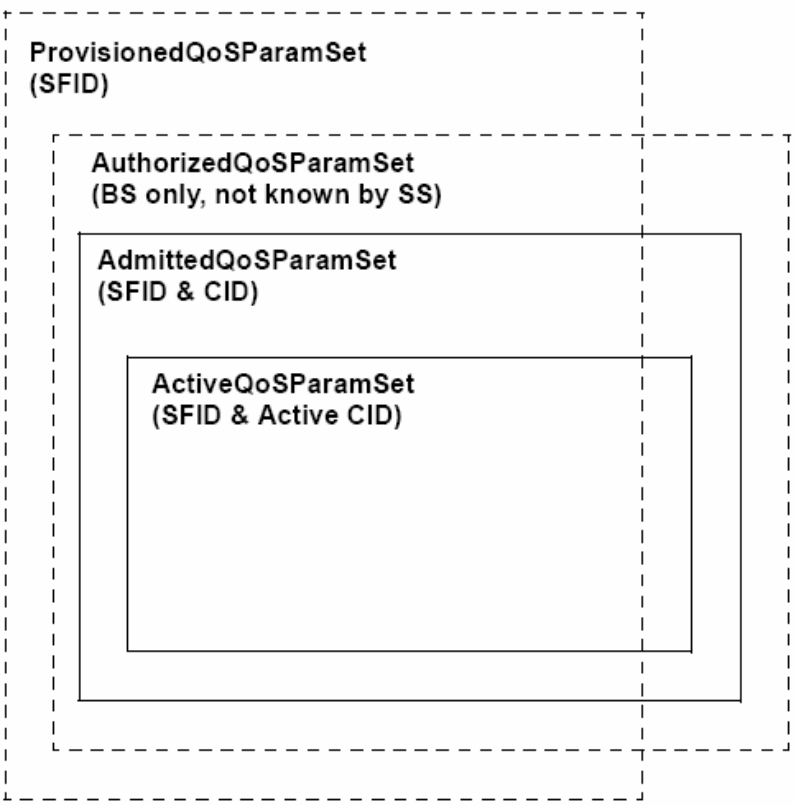

Figura 3.4: “Envelopes” do Modelo de Autorização Dinâmico [1].

É interessante pensar em três tipos de fluxos de serviço [1] [5] [8]:

- *Provisionado*: Conhecido pela provisão, por exemplo, do sistema de gerenciamento de rede. Os conjuntos de parâmetros *AdmittedQoSParamSet* e *ActiveQoSParamSet* para esse tipo de fluxo são ambos nulos.
- *Admitido*: Esse tipo de fluxo de serviço possui recursos reservados pela BS para o seu conjunto de parâmetros *AdmittedQoSParamSet*, mas esses parâmetros não estão ativos (o *ActiveQoSParamSet* é nulo). Os fluxos de serviço admitidos podem ter sido provisionados ou sinalizados por algum outro mecanismo.
- *Ativo*: Esse tipo de fluxo de serviço apresenta recursos comprometidos pela BS para o seu conjunto de parâmetros *ActiveQoSParamSet*. Por exemplo, a BS pode estar enviando mapas contendo concessões não solicitadas para a transmissão de um fluxo de serviço UGS. O conjunto de parâmetros *ActiveQoSParamSet* desse fluxo não é nulo.

3.4.5. Mecanismo de Classificação dos Fluxos de Serviço

O mecanismo de classificação dos fluxos é responsável pelo mapeamento dos pacotes a um fluxo de serviço. Assim, o mecanismo de classificação dos fluxos de serviço desempenha um papel fundamental na provisão de QoS no padrão IEEE 802.16. As Figuras 3.5 e 3.6 ilustram o mecanismo de classificação no sentido *uplink* e *dowlink*, respectivamente.

Depois que a aplicação efetua seu registro na rede, ocorrerá uma associação da aplicação a um fluxo de serviço através da atribuição de um identificador único ou um SFID. Cada pacote é rotulado com a atribuição de um SFID, de modo que a rede possa prover a QoS adequada. Ao enviar pacotes, as aplicações solicitam o estabelecimento de uma conexão com a rede e recebe um CID. A classificação das MAC SDUs e a atribuição de SFIDs e CIDs são realizadas na Subcamada de Convergência Específica da MAC. Portanto, os pacotes da rede de acesso IEEE 802.16 incluem dois identificadores, por fluxo e por conexão, o que torna camada MAC do padrão IEEE 802.16 orientada à conexão.

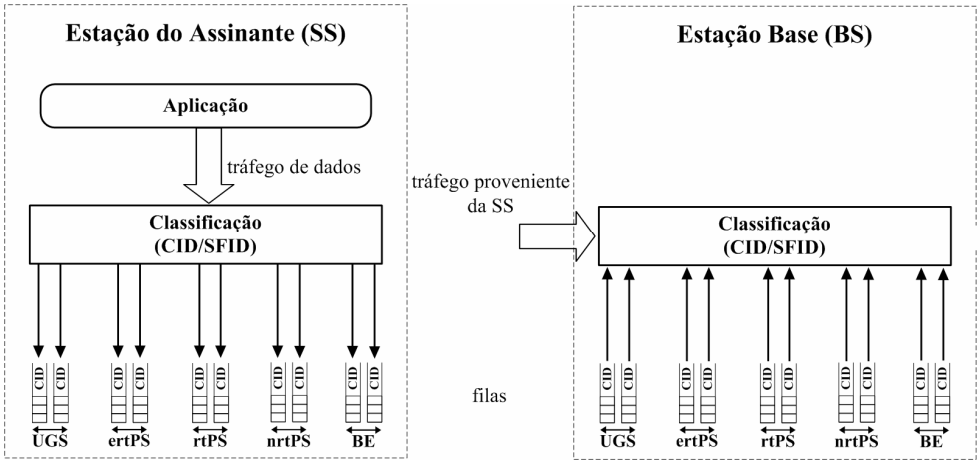

Figura 3.5: Mecanismo de Classificação do Padrão IEEE 802.16 (*uplink*) [1]

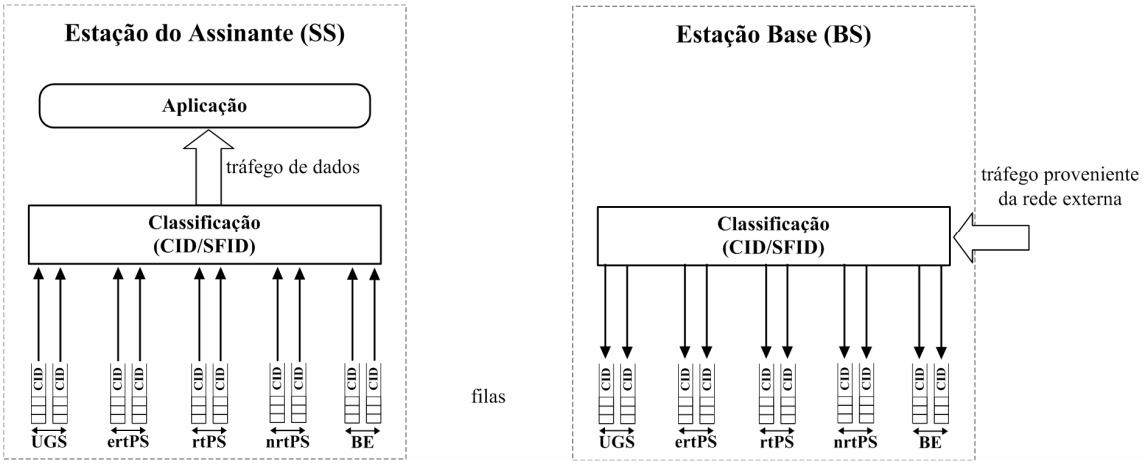

Figura 3.6: Mecanismo de Classificação do Padrão IEEE 802.16 (*downlink*) [1].

O grupo de ferramentas que oferece suporte à provisão de QoS para tráfegos *downlink* e *uplink* incluem: funções de configuração e registro dos fluxos de serviço, sinalização para o estabelecimento dinâmico de QoS com base nos fluxos de serviço e nos parâmetros de tráfego, escalonamento e parâmetros de tráfego para fluxos de serviço no sentido *downlink* e *uplink* e agrupamento de propriedades do fluxo de serviço em classes de serviço a fim de agregar requisições.

O canal de acesso do padrão IEEE 802.16 utiliza TDM no sentido *downlink* e TDMA no sentido *uplink*. O módulo de escalonamento de pacotes aloca largura de banda para conexões em função do número de *slots* alocados por conexão pelo canal TDM. Este módulo também determina quando uma conexão terá permissão para transmitir, caracterizando a conexão como ativa ou inativa.

3.5. Disciplinas de Escalonamento

As disciplinas de escalonamento definem a política de classificação e enfileiramento dos pacotes escolhidos para a transmissão na interface de saída da BS ou da SS. Em outras palavras, é o algoritmo (“disciplina”) de escalonamento que decide qual o próximo pacote será servido na fila de espera. Este algoritmo é um dos mecanismos responsáveis por distribuir a largura de banda do enlace para os diferentes fluxos.

Um algoritmo de escalonamento pode ser do tipo *work-conserving* ou *non-work conserving* [17]. No primeiro caso, o servidor “trabalha” sempre, isto é, havendo pacotes em espera, eles serão sempre transmitidos. No segundo caso, um nó só pode transmitir um pacote quando este se torna elegível, isto é, quando o tempo necessário para ele se manter em espera termina. Portanto, se no nó encontrarem apenas pacotes não elegíveis em espera, então o servidor permanecerá inativo. Este tipo de algoritmo de escalonamento foi projetado para aplicações que não toleram variações no atraso de transmissão. A desvantagem óbvia destes algoritmos é o desperdício de largura de banda durante os períodos em que apenas existem pacotes não elegíveis em espera.

3.5.1. Exemplos de Disciplinas de Escalonamento

A seguir serão apresentadas algumas das principais disciplinas de escalonamento encontradas na literatura.

3.5.1.1. FIFO (First In First Out)

Esta é a disciplina de escalonamento mais simples, todos os pacotes que chegam são colocados em uma fila comum e servidos pela ordem de chegada, como ilustra a Figura 3.7. Nenhum conceito de prioridade ou classe de tráfego é utilizado, todos os pacotes são tratados igualmente. Portanto, existe apenas uma única fila de saída, onde os pacotes recebidos são armazenados e enviados na mesma ordem em que chegaram [12].

Quando os pacotes encontram a fila cheia são descartados, desta maneira, estes pacotes descartados são perdidos. Como a disciplina FIFO trata todos os pacotes de maneira igual, não é possível prover diferentes níveis de QoS para fluxos distintos. Nesta disciplina as fontes de tráfego mal comportadas podem consumir toda a largura de banda disponível. Assim, tráfegos em rajada podem causar atrasos inaceitáveis em tráfegos sensíveis a atraso e

pacotes pertencentes a tráfegos de maior prioridade podem ser perdidos devido ao *overflow* na fila, provocados possivelmente pelos tráfegos de menor prioridade.

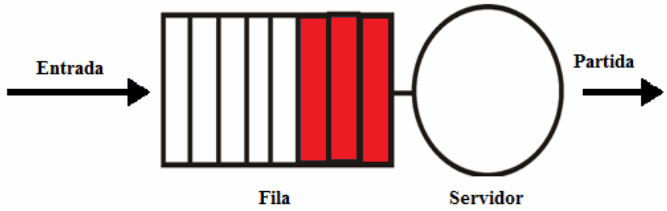

Figura 3.7: Modo de Operação da Disciplina FIFO [12].

3.5.1.2. RR (*Round Robin*)

No algoritmo RR seleciona-se o tráfego através de uma forma rotativa. Assim, o algoritmo percorre as classes presentes na fila, servindo um pacote de cada classe que contenha pelo menos um representante na fila [12]. Em outras palavras, o sistema seleciona um pacote de cada fila de espera de maneira rotativa. Este algoritmo também é muito simples, porém favorece os fluxos que contêm pacotes com maior comprimento, pois o pacote é servido independentemente do seu comprimento. Na Figura 3.8 apresenta-se uma abstração da forma que o algoritmo trabalha. Neste caso, atribuiu-se duas classes de serviço, sendo que o pacote da classe com a coloração verde é servido primeiramente que o segundo pacote da classe com a coloração vermelhada, devido à função de rotatividade do algoritmo RR, mesmo que o segundo pacote da classe com coloração vermelha tenha chegado primeiro, servindo assim um pacote de cada classe [12].

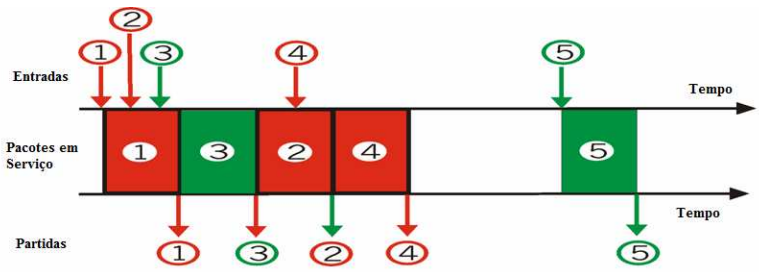

Figura 3.8: Modo de Operação da Disciplina RR [12].

3.5.1.3. WRR (Weighted Round-Robin)

A disciplina de escalonamento WRR define várias filas de espera com prioridades distintas, cujos fluxos de serviço são classificados e destinados a uma determinada fila de espera [12]. O tráfego é selecionado de forma rotativa com pesos. Dessa forma, em contraste com a disciplina de escalonamento RR que favorece fluxos com pacotes de maior comprimento, o WRR serve os pacotes de tamanho variável sem prejudicar os pacotes de menor tamanho e atribui uma melhor distribuição da largura de banda a cada fila de espera.

3.5.1.4. WFQ (Weighted Fair Queuing)

O algoritmo WFQ, que é também denominado de *Packet Generalized Processor Sharing* (PGPS), opera com a mesma filosofia do algoritmo GPS (*Generalized Processor Sharing*), ou seja, WFQ é uma aproximação baseada em pacote do algoritmo GPS [14]. GPS é um algoritmo idealizado que assume que um pacote pode ser dividido em bits e cada bit pode ser escalonado separadamente. WFQ é uma aplicação prática do GPS conforme atribuem tempos finais para pacotes e seleciona pacotes na ordem crescente dos seus tempos finais. A WFQ diferencia do algoritmo GPS em dois aspectos [14]:

- A WFQ emula o algoritmo GPS, porém as filas têm pesos diferentes. A ponderação permite que cada uma das conexões obtenha uma porção ponderada da largura de banda do canal;
- Os tempos de finalização (ou tempos virtuais de término) com que se marca cada pacote de uma conexão, corresponde ao tempo em que esse pacote deveria abandonar a fila, segundo o estabelecido por uma disciplina GPS. No WFQ é necessário calcular o instante em que o pacote deixaria o servidor num sistema GPS. Os pacotes vão posteriormente ser servidos por ordem destes instantes de partida.

A Figura 3.9 apresenta uma ilustração do modo de operação da disciplina WFQ, observa-se que os pacotes que chegam são classificados e enfileirados por classe em

diferentes filas. O escalonador atende de modo cíclico as filas. Os pesos de cada fila são denotados pelo valor de w_i.

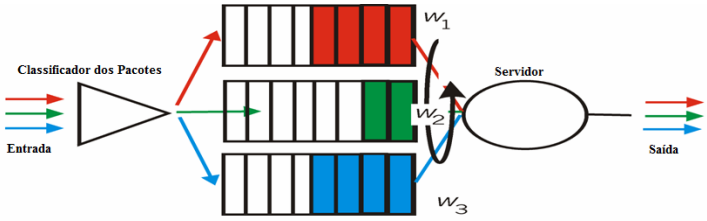

Figura 3.9: Modo de Operação da Disciplina WFQ [12].

Existem outros algoritmos de escalonamento que são variantes do WFQ, dentre estes, podem-se citar o *Self-Clocking Fair Queuing* (SCFQ) e o *Virtual Clock* (VC).

3.5.2. Características Desejáveis para as Disciplinas de Escalonamento

Uma disciplina de escalonamento desempenha um papel fundamental para garantir que os requisitos de QoS sejam atendidos e tem grande influência no desempenho da rede. Em [17] citam-se algumas das características que tais disciplinas têm que apresentar para que elas consigam satisfazer tais exigências:

- Flexibilidade: O algoritmo de escalonamento deve ser capaz de suportar usuários com diversos requisitos de QoS e também satisfazer suas exigências. Idealmente, o projeto de um algoritmo de escalonamento deve ser flexível o suficiente para que necessite de mudanças mínimas, quando for utilizado em diferentes redes ou até mesmo em uma tecnologia diferente.
- Simplicidade: O algoritmo de escalonamento deve ser simples. Neste sentido, ele deve apresentar uma simplicidade conceitual que permita realizar uma análise controlável do algoritmo de tal forma que a análise da distribuição ou o pior caso de determinados parâmetros, tais como atraso e vazão, possa ser analisado. A simplicidade de implementação deve permitir a utilização eficiente do algoritmo em larga escala.

- Proteção: Um algoritmo de escalonamento precisa ser capaz de proteger os usuários bem-comportados de fontes de dados variáveis, tais como aquelas que geram tráfego de melhor esforço (BE). Após a admissão na rede, os usuários negociam um acordo de nível de serviço (SLA) (por exemplo, um usuário irá especificar a taxa de pico e a taxa média de tráfego). Quando uma conexão não suportar o SLA, pode provocar flutuações do tráfego na rede. Neste caso, o algoritmo de escalonamento tem que garantir que tais flutuações não afetem o comportamento das outras conexões na rede.
- Justiça: Além de satisfazer os requisitos de QoS do usuário, o algoritmo de escalonamento deve garantir que um nível razoável de justiça seja mantido entre os usuários.
- Utilização do Enlace: O algoritmo de escalonamento é requerido para alocar largura de banda entre os usuários, de tal maneira que maximize a utilização do enlace. O algoritmo de escalonamento precisa garantir que os recursos não sejam alocados para usuários que não tenham dados suficientes para transmitir, evitando desperdício de recursos.

3.5.3. Disciplinas de Escalonamento no Padrão IEEE 802.16

No padrão IEEE 802.16, um fluxo de serviço com os parâmetros de QoS é gerado quando uma requisição de conexão é concedida. O escalonador implementado na BS calcula os requisitos de atraso e vazão para o tráfego *downlink* e *uplink*, e provê as concessões e *polls* em intervalos de tempos adequados. O tráfego *downlink* é realizado através de *broadcast*, onde o escalonador monta e enfileira as rajadas de acordo com os parâmetros de QoS dos *frames*.

O escalonamento no sentido *uplink* utiliza um esquema de concessão e *polls* mais complexo do que no sentindo *downlink*, pois exige coordenação entre a BS e cada SS

individualmente. Os algoritmos de escalonamento na SS realizam a distribuição da alocação da largura de banda concedida pela BS entre suas conexões. Neste mecanismo, não é necessário que a BS conceda largura de banda para cada conexão separadamente; este esquema é denominado de concessão de banda por estação assinante (GPSS). Contudo, o algoritmo de escalonamento implementado na SS pode ser diferente do implementado na BS. Porém, quando o mecanismo de concessão de banda por conexão (GPC) é utilizado, não necessita de um algoritmo de escalonamento na SS para decidir a alocação de largura de banda entre suas conexões, pois a BS concede banda por conexão.

O padrão IEEE 802.16 especifica um conjunto de parâmetros e funções para prover QoS, tais como sinalização, estabelecimento de conexão, classificação dos fluxos (UGS, rtPS, nrtPS, BE) e dentre outras, porém deixa em aberto qual disciplina de escalonamento deve ser usada.

Uma disciplina de escalonamento eficiente é essencial para garantir que os requisitos de QoS sejam atendidos e tem grande influência no desempenho da rede. Diante da diversidade de disciplinas de escalonamento encontradas, em [17] elas são classificadas em três grupos (Figura 3.10):

- **Algoritmos de escalonamento homogêneos**: são os algoritmos originais que foram propostos para as redes cabeada que são utilizados no padrão IEEE 802.16, com o objetivo de satisfazer os requerimentos de QoS para as diferentes classes de serviço. Algoritmos desta categoria não consideram a questão da qualidade do canal.
- **Algoritmos de escalonamento híbridos**: esta categoria contém algoritmos que usam a combinação de dois ou vários algoritmos de escalonamento propostos para as redes cabeadas na tentativa de satisfazer os requisitos de QoS para as diferentes classes de serviço. Alguns algoritmos desta categoria abordam a questão da variação da condição do canal.

- **Algoritmos de escalonamento oportunistas**: os algoritmos nesta categoria exploram primeiramente a variabilidade das condições do canal. Os algoritmos desta categoria também satisfazem os requerimentos de QoS e mantêm justiça entre as SSs.

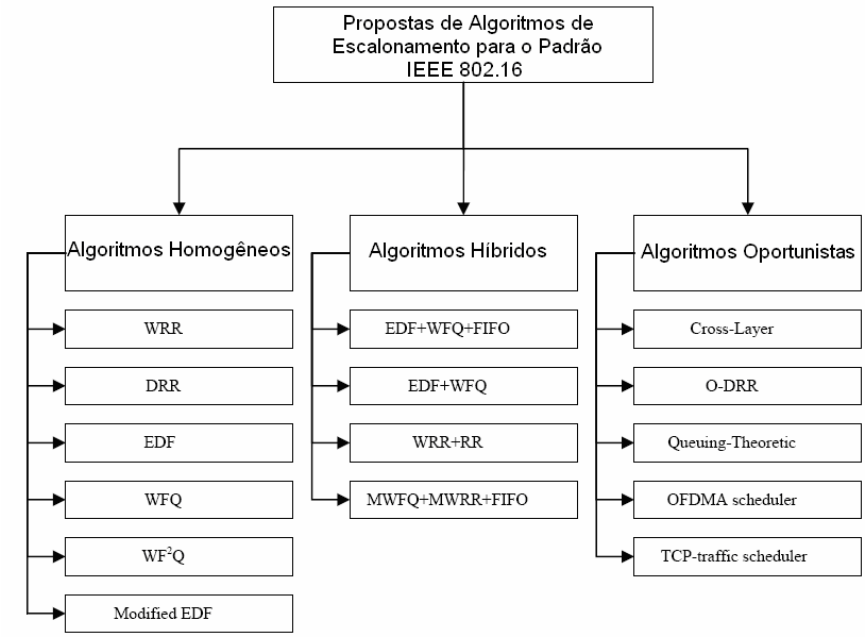

Figura 3.10: Taxonomia das Disciplinas de Escalonamento no Padrão IEEE 802.16 [17].

3.5.3.1. Propostas de Algoritmos de Escalonamento Homogêneos para o Padrão IEEE 802.16

As disciplinas WRR e DRR (*Deficit Round Robin*) são propostas e avaliadas para redes IEEE 802.16 em [20]. A disciplina WRR foi avaliada apenas no sentido *uplink*. Os pesos foram atribuídos em função das prioridades das classes de serviço, assim as classes com maior prioridade recebiam valores mais altos, por exemplo, para classe rtPS atribuiu-se valor mais alto em comparação com os pesos das classes nrtPS e BE. A disciplina DRR foi empregada no sentido *downlink*.

Em [22] avalia-se o desempenho da disciplina EDF (*Earliest Deadline First*) nas redes IEEE 802.16. A EDF é uma disciplina *work-conserving* originalmente proposta para aplicações de tempo real em redes de área de cobertura extensa. A disciplina atribui um prazo para cada pacote e aloca largura de banda para a SS que tem o pacote com o primeiro prazo.

O prazo pode ser atribuído para os pacotes da SS baseado nos requerimentos de atraso máximo nas SSs. A disciplina EDF é adequada para as classes de serviço que tenham requerimentos de atrasos rigorosos. Assim, os pacotes das classes UGS e rtPS serão escalonados primeiramente que os pacotes das classes nrtPS e BE. A disciplina escalona os pacotes das classes nrtPS e BE somente se não existir pacotes das classes UGS e rtPS. Portanto, uma deficiência desta disciplina é se o tráfego originado dos fluxos UGS e rtPS aumenta, os pacotes das classes nrtPS e BE não serão escalonados.

A disciplina de escalonamento WFQ (*Weighted Fair Queuing*) também é analisada em [22] no sentindo *uplink.* O desempenho da disciplina WFQ é comparado com a disciplina WRR. O seu desempenho foi superior na presença de pacotes de tamanho variável.

Em [23] um modelo hierárquico para o escalonamento de pacotes baseado na proposta de Bennet e Zhang [24] é proposto. O modelo é composto por três servidores de escalonamento: um servidor de escalonamento *hard*-QoS, um servidor de escalonamento *soft*-QoS e um servidor de escalonamento *best effort*. Todos os servidores implementam a disciplina WF^2Q (*Worst-case Fair Weighted Fair Queuing*) no sentindo *uplink* [25].

Em [26] a disciplina de escalonamento WF^2Q foi avaliada no sentido *downlink* nas redes de acesso IEEE 802.16. Algumas técnicas de compensação de erro foram adicionadas na disciplina WF^2Q. O algoritmo de escalonamento proposto tem o objetivo de preservar a diferenciação de QoS e a justiça no tráfego *downlink*.

3.5.3.2. Propostas de Algoritmos de Escalonamento Híbridos para o Padrão IEEE 802.16

Em [27] os autores propõem um esquema de escalonamento híbrido que combina os algoritmos de escalonamentos EDF, WFQ e FIFO. A alocação de largura de banda é realizada da seguinte maneira: todas as SSs que geram fluxos de serviço com prioridade alta alocam largura de banda até que elas não tenham quaisquer pacotes para serem enviados. O algoritmo

EDF é usado pela classe rtPS, WFQ para a classe nrtPS e FIFO para a classe BE. Além do algoritmo de escalonamento, é proposto um mecanismo de policiamento. Todos estes componentes juntos consistem em uma arquitetura de QoS. A desvantagem desta proposta é que as SSs que geram fluxos de serviço com prioridades menores irão essencialmente sofrer na presença de um número alto de SSs que geram fluxos com prioridades maiores, devido ao rigoroso mecanismo de alocação de banda.

Um esquema híbrido que utiliza EDF para a classe rtPS e WFQ para as classes nrtPS e BE é proposto em [28]. Este esquema difere do especificado em [27], de duas maneiras. Primeiro, o algoritmo WFQ é usado nas SSs de ambas as classes, nrtPS e BE. Segundo, a alocação de largura de banda não é feita de maneira estritamente rigorosa. Apesar dos detalhes da alocação de banda não serem especificados, é sucintamente mencionado que a largura de banda é alocada entre as classes de maneira justa. Porém, a desvantagem desta proposta é utilizar a disciplina WFQ, que apresenta uma alta complexidade computacional, para os fluxos BE que não exigem nenhuma garantia de QoS.

Uma arquitetura denominada de MUFSS (*Multi-class Uplink Fair Scheduling Structure*) foi proposta para satisfazer os requerimentos de atraso e vazão das múltiplas classes de serviço do padrão IEEE 802.16 em [29]. Neste trabalho apresentam-se duas propostas de disciplinas de escalomento: a MWRR (*Modified* WRR) e a MWFQ (*Modified* WFQ) baseadas nas disciplinas WRR e WFQ, respectivamente. A disciplina MWFQ é utilizada para escalonar os pacotes das classes UGS e rtPS, a MWRR para escalonar os pacotes da classe nrtPS e a FIFO para escalonar os pacotes da classe BE. Utiliza-se o modo de concessão de banda GPSS; assim, os escalonadores também são implementados nas SSs para alocar a largura de banda entre suas conexões.

3.5.3.3. Propostas de Algoritmos de Escalonamento Oportunistas para o Padrão IEEE 802.16

Os algoritmos oportunistas propostos para o padrão IEEE 802.16 exploram a variação da qualidade do canal dando prioridade para as SSs com melhor qualidade de canal, ao mesmo tempo em que tenta satisfazer os requisitos de QoS para tráfego de múltiplas classes.

Em [31] propõe-se uma extensão oportunista da disciplina DRR como proposta para satisfazer as exigências de limite de atraso das múltiplas classes de serviço do padrão IEEE 802.16. Em [32] apresenta-se uma extensão oportunista das disciplinas WRR e RR. Os algoritmos são oportunistas no sentido de selecionarem as SSs com o perfil de rajada mais robusto. Estes algoritmos também são eficazes para as SSs com as menores prioridades na presença de um grande número de SSs com maiores prioridades.

Um algoritmo de escalonamento para sistema OFDMA com uma estrutura de quadro TDD para tráfego *uplink* e *downlink* no padrão IEEE 802.16 é apresentado em [30]. Um problema de otimização é primeiramente formulado para a alocação de largura de banda para as SSs. A alocação de largura de banda entre os serviços de escalonamento é baseada na prioridade, similar ao algoritmo híbrido proposto em [27]. Mais especificamente, a BS tenta satisfazer os requisitos das conexões UGS, em seguida, das conexões rtPS e nrtPS. Finalmente, qualquer largura de banda residual é distribuída entre as conexões BE.

3.6. Considerações Finais

Neste capítulo apresentou-se a importância da QoS na redes de computadores e alguns mecanismos de provisão de QoS nas redes IEEE 802.16. Uma das vantagens do padrão IEEE 802.16 é ser orientado a conexão; assim, os pacotes da rede de acesso IEEE 802.16 incluem dois identificadores, um por fluxo e um por conexão. O padrão IEEE 802.16 foi desenvolvido com a intenção de prover QoS para os diferentes tipos de aplicações, especificando várias

classes de serviço (UGS, rtPS, nrtPS e BE). Apesar do padrão IEEE 802.16 apresentar um conjunto de parâmetros e funções para prover QoS, não especifica qual disciplina de escalonamento deve ser utilizada.

Neste capítulo também apresentou-se alguns conceitos referentes às disciplinas de escalonamento e algumas considerações sobre a importância das disciplinas de escalonamento na provisão de QoS nas rede de acesso IEEE 802.16.

Diversas propostas são encontradas na literatura referentes a este tema. Neste capítulo abordaram-se algumas destas propostas, seguindo a classificação de [17], em três grupos: disciplinas de escalonamento homogêneas, híbridas e oportunistas.

Capítulo 4

PROPOSTA DE ALGORITMO DE CONTROLE DE ADMISSÃO DE CONEXÕES (CAC) PARA AS REDES IEEE 802.16

4.1. Introdução

No capítulo 3 foi descrito que apesar do padrão IEEE 802.16 ter sido desenvolvido com QoS em mente, deixou em aberto dois dos principais mecanismos na provisão de QoS: os mecanismos de escalonamento e o CAC. Todavia, a maioria dos trabalhos encontrados na literatura enfatizam apenas a questão do escalonamento, enquanto poucos destes trabalhos abordam a questão do CAC. Alguns trabalhos também abordam propostas de algoritmos de escalonamento juntamente com algoritmos simples de CAC, porém estes algoritmos de CAC não conseguem atender os requisitos mínimos dos fluxos de serviço para prover QoS a eles, principalmente para os fluxos de serviço típicos de aplicações em tempo real, que são sensíveis ao atraso. Em vista disto, neste capítulo será apresentada uma proposta de algoritmo de CAC baseado em *threshold* para as redes IEEE 802.16.

O texto deste capítulo será organizado da seguinte forma: A Seção 4.2 realiza uma abordagem sobre o mecanismo de CAC. A Seção 4.3 apresenta algumas questões importantes

do mecanismo de CAC no padrão IEEE 802.16, enfatizando que o padrão especifica uma arquitetura de QoS, porém não define como o mecanismo de CAC deve ser implementado. A Seção 4.4 aborda algumas propostas de CAC encontradas na literatura para o padrão IEEE 802.16. Em seguida, na Seção 4.5 apresenta-se o mecanismo de CAC proposto. Finalmente, na Seção 4.6 as considerações finais deste capítulo são apresentadas.

4.2. Controle de Admissão de Conexões

Em redes de comunicação uma solicitação de conexão é qualquer requisição para utilizar recursos da rede para diversos serviços, tais como voz, vídeo, e *web browsing*. Cada solicitação de conexão possui suas características e requisitos que devem ser atendidos para ter o mínimo de QoS. Uma solicitação de conexão será aceita ou rejeitada dependendo das condições da rede. O mecanismo que gerencia a admissão ou rejeição de conexões é conhecido como Controle de Admissão de Conexões (*CAC - Connection Admission Control*) e seu bom funcionamento é de fundamental importância para o bom desempenho da rede como um todo.

Em outras palavras, o CAC pode ser definido como o conjunto de ações tomadas pela rede durante a fase de estabelecimento da conexão que define quando um pedido de conexão pode ser aceito ou rejeitado dependendo das condições da rede [34].

Os mecanismos de CAC e policiamento são necessários para assegurar a utilização justa e eficiente da rede, principalmente em aplicações multimídias devido apresentarem tráfego de natureza variável [9]. O policiamento protege a rede de tráfegos que violam os parâmetros negociados durante o estabelecimento da conexão. O mecanismo de CAC restringe o número de usuários simultâneos na rede de forma a evitar a saturação do enlace. O esquema de CAC define se uma conexão é aceita ou rejeitada dependendo dos recursos da rede já alocados.

Um mecanismo de CAC eficiente deve atender vários requisitos para suportar toda a flexibilidade inerente dos diferentes tipos de tráfego e infra-estruturas de rede. Entre estes requisitos, podem ser citados:

- O tempo de resposta deve ser tal que o mecanismo de CAC possa tomar uma decisão em um curto intervalo de tempo;
- Deve existir uma margem de segurança de modo a garantir que os parâmetros de QoS negociados sejam satisfeitos quando todas as fontes estiverem se comportando como o negociado durante o contrato feito no estabelecimento da conexão;
- É interessante ao esquema de CAC possuir um mecanismo de policiamento associado para verificar a conformidade entre o tráfego negociado e o real, de modo a evitar que um tráfego excessivo na rede possa prejudicar a QoS das conexões admitidas na rede;
- O mecanismo de CAC deve suportar os tráfegos com taxa variável.

Na literatura encontram-se duas importantes abordagens referentes ao controle de admissão de conexões [63] [64] [65]: o controle de admissão baseado em parâmetros (PBAC - *Parameter-Based Admission Control*) e baseado em medidas (MBAC - *Measurement-Based Admission Control*).

Os métodos baseados em parâmetros utilizam descrições de tráfego previamente estabelecidas para calcular os recursos de rede disponíveis e decidir sobre a admissão de novas conexões. As descrições de tráfego seguem modelos determinísticos ou estocásticos, ou seja, a caracterização das conexões pode seguir modelos determinísticos ou estocásticos.

Os métodos PBAC geralmente oferecem garantias mais restritas de QoS (em termos de atraso e perdas de pacotes), baixo custo computacional e não maximizam os níveis de

utilização da rede. Em [63] [64] os autores identificam dois problemas do método PBAC, sendo a dificuldade de se caracterizar o tráfego de fontes com comportamento em rajadas, podendo ocorrer erros que superestimam ou subestimam a necessidade real de recursos e a dificuldade de policiar este tráfego estatisticamente modelado para evitar que utilize mais recursos do que foi negociado na ocasião da admissão.

Os métodos baseados em medidas não requerem uma caracterização precisa do tráfego, ou seja, não requerem uma especificação de tráfego muito precisa para exercer sua função. O MBAC não necessita guardar informações sobre as conexões já admitidas para avaliar a quantidade de recursos disponível na rede. A quantidade de recursos é estimada usando medidas realizadas diretamente sobre o tráfego presente na rede a cada instante. O processo de admissão de uma nova conexão é realizado através da comparação dos parâmetros da nova conexão com as medidas de cada instante da rede.

Uma vantagem que se pode citar dos métodos MBAC em comparação aos métodos PBAC, é que os métodos MBAC amenizam os problemas citados acima com relação à imprecisão da caracterização das conexões, porém diminui a capacidade do controle de admissão de oferecer garantias absolutas a aplicações pouco tolerantes às variações do atraso e da taxa de perdas.

Em [63] [64] apresentam-se algumas importantes questões a serem consideradas na elaboração dos algoritmos de MBAC: o risco de erros de estimativa podendo provocar erros nas decisões de admissão, a influência da dinâmica de chegada e partida de fluxos nessas estimativas e a quantidade de informação sobre o "passado" dos fluxos a ser usada no cálculo.

Os dois componentes básicos do MBAC são:

- O mecanismo de medição: é utilizado para estimar a carga atual da rede. Em [63] [64] citam-se três diferentes técnicas de mecanismos de medição: a janelas

de tempos (*Time-Window*), amostras de pontos (*Point Samples*) e média exponencial (*Exponencial Averaging*);

- O algoritmo de decisão: é usado para decidir se uma nova conexão será ou não admitida. Em [63] [64] apresentam-se quatro algoritmos de decisão: soma simples (*Simple Sum*), soma medida (*Measured Sum*), região de aceitação (*Acceptance Region*) e banda equivalente (*Equivalent Bandwidth*).

Normalmente, nas redes sem fio os recursos de radio são escassos e caros. Portanto, o uso eficiente de recursos de radio tem sido uma área de pesquisa constante em redes de comunicação sem fio. Os principais objetivos dos algoritmos de CAC nas redes sem fio são: fazer o melhor uso dos recursos de radio disponíveis, assegurar que os requisitos de QoS de todas as conexões admitidas sejam satisfeitos e evitar a saturação do enlace.

4.3. Controle de Admissão de Conexões no Padrão IEEE 802.16

O padrão IEEE 802.16 foi desenvolvido com QoS em mente, e especifica um conjunto de parâmetros e funções para prover QoS, tais como sinalização, estabelecimento de conexão, classificação dos fluxos, porém deixa em aberto como o mecanismo de CAC deve ser implementando. O padrão apenas define que a subcamada de Parte Comum da MAC é a responsável pelo esquema de CAC. O padrão IEEE 802.16 especifica que a BS ou a SS ao criar uma nova conexão associa esta conexão com um dos fluxos de serviço [1].

Para realizar o gerenciamento dos fluxos de serviço existem três tipos de mensagem [1] [4] [9]: DSA (*Dynamic Service Addition*) para adição de novo fluxo, DSC (*Dynamic Service Change*) para modificação dos parâmetros do fluxo de serviço e a DSD (*Dynamic Service Delete*) para excluir um fluxo de serviço existente. O processo de admissão de novo fluxo será realizado através de um processo de três vias (*three-way handshake*), para realizar tal processo definem-se três mensagens de gerenciamento DSA (Figura 4.1): DSA-REQ

(*Dynamic Service Addition Request*), DSA-RSP (*Dynamic Service Addition Response*) e DSA-ACK (*Dynamic Service Addition Acknowledgment*).

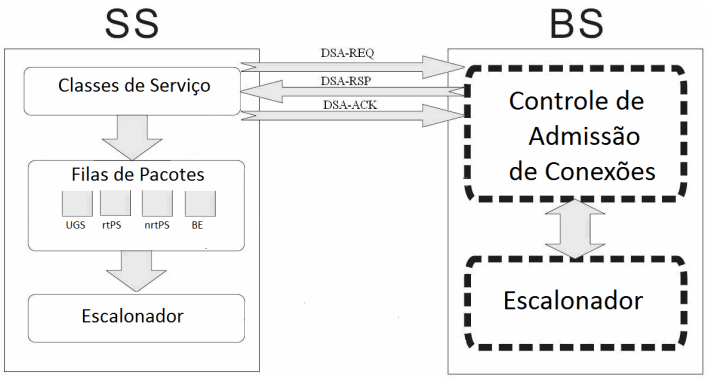

Figura 4.1: Processo de Adição de um Novo Fluxo de Serviço.

Na Figura 4.1 apresenta-se o processo de admissão de uma nova conexão, utilizando as mensagens de DSA. A seguir, apresenta-se com mais detalhes o processo de admissão de uma nova conexão na rede:

- Primeiramente, a SS envia uma solicitação para a BS através da mensagem DSA-REQ;
- A BS recebe a mensagem DSA-REQ da SS, em seguida o mecanismo de CAC utiliza as informações contidas na mensagem DSA-REQ para decidir ser admite ou não a nova conexão;
- Se a rede não tiver recursos disponíveis, a solicitação da SS será rejeitada. Para informar a SS que a solicitação foi rejeitada, a BS envia uma mensagem DSA-RSP para SS informando que a solicitação foi rejeitada;
- Caso a rede tenha recursos disponíveis e atenda as exigências da solicitação, a solicitação é aceita, e a nova conexão é admitida. A BS envia uma mensagem DSA-RSP para a SS informando que a solicitação foi aceita;
- A SS por final envia uma mensagem DSA-ACK para a BS confirmando o processo de admissão da nova conexão, caso a conexão seja admitida.

O padrão IEEE 802.16 especifica duas formas de concessão de largura de banda, concessão por conexão (GPC) e por estação assinante (GPSS) [1]. Quando a BS utiliza o modo de concessão GPC, o mecanismo de CAC realiza o controle por conexão separadamente. Em contraste, quando utiliza modo de concessão GPSS, o mecanismo CAC realiza o controle por SS; neste caso, o mecanismo de escalonamento implementado na SS que é responsável pela distribuição de largura de banda entre as conexões na SS.

Na Seção seguinte serão apresentas algumas propostas de CAC encontradas na literatura para o padrão IEEE 802.16.

4.4. Trabalhos Relacionados

Em [37] [38] apresenta-se um mecanismo de CAC muito simples baseado apenas na solicitação de banda. Neste mecanismo, a decisão se uma conexão é aceita ou rejeitada é baseada na largura de banda disponível do enlace. Se tiver largura de banda disponível e esta largura de banda for maior que a largura de banda solicitada pela nova conexão, a nova conexão é aceita, caso contrário, é rejeitada.

Em [3] apresenta-se uma proposta de um algoritmo de CAC para o padrão IEEE 802.16, no modo de operação PMP, usando concessão de largura de banda para as SSs (GPSS). O algoritmo implementado na BS requer informação sobre os atrasos dos fluxos na rede para prover garantias em termos de largura de banda e atraso. Os autores denominam esta proposta de "*predictive CAC*". O algoritmo verifica se a admissão do novo fluxo causará impacto nos outros fluxos já admitidos na rede, seguindo dois passos: verificação de largura de banda disponível e controle de atraso para os fluxos da classe de serviço UGS e rtPS. O algoritmo de CAC, com base nestas informações decide se admite ou rejeita a solicitação de um novo fluxo.

Em [40] os autores propõem um algoritmo de CAC que admite uma nova conexão somente se conseguir manter as garantias de QoS negociadas para todas as outras conexões admitidas e para a nova conexão. O algoritmo de CAC proposto decide se admite uma nova conexão dependendo das condições descritas abaixo:

- Conexões UGS: se a solicitação for do tipo UGS, ela deverá satisfazer a seguinte condição: a solicitação de *slots* baseada na máxima taxa dentro de seu intervalo de concessão nominal deverá ser menor ou igual ao número total de *slots* que podem ser acomodados pelo *tolerated grant jitter* baseada na largura de banda;
- Conexões rtPS: se a solicitação for do tipo rtPS, ela deverá satisfazer a seguinte condição: o número de *slots* dentro de um intervalo de *polling* nominal com suas mínimas taxas deverá se menor ou igual que o número total de *slots* que podem ser acomodados dentro de um *tolerated polling jitter* pela largura de banda total;
- Conexões nrtPS: é similar às conexões rtPS exceto que os valores dos parâmetros do intervalo de *polling* nominal e *tolerated polling jitter* são maiores que os rtPS devido a baixa prioridade em comparação com as conexões rtPS;
- Conexões BE: o algoritmo BE não requer qualquer garantia, o algoritmo verifica se existem *slots* livres disponíveis e os aloca para as conexões BE.

Em [41] apresenta-se uma proposta de uma arquitetura de QoS para o padrão IEEE 802.16. A arquitetura proposta consiste de um algoritmo de CAC e um algoritmo de escalonamento hierárquico. Ambos os algoritmos de escalonamento e CAC são baseados no esquema de codificação e modulação adaptativa (AMC – *Adaptive Modulation and Coding*), ou seja, o tempo necessário para transmitir *n* pacotes é calculado baseado no esquema AMC usado, isto é, o número de bits por símbolos OFDM.

Em [42] os autores apresentam uma proposta de CAC para o padrão IEEE 802.16 baseada em processos estocásticos e uma associação com as cadeias de Markov. A proposta é dividida em duas partes:

- Na primeira parte, modela-se a chegada de novas conexões de diferentes classes (UGS, rtPS, nrtPS). Entretanto, não realiza a modelagem das conexões BE, pois elas são sempre admitidas, porém não requerem nenhuma garantia.
- Na segunda parte, calcula-se os resultados do modelo (probabilidade de conexão e atraso) com base nos resultados obtidos na primeira parte.

Em [43] os autores propõem um esquema de CAC baseado no modelo de degradação. Neste modelo, a largura de banda para a classe de serviço nrtPS é degradada para um certo nível para satisfazer uma nova conexão em tempo real (UGS e rtPS). O modelo não fornece a largura de banda alocada para as conexões BE já admitidas para uma nova solicitação de conexão de um fluxo de serviço com prioridade alta, tais como o fluxo de serviço UGS e rtPS.

Alguns trabalhos apresentam uma proposta de CAC baseada no mecanismo de *token bucket* [44] [45]. Neste caso, cada conexão é controlada por dois parâmetros do mecanismo de *token bucket:* a taxa de *token bucket r* (bps) e o tamanho do *token bucket b*. Quando um fluxo espera para estabelecer uma conexão com a BS, envia estes dois parâmetros para a BS e espera a sua resposta, aceitando ou não a admissão de uma nova conexão. O fluxo rtPS envia um parâmetros extra *d* para especificar os requisitos de atraso.

Em [46] propõe-se um mecanismo de CAC baseado em reservas, denominado de *Reservation Based Connection Admission Control* (R-CAC), que define valores reservados para cada classe de serviço. A Figura 4.2 apresenta o esquema de alocação de banda proposto pelos autores. Considera-se que a BS rejeita as solicitações das conexões nrtPS somente quando a solicitação mínima de largura de banda não pode ser satisfeita. A conexão BE é

sempre admitida para usar toda a largura de banda disponível *C*, mas a largura de banda alocada para a conexão BE pode ser usada pelas conexões de prioridade mais alta; assim, a largura de banda para as conexões BE não pode ser garantida.

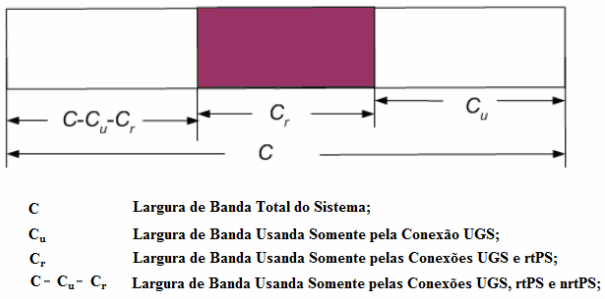

Figura 4.2: Proposta de CAC Baseado em Reservas [46].

Em [47] realiza-se um estudo da alocação de largura de banda sobre o nível de pacotes e introduz-se o mecanismo de CAC baseado na probabilidade de bloqueio ao nível de chamada e ao nível de rajada. Realizou-se neste trabalho uma análise diferenciada para os vários tipos de fluxos de serviço (UGS, rtPS, nrtPS e BE). Considerou-se que as conexões com taxas de transmissão menores somente no nível de chamada. As conexões com taxas maiores que são causadas pelo *download* de grandes arquivos ou períodos longos de atividades de vídeo VBR, aplicações típicas das classes nrtPS e rtPS, respectivamente, foram analisadas somente no nível de rajada. E por fim, a proposta do mecanismo de CAC foi analisada para as aplicações multimídias.

4.5. Proposta de Algoritmo de CAC Baseado em *Threshold* para as Redes IEEE 802.16

O algoritmo de CAC proposto é baseado em valores de *threshold*, onde o valor do *threshold* é a largura de banda máxima atribuída para cada fluxo; assim, a capacidade do enlace será divida entre os fluxos (UGS, rtPS e nrtPS). Desta forma, divide-se a capacidade do enlace em três faixas de largura de banda denominadas *threshold-UGS*, *threshold-rtPS* e *threshold-nrtPS*, que representam a largura de banda reservada para os fluxos UGS, rtPS e

nrtPS, respectivamente. Em outras palavras, o algoritmo de CAC baseado em *threshold* decide se admite ou rejeita uma conexão baseado nas faixas de largura de banda máximas atribuídas para os fluxos UGS, rtPS e nrtPS. O algoritmo de CAC baseado em *threshold* não reserva largura de banda para o fluxo BE, porque o padrão IEEE 802.16 especifica que o fluxo BE não recebe nenhuma garantia. Assim, os fluxos BE sempre serão admitidos na rede, porém, transmitem dados somente se existir largura de banda disponível.

Os valores de *threshold-UGS*, *threshold-rtPS* e *threshold-nrtPS* serão atribuídos na BS pelos administradores de rede, e o valor determinado a cada um destes *threshold* é baseado nos contratos de serviços (SLAs) feitos pelo provedor de serviço. Assim, a faixa de largura de banda reservada para cada um dos fluxos é baseada nos SLAs vendidos pelo provedor de serviço.

Os fluxos serão admitidos seguindo uma fila de prioridade; desta forma o fluxo UGS tem maior prioridade que o fluxo rtPS, a prioridade do fluxo rtPS é maior que a do fluxo nrtPS, e o fluxo BE não tem prioridade alguma. Este mecanismo de prioridade é utilizado com a intenção de priorizar os fluxos UGS e rtPS que são sensíveis ao atraso.

A seguir serão apresentadas algumas condições para que os fluxos sejam admitidos na rede [66-71]:

- Uma conexão UGS somente será aceita se a seguinte condição for satisfeita:

$$BW_{UGS} \leq C_{\text{threshold-UGS}} - \sum_{j=0}^{J} r_{\max-UGS}(j) \tag{4.1}$$

  onde BW_{UGS} é o valor da solicitação de largura de banda da conexão UGS, $C_{threshold\text{-}UGS}$ é o valor da largura de banda reservada para o fluxo UGS, $r_{max\text{-}UGS}$ é a taxa máxima solicitada para cada conexão ativa da classe de serviço UGS, J indica o número de conexões admitidas na rede;

- Uma conexão rtPS somente será aceita se a seguinte condição for satisfeita:

$$BW_{rtPS} \leq C_{\text{threshold-rtPS}} - \sum_{j=0}^{J} r_{mim-rtPS}(j) \quad (4.2)$$

onde BW_{rtPS} é o valor da solicitação de largura de banda da conexão rtPS, $C_{threshold\text{-}rtPS}$ é o valor da largura de banda reservada para o fluxo rtPS, e $r_{min\text{-}rtPS}$ é a taxa mínima solicitada para cada conexão ativa da classe de serviço rtPS;

- Uma conexão nrtPS somente será aceita se a seguinte condição for satisfeita:

$$BW_{nrtPS} \leq C_{\text{threshold-nrtPS}} - \sum_{j=0}^{J} r_{mim-nrtPS}(j) \quad (4.3)$$

onde BW_{nrtPS} é o valor da solicitação de largura de banda da conexão nrtPS, $C_{threshold\text{-}nrtPS}$ é o valor da largura de banda reservada para o fluxo nrtPS, e $r_{min\text{-}nrtPS}$ é a taxa mínima solicitada para cada conexão ativa da classe de serviço nrtPS;

- Conexões BE sempre serão admitidas na rede. Contudo, as conexões BE somente irão transmitir, se existir largura de banda disponível na rede; assim, para que uma conexão BE consiga transmitir dados na rede, os requisitos de prioridade das demais classes de serviço já devem ter sido atendidos, bem com a seguinte condição tem que ser satisfeita:

$$BW_{BE} \leq C_{total} - \sum_{i=0}^{I} \sum_{j=0}^{J_i - 1} r(i, j) \quad (4.4)$$

onde BW_{BE} é o valor da solicitação de largura de banda da conexão BE, C_{total} é a capacidade do enlace, e $r(i, j)$ é a taxa solicitada pelas j^{th} conexões das i^{th} classes de serviço já admitidas na rede.

A Figura 4.3 ilustra um exemplo do modo de operação do algoritmo de CAC proposto. Observa-se que a capacidade do enlace é dividida em faixas de largura de banda, *threshold-UGS, threshold-rtP*S e *threshold-nrtPS,* que representam a largura de banda máxima atribuída para os fluxos UGS, rtPS e nrtPS, respectivamente. Neste cenário, nota-se que os fluxos BE

admitidos na rede utilizam a largura de banda disponível para transmitir dados. Uma importante consideração a respeito do fluxo BE é que a largura de banda alocada para ele não é reservada, assim, o fluxo BE não tem nenhuma garantia.

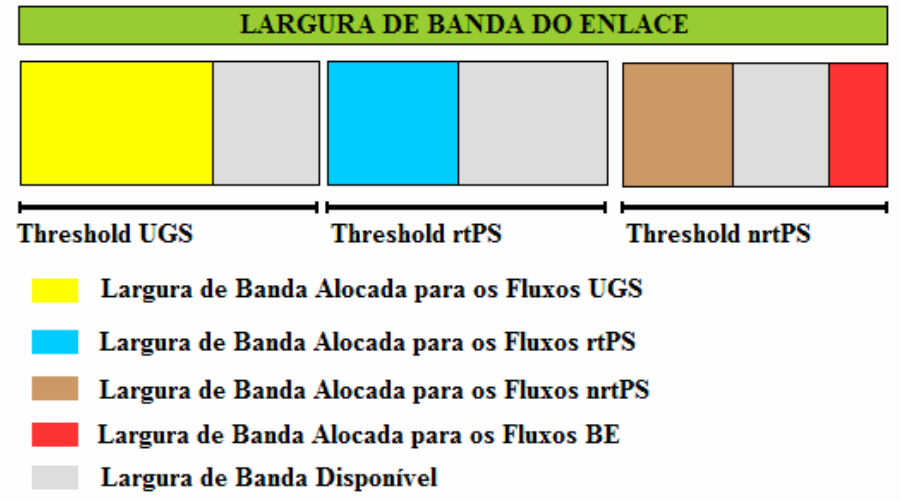

Figura 4.3: Representação da Proposta de CAC Baseada em *Threshold* [71].

O algoritmo de CAC proposto usará as mensagens DSA-REQ, DSA-RSP e DSA-ACK para realizar o processo de admissão de uma nova conexão e o modo de concessão de banda GPC.

4.5.1. Pseudo-algoritmo da Proposta do Algoritmo de CAC Baseado em *Threshold*

Nesta Seção será apresentado o pseudo-algoritmo da proposta do algoritmo CAC baseado em *threshold* [67].

- Fluxo UGS

- A BS recebe uma mensagem DSA-REQ de uma nova conexão UGS;
if (*UGS-threshold >= Largura-De-Banda-Solicitada*)
    - A BS aceita a solicitação para o fluxo UGS;
    - Atualiza *UGS-threshold = UGS-threshold - Largura-De-Banda-Solicitada*;
    - A BS envia uma mensagem DSA-RSP admitindo a nova conexão UGS;
else
    - A BS rejeita a solicitação UGS;
    - A BS envia uma mensagem DSA-RSP rejeitando a nova conexão UGS;

- Fluxo rtPS

- A BS recebe uma mensagem DSA-REQ de uma nova conexão rtPS;
if (*rtPS-threshold >= Largura-De-Banda-Solicitada*)
    - A BS aceita a solicitação para o fluxo rtPS;
    - Atualiza *rtPS-threshold = rtPS-threshold - Largura-De-Banda-Solicitada*;
    - A BS envia uma mensagem DSA-RSP admitindo a nova conexão rtPS;
else
    - A BS rejeita a solicitação rtPS;
    - A BS envia uma mensagem DSA-RSP rejeitando a nova conexão rtPS;

- Fluxo nrtPS

- A BS recebe uma mensagem DSA-REQ de uma nova conexão nrtPS;
if (*nrtPS-threshold >= Largura-De-Banda-Solicitada*)
    - A BS aceita a solicitação para o fluxo nrtPS;
    - Atualiza *nrtPS-threshold = nrtPS-threshold - Largura-De-Banda-Solicitada*;
    - A BS envia uma mensagem DSA-RSP admitindo a nova conexão nrtPS;
else
    - A BS rejeita a solicitação nrtPS;
    - A BS envia uma mensagem DSA-RSP rejeitando a nova conexão nrtPS;

4.6. Conclusões

Neste capítulo apresentou-se a importância do mecanismo de CAC no padrão IEEE 802.16 na provisão de QoS. Foram abordadas algumas características desejáveis de um algoritmo de CAC.

Como o CAC está em aberto nas redes de acesso IEEE 802.16, propõe-se um algoritmo de CAC baseado em *threshold*. O algoritmo de CAC proposto atribui uma largura de banda máxima para os fluxos UGS, rtPS e nrtPS, dividindo a capacidade do enlace entre os fluxos UGS, rtPS e nrtPS. Para o fluxo BE, o algoritmo de CAC baseado em *threshold* não reserva largura de banda, devido ao padrão IEEE 802.16 especificar que o fluxo BE não exige nenhuma garantia. Todavia, um fluxo BE transmitirá somente se a condição de prioridade dos demais fluxos estiver atendida.

No próximo capítulo o desempenho desta proposta será analisado, através de modelagem e simulação no sentido *uplink* de uma rede de acesso IEEE 802.16 com topologia PMP.

Capítulo 5

ANÁLISE DA PROPOSTA DE ALGORITMO DE CAC BASEADO EM *THRESHOLD* PARA AS REDES IEEE 802.16

5.1. Introdução

Neste capítulo a proposta de algoritmo de CAC baseado em *threshold* para as redes IEEE 802.16 descrita no Capítulo 4 será analisada por meio de modelagem e simulação. Para isso, implementou-se o algoritmo de CAC proposto no módulo WiMAX desenvolvido pelo NIST para *Network Simulator*, com a intenção de utilizá-lo nos experimentos de simulação, os quais foram realizados para diferentes cenários, tendo em vista a avaliação do comportamento do algoritmo de CAC proposto.

Na Seção 5.2 descreve-se as justificativas de se utilizar simulação, alguns simuladores existentes e a implementação do simulador para a proposta de CAC. Na Seção 5.3 apresenta-se a análise dos resultados obtidos através de modelagem e simulação de diferentes cenários. Finalmente, a Seção 5.4 contempla as conclusões deste capítulo.

5.2. Modelagem e Simulação

A utilização da modelagem e simulação tornou-se a maneira mais viável de realizar estudos de avaliação de desempenho na área de redes de computadores, pois, por exemplo, a implementação de um protocolo de redes em uma infra-estrutura adequada implicaria em grandes investimentos em equipamentos, o que pode inviabilizar estes estudos devido ao elevado custo. Por estas razões, quase sempre utilizam-se simuladores para fazer estudos de avaliação de protocolos de redes, principalmente no meio acadêmico.

5.2.1. Alguns Simuladores de Rede Existentes

Atualmente, existe uma variedade de simuladores utilizados para simular redes de computadores, sendo estes de códigos abertos ou não, gratuitos ou pagos, sendo de grande utilização na área acadêmica. A seguir, alguns dos principais simuladores serão descritos.

OMNeT++ (*Objective Modular Network Testbed* in C++) [48] é uma ferramenta de código aberto, implementada na linguagem C++, para a simulação de eventos discretos orientados a objeto. O OMNeT++ está disponível em diferentes versões para diversos sistemas operacionais, tais como, Unix, Linux e Windows. Atualmente, o OMNET++ está sendo bastante utilizado pela comunidade científica para estudos de avaliação de desempenho.

OPNET [49] é um simulador comercial que contém um conjunto de pacotes de produtos que permitem projetar, desenvolver, gerenciar e simular a infra-estrutura, os equipamentos e as aplicações de uma rede de computadores. Oferece também editor gráfico e animação da simulação.

GloMoSim (*Global Mobile Information Systems Simulation Library*) [50] é um simulador desenvolvido exclusivamente para redes sem fio, criado pelo Laboratório de Computação Paralela da Universidade da Califórnia em Los Angeles (UVLA). A coleção de bibliotecas do GloMoSim é implementada em PARSEC (*Parallel Simulation Environment for*

Complex Systems), que é uma linguagem baseada em C. O GloMoSim implementa um conjunto de protocolos de rede para comunicação sem fio, os quais estão organizados em uma arquitetura em camadas. No GloMoSim novos protocolos e módulos implementados em PARSEC podem ser facilmente adicionados à sua biblioteca para compor diferentes simulações.

NCTUns (*National Chiao Tung University Network Simulator*) é um simulador e emulador de redes capaz de simular diversos protocolos utilizados tanto nas redes IP cabeadas e sem fio. Sua tecnologia principal é baseada na inovadora metodologia de re-utilização do *kernel* do linux inventada por *Wang* [51] quando o mesmo estava obtendo seu título de PhD na universidade americana Harvard. Devido a essa metodologia revolucionária, NCTUns apresenta diversas vantagens que não são encontradas em simuladores tradicionais.

NS (*Network Simulator*) [52] é um simulador de eventos discretos, voltado para o desenvolvimento de pesquisas em redes de computadores. Neste trabalho, utiliza-se NS para simular a proposta devido aos seguintes fatores: possuir código de domínio público, ter um alto poder e versatilidade para criação de novos módulos, mesmo apresentando complexidade no desenvolvimento de tais módulos é bastante difundido no meio acadêmico e alvo de diversas pesquisas. A próxima Seção apresentará algumas características adicionais do *Network Simulator*.

5.2.2. *Network Simulator*

O NS é um dos mais populares simuladores de redes de computadores e muito difundido pela comunidade acadêmica. O NS é um simulador de eventos discretos, resultado de um projeto de desenvolvimento e pesquisa conhecido como VINT (*Virtual InterNetwork Tesbed*) [52].

Em 1989 surgiu a primeira versão do NS. Esta versão foi baseada no software REAL *Network Simulator*. A partir de 1995 passou a ter o apoio do DARPA, LBnL, Xerox PARC, UCB e da Universidade de Berkeley. O projeto do NS recebe contribuições substanciais de

diversos pesquisadores, pelo fato de ele ser um simulador de código aberto, ser totalmente gratuito, o que permite aos usuários mais experientes fazerem os ajustes que julgarem necessários.

O NS fornece suporte à simulação de diversas tecnologias de rede, e é suportado por vários sistemas operacionais, tais como, Linux, SunOS, FreeBSD, Solaris e Windows. No Windows, utiliza-se um emulador do *shell* do Linux denominado *Cygwin*.

O NS foi desenvolvido em duas linguagens: C++ (para manipulação dos dados) e OTCL (para a geração do *script* de simulação). O NS utiliza duas linguagens para aproveitar o que elas têm de melhor, a linguagem C++ é muito robusta e eficiente, apresentando eficiência na manipulação de *bytes* e cabeçalhos de pacotes, permitindo a construção de algoritmos que trabalham com grandes fluxos de dados. Por este motivo, o núcleo do NS foi desenvolvido em C++. A linguagem OTCL está mais relacionada à simulação, apresentando melhores resultados. É muito comum em uma simulação realizar algumas mudanças nos parâmetros do cenário que está sendo estudado, por exemplo, aumentar o número de estações e a carga da rede. A OTCL é uma linguagem interpretada e interativa.

Os *scripts* para as simulações são escritos em OTCL, devido à linguagem apresentar flexibilidade e facilidade nas mudanças dos parâmetros de simulação. A desvantagem de se utilizar a linguagem OTCL é o fato de ser mais lenta que a linguagem C++, isto, devido ela ser uma linguagem interpretada.

Quando se realiza uma simulação, o NS produz um ou mais arquivos de texto de saída contendo informações da simulação. Estes arquivos de textos são denominados de arquivos *trace*, e contém o *log* de todos os eventos ocorridos durante a simulação. O arquivo *trace* pode chegar a vários megabytes de tamanho, dependendo da simulação. A análise do arquivo *trace* deve ser realizada com muito cuidado, pois é uma das fases mais importantes após uma

simulação. Isto porque estes arquivos serão utilizados para gerar gráficos que terão grande importância na apresentação dos resultados da simulação.

Para a manipulação dos resultados obtidos através da simulação, foram desenvolvidos alguns programas em C e em *script* AWK (*Aho, Weinberger and Kernighan*) com o objetivo de manipular os arquivos *trace* para análise de alguns parâmetros de QoS, tais como, vazão e atraso. Os *scripts* em AWK foram desenvolvidos para organizar os dados gerados pela simulação em outro arquivo de dados, que serão utilizados pelo programa de análise desenvolvido em C. Neste trabalho, os resultados são coletados no nível MAC.

5.2.2.1 Descrição de Alguns Módulos do NS para redes IEEE 802.16

No NS não existe incorporado um módulo para as redes IEEE 802.16. Como o NS é uma ferramenta de simulação muito difundida na comunidade acadêmica, alguns módulos para o padrão IEEE 802.16 foram desenvolvidos por alguns grupos de pesquisa, destes podem ser citados: o NIST (*National Institute of Standards and Tecnology*) [54], NDSL (*Networks & Distributed Systems Laboratory*) [55], LWX (*Light Wimax*) [56], IC-UNICAMP (Universidade Estadual de Campinas) [57] e NS2MESH [58].

As principais características do módulo desenvolvido pelo NIST para as redes IEEE 802.16 são a inclusão da camada física baseada em OFDM e operando no modo TDD. Foram adicionadas algumas funcionalidades da camada MAC, tais como, gerenciamento de fluxo, escalonamento, mobilidade (IEEE 802.16e). Os principais parâmetros de configuração deste módulo consistem na largura de banda, freqüência, duração do *frame*, modulação e codificação do canal, dentre outros. Uma importante consideração referente a este módulo é que ele não contém qualquer mecanismo de CAC implementado.

O módulo NDSL foi desenvolvido pelos membros da Universidade de Gang Gung sob a supervisão de Chen. Este módulo fornece suporte a topologia PMP para os padrões IEEE

802.16d, IEEE 802.16e e IEEE 802.16j. O módulo contém funções fundamentais das subcamadas da MAC (CS e CPS) e da camada física. Além disso, implementa um mecanismo bem simples de CAC e escalonamento.

O módulo LWX suporta os padrões IEEE 802.16 e IEEE 802.16j. As principais metas do LWX são fornecer um módulo simples e flexível para as redes IEEE 802.16 para os diversos pesquisadores que tenham interesse em analisar as redes IEEE 802.16. O módulo inclui funcionalidades da camada MAC, tais como: diferentes taxas de modulação, retransmissão, vários algoritmos de alocação de banda para as redes IEEE 802.16 e IEEE 802.16j e algoritmos de escalonamento de fluxos.

O Laboratório de Redes de Computadores do Instituto de Computação da Unicamp desenvolveu um módulo para o padrão IEEE 802.16. Este módulo implementa algumas funcionalidades da camada MAC, tais como, mecanismo de alocação de banda, as classes de serviço, mecanismos de solicitação e concessão de banda, e suporte a QoS. Ele permite aos usuários configurar os requisitos de cada aplicação e suporta TDD e a topologia PMP. O canal *wireless* disponibilizado pelo NS foi utilizado. O módulo foi desenvolvido baseado no módulo projetado para o padrão DOCSIS (*Data Over Cable Service Interface Specification*) [59], entretanto, várias modificações foram realizadas para deixar o módulo compatível com o padrão IEEE 802.16.

Todos os módulos apresentados até agora foram desenvolvidos para as redes IEEE 802.16 na topologia PMP. Um módulo que permite simular a topologia *mesh* é o NS2MESH desenvolvido por Cicconetti. O acesso para a negociação de *sub-frame* de dados segue a especificação do padrão, através do processo *three-way handshake*, enquanto o escalonamento é implementado de acordo com o algoritmo *Fair End-to-end Bandwidth Access* (FEBA) descrito em [60]. O acesso ao *sub-frame* de controle é implementado de acordo com a função *Distributed Election* apresentada em [61]. A camada MAC

implementada por este módulo não está relacionada com os algoritmos de roteamento e os módulos de interferência física do NS.

5.2.3. Descrição da Implementação do Algoritmo de CAC Proposto no Módulo do NIST

Neste trabalho utilizou-se o módulo NIST descrito em [62]. Devido este módulo ser confiável, não ter qualquer mecanismo de CAC implementado e muito utilizado pela comunidade acadêmica. Para avaliação da proposta de algoritmo de CAC, este módulo foi modificado, conforme pode ser observado no diagrama de classes da Figura 5.1. Apenas a classe *MAC802_16* é apresentada, uma vez que a contribuição deste trabalho é implementada apenas nesta classe. A classe *MAC802_16* representa a camada MAC do padrão IEEE 802.16 e relaciona-se com as classes *peerNode*, *WimaxScheduler* e *ServiceFlowHandler*.

O algoritmo de CAC proposto, descrito na Seção 4.5, foi implementado na classe *WiMAXCAC*. Esta classe foi adicionada ao módulo NIST e está relacionada com a classe *ServiceFlowHandler*. Ela utiliza as mensagens de DSA-REQ, DSA-RSP e DSA-ACK da classe *ServiceFlowHandle* para a negociação da admissão de uma nova conexão. Quando a BS recebe uma nova solicitação de conexão através da mensagem DSA-REQ, o algoritmo de CAC proposto decidirá se a conexão será admitida ou não. Caso a rede tenha recursos para admitir a nova conexão, a BS envia à SS uma mensagem DSA-RSP confirmando que a rede tem recursos disponíveis para admitir a nova conexão, caso contrário, a BS envia à SS uma mensagem DSA-RSP informando que a nova conexão não pode ser admitida. A SS envia uma mensagem DSA-ACK à BS confirmando o recebimento da mensagem DSA-RSP. O procedimento de admissão de um novo fluxo através de três vias (*three-way handshake*) é implementado como especificado pelo padrão IEEE 802.16.

Os fluxos serão admitidos seguindo uma fila de prioridade, desta forma o fluxo UGS tem maior prioridade que o fluxo rtPS, a prioridade do fluxo rtPS é maior que a do fluxo nrtPS, e o fluxo BE não tem prioridade alguma.

A classe *ServiceFlowHandler* é responsável pelo gerenciamento das conexões *uplink* e *downlink,* e também armazena a lista dos fluxos de cada estação. A classe *ServiceFlowHandler* é responsável pela associação de cada conexão admitida com uma das classes de serviço da classe *ServiceFlow,* que contém os seus parâmetros de QoS. No anexo A contém o código fonte das classes *ServiceFlowHandler* e *WimaxCAC*.

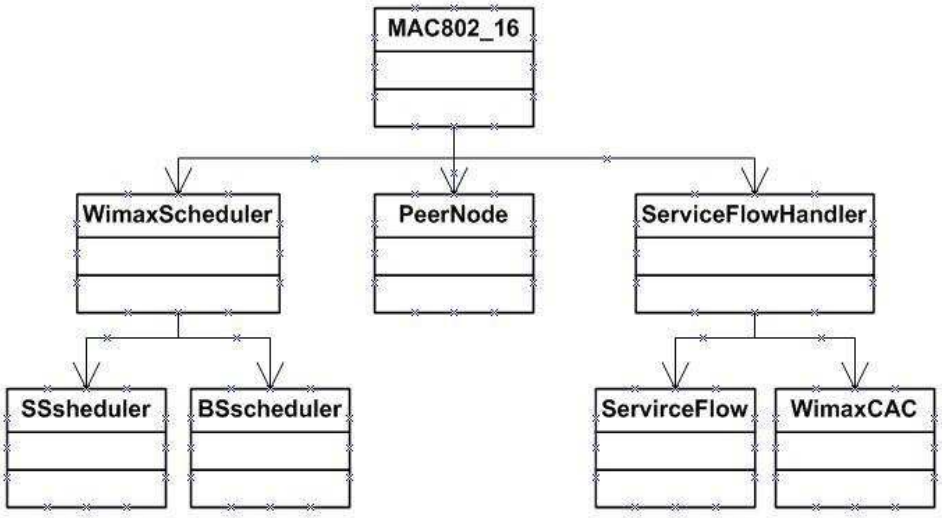

Figura 5.1: Diagrama de Classes.

5.3. Apresentação e Análise dos Resultados

Esta seção apresenta os resultados obtidos por meio de modelagem e simulação de diferentes cenários. O objetivo destes experimentos é investigar a eficiência do mecanismo de CAC proposto na provisão de QoS sob diferentes cenários, com diversas cargas de tráfego oferecido, bem como sua capacidade em compartilhar a largura de banda de maneira justa entre as classes de serviços.

Cada simulação foi rodada 5 vezes com sementes diferentes para gerar o intervalo de confiança de 95%. As Figuras a seguir mostraram valores médios obtidos e o intervalo de confiança de 95%.

5.3.1. Cenário 1

O primeiro cenário inclui uma BS, 5 SSs com conexões UGS, 10 SSs com conexões BE e o número de SSs com conexões rtPS varia de 5 a 30. As SSs consideradas são uniformemente distribuídas. Cada fonte de tráfego UGS gera um tráfego CBR (*Constant Bit Rate*) com taxa de 200 kbps no sentido *uplink*. As fontes de tráfego rtPS geram tráfego típico VBR (*Variable Bit Rate*) com taxa de 400 kbps no sentido *uplink*. Um esquema de escalonamento híbrido será utilizado; assim, para o fluxo rtPS a disciplina de escalonamento utilizada será a WRR (*Weighted Round Robin*) e para o fluxo BE a disciplina de escalonamento utilizada será a RR (*Round Robin*). A Tabela 5.1 descreve outros parâmetros importantes do cenário de simulação. Neste primeiro cenário, o mecanismo de CAC não será empregado. O objetivo deste primeiro experimento de simulação é mostrar o comportamento dos fluxos UGS, rtPS e BE com o aumento do número de conexões rtPS.

Tabela 5.1: Principais Parâmetros de Simulação

Parâmetros	***Valores***
Freqüência	5 GHz
Duplexação	TDD
Camada Física	OFDM
Ganho da Antena Transmissora	1
Ganho da Antena Receptora	1
Fator de Erro do Sistema	1
Potência de Transmissão	0,05
Receive power threshold	1,5e-10
Carrier sense power threshold	1,3e-10
Link adaptation	sim
Duração *Frame*	20 ms
Cyclic prefix (CP)	0,25
Tamanho dos Pacotes UGS	1024 bytes
Tempo de Simulação	100s
Modelo Propagação	*Two Ray Ground*
Antena	Omnidirecional

A Figura 5.2 apresenta a vazão das conexões UGS, rtPS e BE. Nota-se que a vazão das conexões BE diminui com o aumento do número de conexões rtPS. Isto acontece, porque as conexões BE somente irão transmitir, se existir largura de banda disponível na rede para elas. Isto acontece também, porque o padrão especifica que as conexões BE não exigem nenhuma garantia de QoS. No entanto, a vazão do tráfego UGS continua a mesma, independentemente do número de conexões rtPS, devido ao mecanismo de alocação de largura de banda constante definido pelo padrão para as conexões UGS.

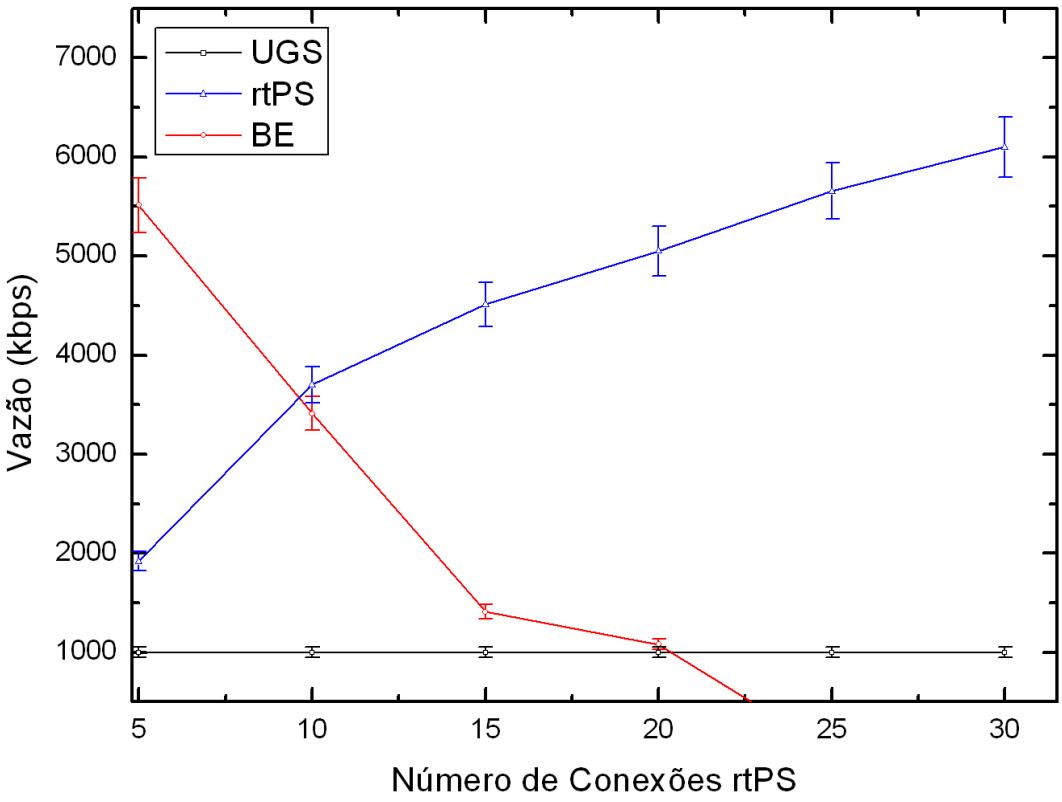

Figura 5.2: Vazão das Conexões UGS, rtPS e BE.

5.3.2. Cenário 2

O objetivo deste experimento é analisar a relação e a eficiência do algoritmo de CAC baseado em *threshold* proposto para as redes de acesso IEEE 802.16 utilizando diferentes disciplinas de escalonamento. A topologia do cenário considerado consiste de uma BS conectada a 12 SSs com conexões UGS e o número de SSs com conexões rtPS conectadas

com a BS varia de 5 a 25. Cada fonte de tráfego UGS gera tráfego CBR com taxa de 200 kbps no sentido *uplink*. As fontes de tráfego rtPS geram tráfego VBR com taxa de 500 kbps no sentido u*plink*. O fluxo rtPS será analisado utilizando três disciplinas de escalonamento: RR, WRR e mSIR. No ambiente de rede que o algoritmo de CAC proposto é utilizado, os valores dos *threshold* UGS e rtPS são 2 e 5 Mbps, respectivamente. A Tabela 5.1 descreve outros parâmetros importantes do cenário de simulação.

A Figura 5.3 apresenta o gráfico do atraso médio das conexões rtPS versus o número de conexões rtPS. Para todas as disciplinas de escalonamento analisadas, observa-se que quando o algoritmo de CAC é utilizado, o atraso médio dos fluxos rtPS é menor do que quando o mesmo não é utilizado. Isto acontece, devido ao algoritmo de CAC proposto restringir o número de conexões na rede, evitando assim a saturação do enlace. Desta forma, pode-se notar que quanto maior o número de conexões rtPS mais cresce o atraso médio dos fluxos rtPS quando nenhum algoritmo de CAC é empregado.

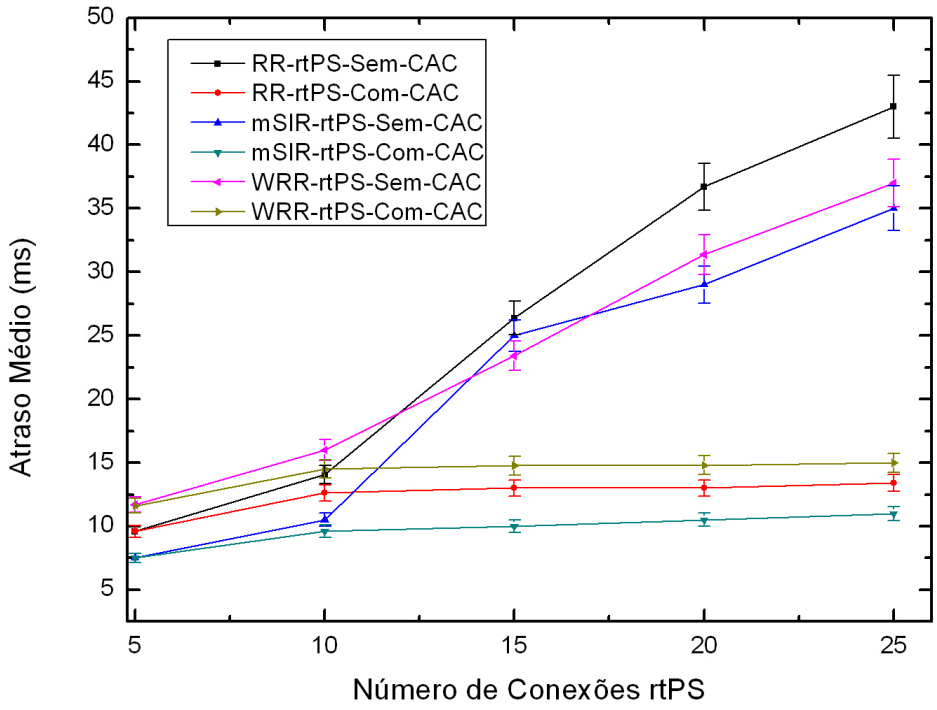

Figura 5.3: Atraso Médio das Conexões rtPS.

Observa-se na Figura 5.3 que quando o algoritmo de CAC proposto foi empregado, os atrasos médios dos fluxos rtPS ficam inalterados quando a taxa de tráfego rtPS gerada é igual ou superior ao valor do *threshold* rtPS (5 Mbps), isto acontece, porque o CAC limita a carga de tráfego das conexões admitidas ao valor do *threshold* rtPS, sendo admitidas na rede somente 10 conexões rtPS.

A Figura 5.4 apresenta o gráfico do atraso médio das conexões UGS versus o número de conexões rtPS. Para os fluxos UGS quando o algoritmo de CAC proposto é utilizado, o atraso médio é menor que quando o algoritmo de CAC não é empregado, devido ao mecanismo de CAC proposto limitar o número de conexões rtPS e UGS na rede, evitando a saturação do enlace. Porém, os atrasos médios dos fluxos UGS apresentaram valores menores que os dos fluxos rtPS porque o padrão IEEE 802.16 especifica um processo de alocação de banda fixa para a classe UGS. Assim, o escalonador implementado na BS aloca os recursos para os fluxos UGS de forma fixa.

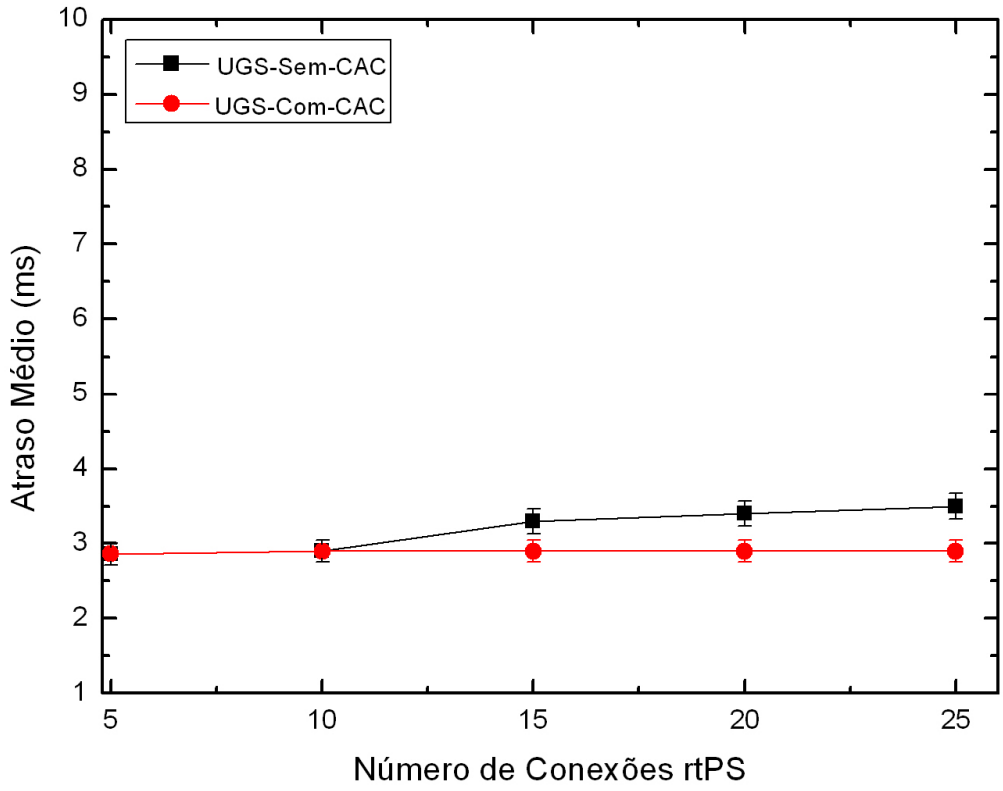

Figura 5.4: Atraso Médio das Conexões UGS.

A Figura 5.5 ilustra o gráfico da vazão das conexões rtPS versus o número de conexões rtPS. Quando o algoritmo de CAC proposto é usado, a vazão rtPS é limitada pelo valor do *threshold* rtPS (5 Mbps), e na simulação realizada a vazão é inferior à alcançada quando não se aplicou nenhum algoritmo de CAC. Desta forma, quando o tráfego gerado rtPS for superior ao valor do *threshold*, o algoritmo de CAC proposto implementado na BS rejeita todas as solicitações de novas conexões para o fluxo.

As conexões rtPS admitidas na rede, no cenário de simulação que emprega o algoritmo de CAC proposto, tem taxa de geração igual ao valor do *threshold* rtPS e com vazão máxima de 4,3 Mbps, 4,1 Mbps e 3,7 Mbps quando utiliza-se a disciplina mSIR, WRR e RR, respectivamente. O canal apresentou maior eficiência quando a disciplina de escalonamento mSIR foi empregada, apresentando uma eficiência de 85%. Isto aconteceu, porque a disciplina mSIR escalona primeiro os pacotes das SSs que possuem melhor relação sinal/ruído (SNR). Em contraste, quando o algoritmo de CAC proposto não está presente, todas as conexões são admitidas, saturando o enlace sem fio e aumentando os atrasos dos fluxos rtPS na rede, como pode ser observado na Figura 5.3. Apesar de limitar a utilização do enlace, o algoritmo de CAC mostrou-se eficiente, pois conseguiu atrasos desejáveis para aplicações típicas dos fluxos rtPS.

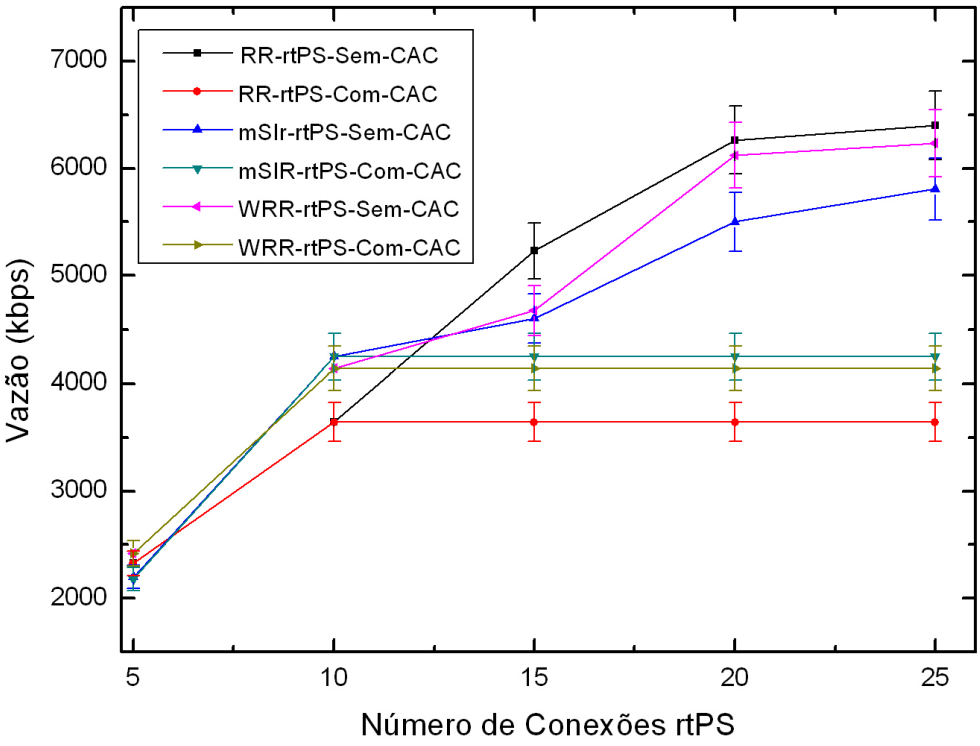

Figura 5.5: Vazão das Conexões rtPS.

Neste cenário considera-se que o valor do *threshold* UGS é de 2 Mbps. Assim, quando o algoritmo de CAC proposto é utilizado, a vazão das conexões UGS é limitada a este valor, como pode ser observado na Figura 5.6. Portanto, quando o algoritmo de CAC não está presente, todas as conexões UGS são admitidas na rede (ou seja, 12 conexões UGS são admitidas), sendo a vazão igual a 2,4 Mbps. Uma consideração importante sobre os resultados apresentados na Figura 5.6 é que a vazão das conexões UGS continua a mesma, independentemente se o número de conexões rtPS aumenta ou não. Isso ocorre porque o escalonador implementado na BS aloca os recursos para os fluxos de serviço UGS de forma fixa.

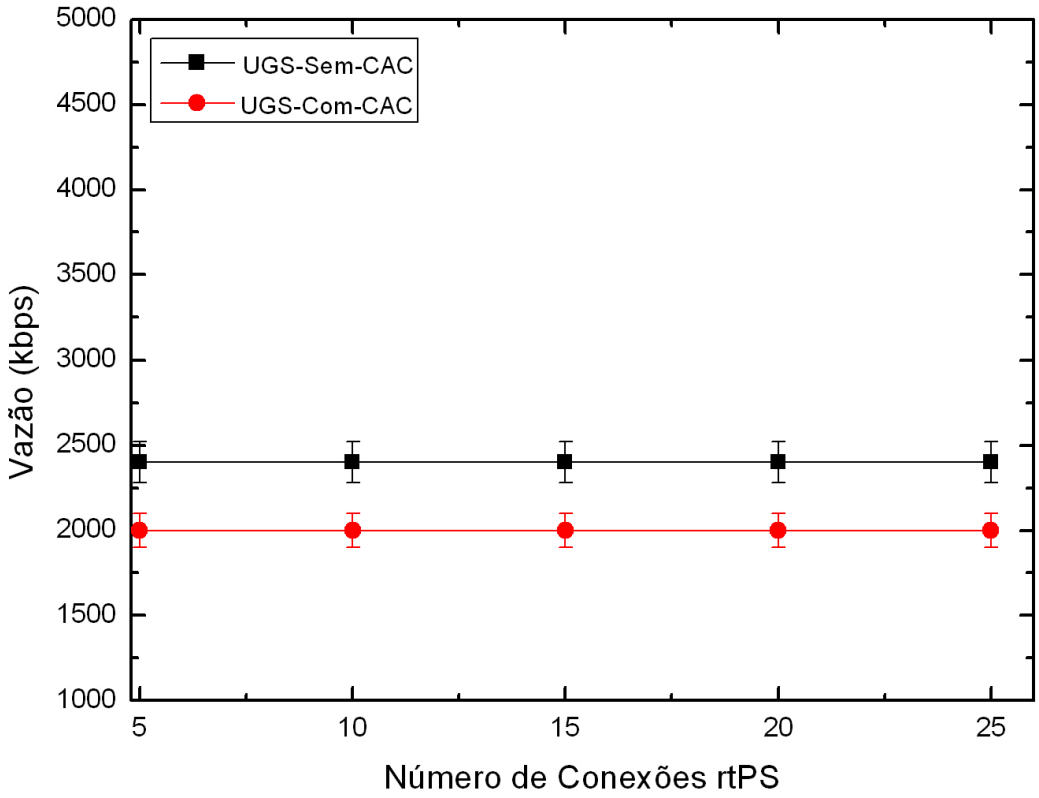

Figura 5.6: Vazão das Conexões UGS.

5.3.3. Cenário 3

Neste cenário considera-se uma BS, 10 SSs com conexões rtPS, e o número de SSs com conexões UGS varia de 5 a 25. Cada fonte de tráfego UGS gera um tráfego CBR com taxa de 200 kbps no sentido *uplink*. As fontes de tráfego rtPS geram tráfego típico VBR com taxa de 500 kbps no sentido *uplink*. A disciplina de escalonamento utilizada para o fluxo rtPS foi a WRR. Considera-se neste experimento que é associada apenas uma conexão para cada SS. Os valores dos *threshold* UGS e rtPS são 3 e 5 Mbps, respectivamente, para os ambientes de rede que empregam o mecanismo de CAC baseado em *threshold*. O objetivo deste experimento é verificar o comportamento dos fluxos UGS e rtPS aumentando-se o número de SSs com conexões UGS.

A Figura 5.7 apresenta o gráfico do atraso das conexões rtPS versus o número de conexões UGS. Observa-se que o atraso médio do fluxo rtPS é no máximo de 17 ms, quando o algoritmo de CAC proposto é empregado. Isto acontece, porque o algoritmo de CAC proposto limita o número de conexões UGS na rede, ou seja, somente 15 conexões UGS (totalizando 3 Mbps) são admitidas na rede. Assim, a BS rejeita todas as novas solicitações de conexões UGS quando o tráfego gerado pelas conexões UGS for igual ou superior ao valor do *threshold* UGS. Quando o algoritmo de CAC não está presente, o atraso médio aumenta em função do aumento do número de conexões rtPS. Desta forma, quanto mais conexões UGS são admitidas na rede maior será o valor do atraso médio do fluxo rtPS.

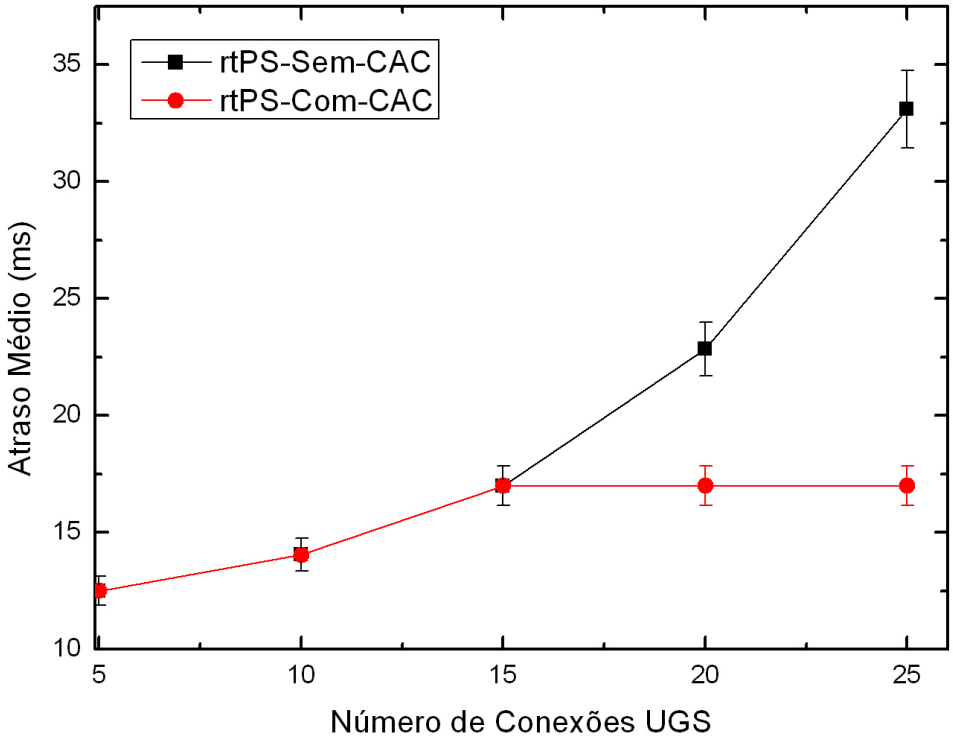

Figura 5.7: Atraso Médio das Conexões rtPS.

A Figura 5.8 ilustra o gráfico da vazão das conexões rtPS versus o número de conexões UGS. Nota-se que aumentando o número de conexões UGS na rede a vazão das conexões rtPS diminui. Entretanto, limitar o número de conexões UGS admitidas na rede através do algoritmo de CAC proposto permitiu minimizar este problema. Assim, são admitidas na rede somente 15 conexões UGS, obtendo uma vazão das conexões rtPS de 3,2

Mbps. A vazão das conexões UGS nesta rede fica limitada ao valor do *threshold* UGS que é 3 Mbps. Portanto, sem a utilização do algoritmo de CAC proposto os fluxos com maiores prioridades utilizam todos os recursos da rede, provocando inanição dos outros fluxos de prioridades menores.

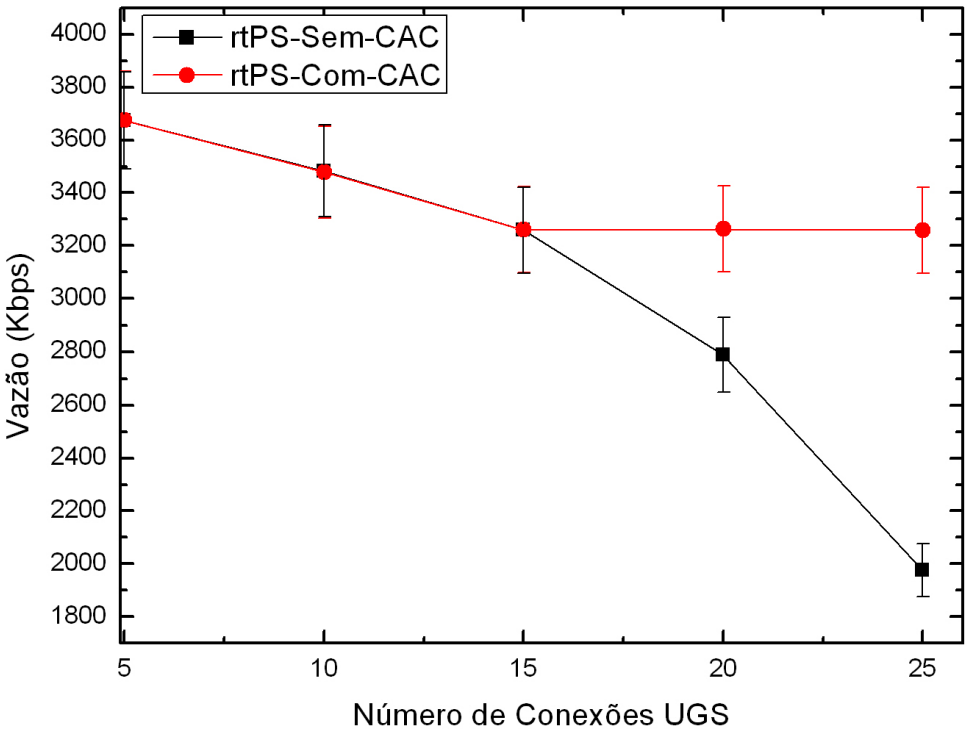

Figura 5.8: Vazão das Conexões rtPS.

A Figura 5.9 apresenta o gráfico do atraso do fluxo UGS versus o número de conexões UGS. Nota-se que quando o algoritmo de CAC proposto não está presente o atraso do fluxo UGS cresce com o aumento do número de conexões UGS na rede. O atraso aumenta porque a BS admite todas as solicitações de nova conexão UGS, causando uma saturação do enlace. Em contraste, quando o algoritmo de CAC baseado em *threshold* é utilizado, a BS limita o número de conexões na rede, utilizado como limiar o valor do *threshold* (3 Mbps). Assim, o algoritmo de CAC implementado na BS limita a vazão das conexões UGS a 3 Mbps, como pode ser observado na Figura 5.10, e conseqüentemente, limita o atraso das conexões UGS em aproximadamente 8 ms.

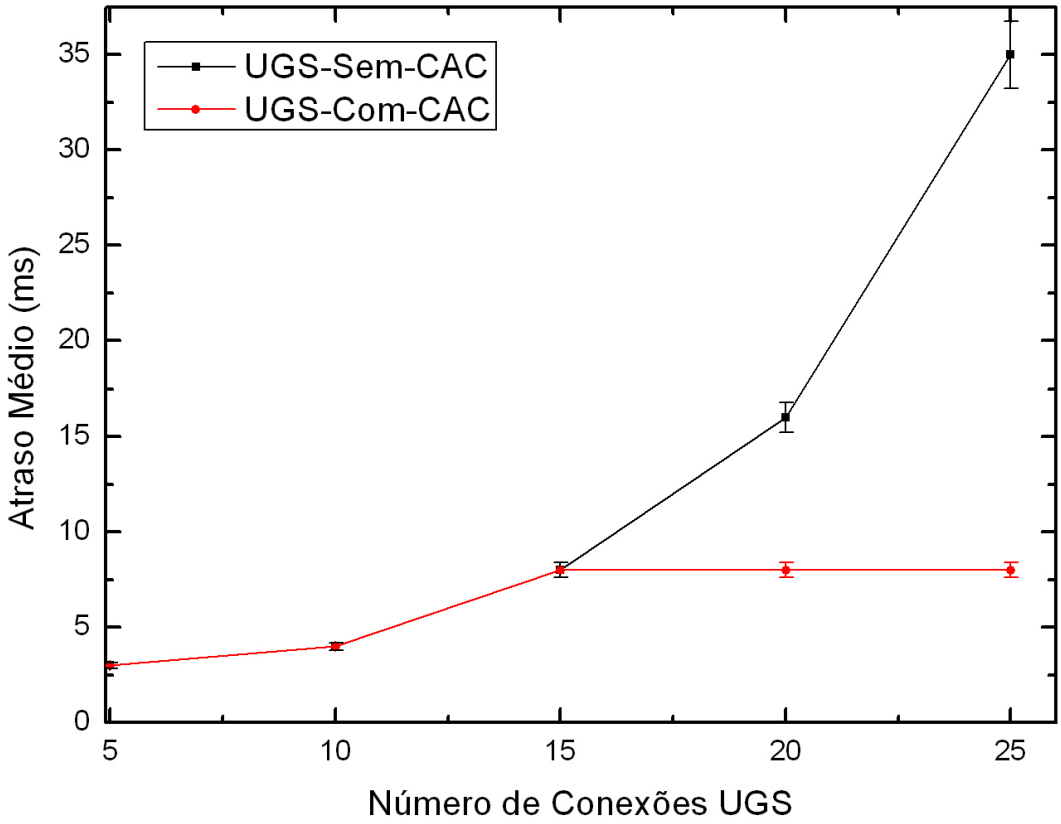

Figura 5.9: Atraso Médio das Conexões UGS.

A Figura 5.10 ilustra o gráfico da vazão das conexões UGS versus o número de conexões UGS. Quando nenhum mecanismo de CAC é empregado, a BS admite todas as conexões na rede; assim, a vazão das conexões UGS cresce proporcionalmente ao aumento do número de conexões UGS admitidas. Nota-se que a vazão do fluxo UGS é igual à taxa de geração dos dados. Isto ocorre, porque o padrão IEEE 802.16 especifica que a classe UGS suporta fluxos típicos de aplicações em tempo real que geram pacotes de dados com tamanho fixo periodicamente e, para suportar este serviço, oferece concessões de tamanho fixo periodicamente.

Observa-se que a vazão das conexões UGS é limitada ao valor do *threshold* UGS quando o algoritmo de CAC proposto é utilizado, ou seja, a vazão das conexões UGS fica limitada a 3 Mbps; assim, apenas 15 conexões UGS com taxa de 200 kbps cada, são admitidas na rede. Na Figura 5.9 observou-se que o aumento do número de conexões UGS provocou o

aumento do atraso, isto mostra que apesar do algoritmo de CAC proposto limitar a vazão das conexões UGS, conforme ilustra a Figura 5.10, conseguiu-se atrasos desejáveis para as aplicações típicas da classe UGS. Quando não se emprega o algoritmo de CAC proposto, admite-se um número maior de conexões que provoca um aumento do atraso proporcional ao aumento do número de conexões, ocasionando atrasos indesejáveis para as aplicações da classe UGS, as quais são sensíveis ao atraso.

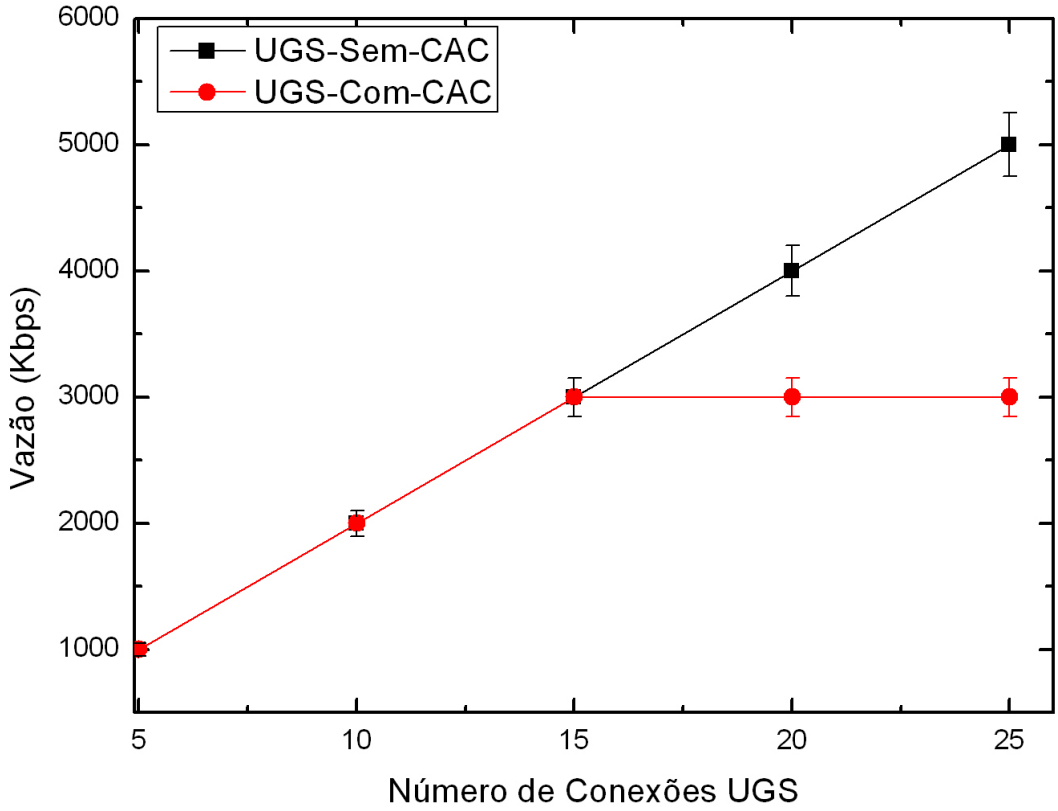

Figura 5.10: Vazão das Conexões UGS.

5.4. Conclusões

Neste capítulo avaliou-se, através de modelagem e simulação, o desempenho do algoritmo de CAC proposto no Capítulo 4. Diferentes cenários foram considerados com o objetivo de investigar a eficiência deste mecanismo de CAC. A relação entre algumas disciplinas de escalonamento e o esquema de CAC proposto também foi analisada. Para todas as disciplinas de escalonamento utilizadas, o algoritmo de CAC proposto apresentou melhores

resultados do que quando o algoritmo de CAC não estava presente. Como o algoritmo de CAC proposto restringe o número de conexões na rede, obteve-se em alguns casos, uma vazão inferior do que quando o algoritmo de CAC proposto não estava presente. Todavia, conseguiu-se, em todos os experimentos, atrasos desejáveis para as aplicações em tempo real, tais como aplicações de voz e vídeo, em uma rede de acesso IEEE 802.16. O algoritmo de CAC proposto evita a saturação do enlace, evitando que os requisitos de QoS sejam violados pela admissão de novas conexões. O algoritmo de CAC proposto mostrou-se eficiente levando em conta estas considerações. Em contraste, quando o algoritmo de CAC não está presente o enlace pode ser saturado, devido todas as conexões serem admitidas sem nenhuma restrição e o atraso médio pode crescer indefinidamente com o aumento do número de conexões admitidas na rede.

Capítulo 6

CONCLUSÕES GERAIS

Este trabalho abordou o padrão IEEE 802.16 como uma das tecnologias mais promissoras para o acesso banda larga sem fio em áreas metropolitanas. O padrão IEEE 802.16 permite a transmissão de dados sem fio provendo qualidade de serviço, sendo este o grande diferencial do padrão em relação a outras tecnologias sem fio. O padrão IEEE 802.16 especifica a camada de acesso ao meio (MAC) e a camada física, e neste trabalho ambas foram exploradas conceitualmente.

A arquitetura de QoS proposta para o padrão IEEE 802.16 foi explorada. Dessa forma, foram apresentados a teoria do modelo de objetos, as classes de tráfego, os fluxos de serviço e a classificação destes fluxos. Apesar de especificar uma arquitetura de QoS e ter sido projetado com QoS em mente, o padrão IEEE 802.16 deixa em aberto dois dos principais mecanismos na provisão de QoS: o escalonamento e o controle de admissão de conexões (CAC). O padrão apenas os define, porém não determina como eles devem ser implementados, deixando a cargo dos pesquisadores e fabricantes este processo de implementação.

O mecanismo de CAC decide se uma conexão é aceita ou rejeitada dependendo dos recursos já alocados na rede. Assim, o número de usuários simultâneos presentes na rede é restringido de forma a evitar a saturação do enlace sem fio. O CAC é fundamental para prover QoS, principalmente para as aplicações multimídia.

Na literatura encontram-se alguns trabalhos que abordam a provisão de Qualidade de Serviço nas redes IEEE 802.16. Destes, a maioria trata da questão de escalonamento, com poucos visando os algoritmos de CAC. Em vista disto, apresentou-se neste trabalho uma proposta de algoritmo de CAC baseado em *threshold* para o padrão IEEE 802.16. O *threshold* é a largura de banda máxima atribuída para cada fluxo; assim, a capacidade do enlace será dividida entre os fluxos (UGS, rtPS e nrtPS). Em outras palavras, o algoritmo de CAC proposto decide se admite ou rejeita uma solicitação de conexão baseado nas faixas de largura de banda máximas reservada para os fluxos UGS, rtPS e nrtPS. O algoritmo de CAC proposto não reserva largura de banda para o fluxo BE, porque o padrão IEEE 802.16 especifica que o fluxo BE não deve receber nenhuma garantia. Desta forma, os fluxos BE sempre serão admitidos na rede, porém, transmitem dados somente se existir largura de banda disponível.

O desempenho da proposta do algoritmo de CAC baseado em *threshold* foi avaliado através de modelagem e simulação. Para realizar a avaliação, utilizou-se a ferramenta de simulação *Network Simulator* (NS), por ser um dos mais populares simuladores de redes de computadores. Entretanto, o NS não disponibiliza um módulo para as redes IEEE 802.16. Como o NS é uma ferramenta de simulação muito difundida na comunidade acadêmica, alguns módulos para o padrão IEEE 802.16 foram desenvolvidos por alguns grupos de pesquisa. Dentre estes módulos encontrados na literatura, o escolhido para o desenvolvimento do trabalho foi o módulo NIST. O processo de implementação no módulo NIST da proposta de CAC foi descrito neste trabalho, e nesta implementação criou-se a classe *WIMAXCAC* no módulo NIST, uma vez que este não contém nenhum mecanismo de CAC implementado.

Para a avaliação da proposta de CAC através de modelagem e simulação, considerou-se diferentes cenários com a intenção de investigar a eficiência do algoritmo de CAC proposto. A relação entre algumas disciplinas de escalonamento e o algoritmo de CAC proposto também foi analisada. Para todas as disciplinas de escalonamento utilizadas, o algoritmo de

CAC apresentou melhores resultados do que quando o algoritmo de CAC não estava presente. Como o algoritmo de CAC proposto restringe o número de conexões na rede, obteve-se em alguns casos, uma vazão inferior do que quando o algoritmo de CAC proposto não estava presente. Além disso, conseguiu-se em todos os experimentos, atrasos desejáveis para as aplicações em tempo real, tais como aplicações de voz e vídeo, em uma rede de acesso IEEE 802.16 com topologia PMP. O algoritmo de CAC proposto evita a saturação do enlace, evitando que os requisitos de QoS sejam violados. Desta forma, o algoritmo de CAC baseado em *threshold* mostrou-se eficiente levando em conta estas considerações. Em contraste, quando o algoritmo de CAC não estava presente, o enlace foi saturado, devido todas as conexões serem admitidas sem nenhuma restrição e o atraso médio crescer com o aumento do número de conexões admitidas na rede.

Por fim, visando a continuidade deste trabalho sugere-se a combinação do algoritmo de CAC proposto com alguma disciplina de escalonamento, utilizando conceitos de *cross layer*, de forma a otimizar os níveis de provisão de QoS para os diferentes tipos de aplicações. Além disto, o algoritmo de CAC proposto poderia considerar as informações dos atrasos dos pacotes como um parâmetro extra para admissão de novas conexões na rede.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] IEEE Std 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks – Part. 16: Air Interface for Fixed Broadband Wireless Access Systems, Outubro 2004.

[2] WIMAX Forum Certification of Broadband Wireless Systems, Disponível em: http://www.wimaxforum.org/technology/downloads/Certification_FAQ_final.pdf, Acessado em: Janeiro 2009.

[3] IEEE 802.16 Working Group Maintenance Process, Disponível em: wirelessman.org/docs/06/80216-06_046.pdf, Acessado em: Junho 2007.

[4] ZHANG, Y. et all, Mobile WiMAX Toward Broadband Wireless Metropolitan Area Networks, ed. Auerbach Publications, 2008.

[5] LIMA, L. et all, WiMAX Padrão IEEE 802.16 para Banda Larga Sem Fio, Pontifícia Universidade Católica do Rio de Janeiro, Setembro 2004.

[6] NUAYMI, L. , WiMAX: Technology for Broadband Wireless Access, ed. John Wiley & Sons, 2007.

[7] BOTH, C. et all, Acesso de Banda Larga Sem-fio (WBA) e Redes Metropolitanas Sem-fio (WLAN) Baseados no Padrão IEEE 802.16 (WiMAX), Sociedade Brasileira da Computação, Agosto 2006.

[8] DELICATO, F. et all, Redes WiMAX: Arquitetura, Protocolos, Segurança e QoS, Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), Rio de Janeiro, Maio 2008.

[9] FREITAG, J. et all, Escalonamento com Qualidade de Serviço em Redes IEEE 802.16, Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), Belém-PA, 2007.

[10] EKLUND, C. et all, IEEE Standard 802.16: A Technical Overview of the WirelessMAN™ Air Interface for Broadband Wireless Access. IEEE Communications Magazine, Julho 2002.

[11] ISO DIS 13236, Information Technology - Quality of Service – Framework, ISO/OSI/ODP, Julho 1995.

[12] KUROSE, J. F. & ROSS, K. W. Redes de Computadores e a Internet: Uma Nova Abordagem Top-down, Addison Wesley, 3ª. Ed., São Paulo 2006.

[13] FREITAG, J., Provisão de Qualidade de Serviço em Redes IEEE 802.11, Universidade Estadual de Campinas, Agosto 2004.

[14] MARTÍNEZ, M., Algoritmos para QoS em Redes de Computadores, Tese, Pontifícia Universidade Católica do Rio de Janeiro, Março 2001.

[15] IEEE 802.16-2005 Working Group. IEEE Standard for Local and Metropolitan Area Networks – Part. 16: Air Interface for Fixed Broadband Wireless Access Systems for Mobile Users, Dezembro 2005.

[16] PAREKH, A. K., GALLAGER, R., A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case, IEEE/ACM Transactions on Networking, Vol.1, No.3, pp. 344-357, Junho 1993.

[17] DHRONA, P., A Performance Study of Uplink Scheduling Algorithms in Point to Multipoint WiMAX Networks, Tese, Queen´s University, Kingston, Canadá, Dezembro 2007.

[18] LI, V. et all, Scheduling Algorithms in Broadband Wireless Networks, Proceedings of the IEEE, pp.76-87, Janeiro 2001.

[19] SKRIKAR, L. et all, Packet Scheduling Algorithms to Support QoS in Networks, Tese, Indian Institute of Technology, Outubro 1999.

[20] CICCONETTTI, C. et all, Performance Evaluation of the IEEE 802.16 MAC for QoS Support, IEEE Transactions on Mobile Computing, vol.6, n°.1, pp.26-38, Janeiro 2007.

[21] KATEVENIS, M. et all, Weighted Round-Robin Cell Multiplexing in a General-Purpose ATM Switch Chip, IEEE Journal on Selected Areas in Communications, vol.9, pp.1265-1279, Outubro 1991.

[22] WANG, L. et all, A Study on the Performance of Scheduling Schemes for Broadband Wireless Access Networks, Proceedings of International Symposium on Communications and Information Technology, pp. 1008-1012, Outubro 2006.

[23] SHANG, Y., An Enhanced Packet Scheduling Algorithm for QoS Support in IEEE 802.16 Wireless Network, Proceedings of 3rd International Conference on Networking and Mobile Computing, pp.652-661, Agosto 2005.

[24] BENNETT, J. et all, Hierarchical Packet Fair Queuing Algorithms, IEEE/ACM Transactions on Networking, vol.5, pp.675-689, Outubro 1997.

[25] BENNETT, J. et all, WF^{2}Q: Worst-case Fair Weighted Fair Queuing, Proceedings of INFOCOM ‘96, pp.120-128, Março 1996.

[26] LERA, A. et all, Channel-Aware Scheduling for QoS and Fairness Provisioning in IEEE 802.16/WiMAX Broadband Wireless Access Systems, IEEE Network, vol. 21, issue 5, pp. 34-41, Outubro 2007.

[27] GANZ, A. et all, Packet Scheduling for QoS Support in IEEE 802.16 Broadband Wireless Access Systems, International Journal of Communication Systems, vol. 16, issue 1, pp. 81-96, Fevereiro 2003.

[28] VINAY, K. et all, Performance Evaluation of End-to-End Delay by Hybrid Scheduling Algorithm for QoS in IEEE 802.16 Network, Proceedings of International Conference on Wireless and Optical Communication Networks, Abril 2006.

[29] LIN, J. et all, Quality of Service Scheduling in IEEE 802.16 Broadband Wireless Networks, Proceedings of First International Conference on Industrial and Information Systems, pp.396-401, Agosto 2006.

[30] SETTEMBRE, M. et all, Performance Analysis of an Efficient Packet-Based IEEE 802.16 MAC Supporting Adaptive Modulation and Coding, Proceedings of International Symposium on Computer Networks, pp.11-16, Junho 2006.

[31] RATH, H., An Opportunistic Uplink Scheduling Scheme to Achieve Bandwidth Fairness and Delay for Multiclass Traffic in Wi-Max (IEEE 802.16) Broadband Wireless Networks, Proceedings of IEEE Global Telecommunications Conference, pp.1-5, Novembro 2006.

[32] SINGH, V. et all, Efficient and Fair Scheduling of Uplink and Downlink in IEEE 802.16 OFDMA Networks, Proceedings of IEEE Wireless Communications and Networking Conference, pp.984-990, Setembro 2006.

[33] NAIR, G. et all, IEEE 802.16 Medium Access Control and Service Provisioning, Intel Technology Journal. Vol. 08, Issue 03, Agosto 2004.

[34] DIAS, K. L. et all, Um Novo Esquema para Controle de Admissão de Chamadas em Redes Móveis sem Fio Baseadas no Protocolo IP, XXI Simpósio Brasileiro de Redes de Computadores (SBRC), Natal, RN, Maio 2003.

[35] ABDALLA, M. F. et all, Análise de Mecanismo de CAC para Redes ATM, 15º Simpósio Brasileiro de Telecomunicações (SBT), Recife, PE, Setembro 1997.

[36] FALOWO, E. O. et all, Joint Call Admission Control Algorithms: Requirements, Approaches, and Design Considerations, Computer Communications, Volume 31, pp. 1200-1217.

[37] CHALKE, J. B. et all, Survey of Call Admission Control – CAC in Wimax IEEE 802.16 Wireless MAN, India Institute Technology, Bombay, Março 2007.

[38] CHEN, J. et all, An Integrated QoS Control Architecture for IEEE 802.16 Broadband Wireless Access Systems, Proceedings of IEEE Global Telecommunications Conference (Globecom 2005), Vol. 5, pp. 3330-3335, Dezembro 2005.

[39] CASTRUCCI, M. et all, Connection Admission Control in WiMAX Networks, ICT-MobileSummit 2008 Conference Proceedings, Stockholm, Sweden, Junho 2008.

[40] CHANDRA, S. et all, An Efficient Call Admission Control for IEEE802.16 Networks, IEEE International Workshop on Local and Metropolitan Area Networks (LANMAN), Princeton, NJ, USA, Junho 2007.

[41] MSADAA, I. C. et all, An Adaptive QoS Architecture for IEEE 802.16 Broadband Wireless Networks, Mobile Ad hoc and Sensor Systems 2007 (MASS 2007), Outubro 2007.

[42] GHAZAL, S. et all, Performance Analysis of UGS, rtPS, nrtPS Admission Control in WIMAX Networks, IEEE International Conference on Communications (ICC 2008), Beijing, China, Maio 2008.

[43] WANG, H. et all, Dynamic Admission Control and QoS for 802.16 Wireless MAN, IEEE Wireless Telecommunications Symposium, pp. 60-66, Abril 2005.

[44] JIANG, C. et all, Token Bucket Based CAC and Packet Scheduling for IEEE 802.16 Broadband Wireless Access Networks," 3rd IEEE Consumer Communications and Networking Conference 2006 (CCNC 2006), Janeiro 2006.

[45] TSAI, T. et all, CAC and Packet Scheduling Using Token Bucket for IEEE 802.16 Networks, Journal of Communication, Vol. 1, Maio 2006.

[46] HOU, F. et all, Performance Analysis of a Reservation Based Connection Admission Scheme in 802.16 Networks, Proceedings of the Global Telecommunications Conference 2006 **(**GLOBECOM**)**, San Francisco, CA, USA, Dezembro 2006**.**

[47] WANG, H. et all, Admission Control and Bandwidth Allocation above Packet Level for IEEE 802.16 Wireless Man, International Parallel and Distributed Systems 2006 (ICPADS 2006), Julho 2006.

[48] OMNeT++ Community Site, Disponível em: http://www.omnetpp.org/, Acessado em: Janeiro 2008.

[49] OPNET Technologies, Disponível em: http://www.opnet.com/, Acessado em: Dezembro 2007.

[50] GloMoSim, Disponível em: http://pcl.cs.ucla.edu/projects/glomosim/, Acessado em: Julho 2008.

[51] NCTUns 5.0 Network Simulator and Emulator, Disponível em: http://nsl.csie.nctu.edu.tw/nctuns.html, Acessado em: Abril 2008.

[52] The Network Simulator - ns-2, Disponível em: http://www.isi.edu/nsnam/ns/, Acessado em: Junho 2007.

[53] COUTINHO, M. M., Network Simulator: Guia Básico para Iniciantes, Universidade Federal do Pará (UFPa), Agosto 2003.

[54] Módulo WiMAX NIST, Disponível em: http://w3.antd.nist.gov/seamlessandsecure/files/80216/doc/wimax_module.pdf, Acessado em: Setembro 2007.

[55] The Design and Implementation of WiMAX Module for ns-2 Simulator, Disponível em: http://ndsl.csie.cgu.edu.tw/wimax_ns2.php, Acessado em: Setembro 2007.

[56] Módulo WiMAX LWX, Disponível em: http://sites.google.com/site/lwxns2/, Acessado em: Setembro 2007.

[57] Módulo WiMAX UNICAMP, Disponível em: http://www.lrc.ic.unicamp.br/wimax_ns2/ , Acesso em: Junho 2008.

[58] Módulo WiMAX NS2MESH, Disponível em: http://cng1.iet.unipi.it/wiki/index.php/Ns2mesh80216, Acessado em: Janeiro 2008.

[59] DOCSIS: Data Over Cable Service Interface Specifications, Disponível em: http://docsis.org/, Acessado em: Setembro 2008.

[60] CICCONETTI, C. et all, Bandwidth Balancing in Multi-Channel IEEE 802.16 Wireless Mesh Networks, Proc. of the 26th Annual IEEE Conference on Computer Communications (INFOCOM 2007), Anchorage (USA), Maio 2007.

[61] CICCONETTI, C. et all, Performance Evaluation of the Mesh Election Procedure of IEEE 802.16/WiMAX, Proc. of the 10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2007), Grécia, Outubro 2007.

[62] BELGHITH, A. et all, Design and Implementation of a QoS-included WIMAX Module for NS-2 Simulator, SIMUTools 2008, França, Março 2008.

[63] FILHO, J. C. M. et all, Controle de Admissão e Controle de Carga para Redes IEEE 802.11 Infra-Estruturadas, IV Workshop de Comunicação Sem Fio (WCSF'02), São Paulo, SP, Outubro 2002.

[64] JAMIN, S. et all, Comparison of Measurement-based Admission Control Algorithms for Controlled-load Service, INFOCOM'97, Kobe, Japan, Abril 1997.

[65] GROSSGLAUSER, M. et all, A framework for Robust Measurement-based Admission Control, IEEE/ACM Transactions on Networking, Junho 1999.

[66] SOARES, C. L. & GUARDIEIRO, P. R., Connection Admission Control (CAC) in IEEE 802.16 Networks, Aceito para publicação na Fifth Advanced International Conference on Telecommunications (AICT), Veneza, Itália, Maio 2009.

[67] SOARES, C. L. & GUARDIEIRO, P. R., Threshold-Based Connection Admission Control for IEEE 802.16 Standard, Aceito para publicação no International Workshop on Wireless Multimedia Networking and Applications (WMNA'09), Wrexham, United Kingdom, Setembro 2009.

[68] SOARES, C. L. & GUARDIEIRO, P. R., A Threshold-Based Connection Admission Control for IEEE 802.16 Networks, Aceito para publicação na Fourth International Conference on Systems and Networks Communications (ICSNC 2009), Porto, Portugal, Setembro 2009.

[69] SOARES, C. L. & GUARDIEIRO, P. R., Proposed Threshold-Based Connection Admission Control (CAC) for IEEE 802.16 Standard, Aceito para publicação no International Workshop on Wireless, Mobile Networks & Applications (WiMoA 2009), Xiamen, China, Outubro 2009.

[70] SOARES, C. L. & GUARDIEIRO, P. R., Algoritmo de Controle de Admissão de Conexões para o Padrão IEEE 802.16, Aceito para publicação na VII Conferência de Estudos em Engenharia Elétrica, Uberlândia, Brasil, Agosto 2009.

[71] SOARES, C. L. & GUARDIEIRO, P. R., Proposta de um Algoritmo de Controle de Admissão de Conexões Baseado em *Threshold* para o Padrão IEEE 802.16, Submetido ao XXVII Simpósio Brasileiro de Telecomunicações (SBrT'09), Blumenau, Brasil, Setembro 2009.

Anexo A

```
#include "serviceflowhandler.h"
#include "mac802_16.h"
#include "scheduling/wimaxscheduler.h"

static int TransactionID = 0;

/*************************  Adicionado por Claiton  ************************/
int x12 = 0;
int banda;
int fluxo_trafego;

int threshold-UGS = 3000;
int threshold-rtps = 5000;
int threshold-nrtps = 4000;

/*************************************************************************/

/*
 * Create a service flow
 * @param mac The Mac where it is located
 */
ServiceFlowHandler::ServiceFlowHandler ()
{
  LIST_INIT (&flow_head_);
  LIST_INIT (&pendingflow_head_);
}

/*
 * Set the mac it is located in
 * @param mac The mac it is located in
 */
void ServiceFlowHandler::setMac (Mac802_16 *mac)
{
  assert (mac);

  mac_ = mac;
}

/**
 * Process the given packet. Only service related packets must be sent here.
 * @param p The packet received
 */
void ServiceFlowHandler::process (Packet * p)
{
  hdr_mac802_16 *wimaxHdr = HDR_MAC802_16(p);
  gen_mac_header_t header = wimaxHdr->header;
```

```
  //we cast to this frame because all management frame start with
  //a type
  mac802_16_dl_map_frame *frame = (mac802_16_dl_map_frame*) p->accessdata();

  switch (frame->type) {
  case MAC_DSA_REQ:
    processDSA_req (p);
    break;
  case MAC_DSA_RSP:
    processDSA_rsp (p);
    break;
  case MAC_DSA_ACK:
    processDSA_ack (p);
    break;
  default:
    printf ("Unknow frame type (%d) in flow handler\n", frame->type);
  }
  Packet::free (p);
}

/**
 * Add a flow with the given qos
 * @param qos The QoS for the flow
 * @return the created ServiceFlow
 */
ServiceFlow* ServiceFlowHandler::addFlow (ServiceFlowQoS * qos) {
  return NULL;
}

/**
 * Remove the flow given its id
 * @param id The flow id
 */
void ServiceFlowHandler::removeFlow (int id) {

}

/**
 * Send a flow request to the given node
 * @param index The node address
 * @param uplink The flow direction
 */
void ServiceFlowHandler::sendFlowRequest (int index, bool uplink)
{
  Packet *p;
  struct hdr_cmn *ch;
  hdr_mac802_16 *wimaxHdr;
  mac802_16_dsa_req_frame *dsa_frame;
  PeerNode *peer;
```

```
  //create packet for request
  peer = mac_->getPeerNode(index);
  p = mac_->getPacket ();
  ch = HDR_CMN(p);
  wimaxHdr = HDR_MAC802_16(p);
  p->allocdata (sizeof (struct mac802_16_dsa_req_frame));
  dsa_frame = (mac802_16_dsa_req_frame*) p->accessdata();
  dsa_frame->type = MAC_DSA_REQ;
  dsa_frame->uplink = uplink;
  dsa_frame->transaction_id = TransactionID++;
  ServiceFlow * serviceflow = new ServiceFlow();
  dsa_frame->serviceflow = serviceflow;
    if (mac_->getScheduler()->getNodeType()==STA_MN)
    ch->size() += GET_DSA_REQ_SIZE1;// (0); // change the data size
  else {
    //assign a CID and include it in the message
    Connection *data = new Connection (CONN_DATA);
    mac_->getCManager()->add_connection (data, uplink);
    if (uplink)
      peer->setInData (data);
    else
      peer->setOutData (data);
    dsa_frame->cid = data->get_cid();
    ch->size() += GET_DSA_REQ_SIZE1;// (1); // change the data size
  }

  wimaxHdr->header.cid = peer->getPrimary()->get_cid();
  peer->getPrimary()->enqueue (p);

}

/**
 * process a flow request
 * @param p The received request
 */
void ServiceFlowHandler::processDSA_req (Packet *p)
{
  mac_->debug ("At %f in Mac %d received DSA request\n", NOW, mac_->addr());

  Packet *rsp;
  struct hdr_cmn *ch;
  hdr_mac802_16 *wimaxHdr_req;
  hdr_mac802_16 *wimaxHdr_rsp;
  mac802_16_dsa_req_frame *dsa_req_frame;
  mac802_16_dsa_rsp_frame *dsa_rsp_frame;
  PeerNode *peer;
  Connection *data;

  //read the request
  wimaxHdr_req = HDR_MAC802_16(p);
```

```
  dsa_req_frame = (mac802_16_dsa_req_frame*) p->accessdata();
  peer = mac_->getCManager ()->get_connection (wimaxHdr_req->header.cid, true)-
>getPeerNode();

  //allocate response
  //create packet for request
  rsp = mac_->getPacket ();
  ch = HDR_CMN(rsp);
  wimaxHdr_rsp = HDR_MAC802_16(rsp);
  rsp->allocdata (sizeof (struct mac802_16_dsa_rsp_frame));
  dsa_rsp_frame = (mac802_16_dsa_rsp_frame*) rsp->accessdata();
  dsa_rsp_frame->type = MAC_DSA_RSP;
  dsa_rsp_frame->transaction_id = dsa_req_frame->transaction_id;
  dsa_rsp_frame->uplink = dsa_req_frame->uplink;
  dsa_rsp_frame->confirmation_code = 0; //OK
  /// Added by Aymen
  dsa_rsp_frame->serviceflow = dsa_req_frame->serviceflow;
  ///
  if (mac_->getScheduler()->getNodeType()==STA_MN) {
    //the message contains the CID for the connection
    data = new Connection (CONN_DATA, dsa_req_frame->cid);
    mac_->getCManager()->add_connection (data, dsa_req_frame->uplink);
    if (dsa_req_frame->uplink)
      peer->setOutData (data);
    else
      peer->setInData (data);
    ch->size() += GET_DSA_RSP_SIZE1;// (0); // change the data size
  } else {
    //allocate new connection
    data = new Connection (CONN_DATA);
    /// Added by Aymen
    // Take the flow parameter from the DSA-REQ
    ServiceFlow * serviceflow;
    serviceflow = dsa_req_frame->serviceflow;
   // Fill up the cid parameter of the serviceflow
    serviceflow->setCID(data->get_cid());

    // Fill up the service flow parameter of the DATA connection to add
    data->set_serviceflow(serviceflow);
    ///
    mac_->getCManager()->add_connection (data, dsa_req_frame->uplink);
    if (dsa_req_frame->uplink)
      peer->setInData (data);
    else
      peer->setOutData (data);
    dsa_rsp_frame->cid = data->get_cid();
    ch->size() += GET_DSA_RSP_SIZE1;// (1); // change the data size
  }

  wimaxHdr_rsp->header.cid = peer->getPrimary()->get_cid();
```

```
  peer->getPrimary()->enqueue (rsp);

}

/**
 * process a flow response
 * @param p The received response
 */

/***************************** Modificado por Claiton  ************************/

/*
* Classe processDSA_rsp modificada para implementação do mecanismo de CAC
*/

void ServiceFlowHandler::processDSA_rsp (Packet *p)
{
  mac_->debug ("At %f in Mac %d received DSA response\n", NOW, mac_->addr());

  Packet *ack;
  struct hdr_cmn *ch;
  hdr_mac802_16 *wimaxHdr_ack;
  hdr_mac802_16 *wimaxHdr_rsp;
  mac802_16_dsa_ack_frame *dsa_ack_frame;
  mac802_16_dsa_rsp_frame *dsa_rsp_frame;
  Connection *data;
  PeerNode *peer;

  // adicionado por claiton
  //int scheduling;
  int teste;
  int teste2;
  //

  //read the request
  wimaxHdr_rsp = HDR_MAC802_16(p);
  dsa_rsp_frame = (mac802_16_dsa_rsp_frame*) p->accessdata();
  peer = mac_->getCManager ()->get_connection (wimaxHdr_rsp->header.cid, true)-
>getPeerNode();

  //TBD: check if status not OK

  if (mac_->getScheduler()->getNodeType()==STA_MN) {
    //the message contains the CID for the connection
    data = new Connection (CONN_DATA, dsa_rsp_frame->cid);

    /// Added by Aymen
    // Fill up the flow parameter of the connection
    ServiceFlow * serviceflow;
    serviceflow = dsa_rsp_frame->serviceflow;
```

```
  serviceflow->setCID(dsa_rsp_frame->cid);
  data->set_serviceflow(serviceflow);
  ///

  /////Adicionado por claiton
  //printf("At %f a Mac %d solicita conexao\n", NOW, mac_->addr());
  //x12 = x12 + serviceflow->getMaximumSustainedTrafficRate ();
  banda = serviceflow->getMinimumReservedTrafficRate ();
  //teste = serviceflow->getSoliticacaoCAC();
  //banda = teste;
  fluxo_trafego = serviceflow->getScheduling ();


  //if (CAC (x12) == 0){
  teste2 = wimaxcac (banda, fluxo_trafego);
  if (teste2 == 0){
        printf("Em %f a solitacao feita pelo fluxo %d da MAC %d foi aceita\n", NOW,
        fluxo_trafego, mac_->addr());
          mac_->getCManager()->add_connection (data, dsa_rsp_frame->uplink);
                                  if (dsa_rsp_frame->uplink)
                                                  peer->setOutData (data);
                                                          else
                                                   peer->setInData (data);
                                  }

 //allocate ack
 //create packet for request
 ack = mac_->getPacket ();
 ch = HDR_CMN(ack);
 wimaxHdr_ack = HDR_MAC802_16(ack);
 ack->allocdata (sizeof (struct mac802_16_dsa_ack_frame));
 dsa_ack_frame = (mac802_16_dsa_ack_frame*) ack->accessdata();
 dsa_ack_frame->type = MAC_DSA_ACK;
 dsa_ack_frame->transaction_id = dsa_rsp_frame->transaction_id;
 dsa_ack_frame->uplink = dsa_rsp_frame->uplink;
 dsa_ack_frame->confirmation_code = 0; //OK
 ch->size() += DSA_ACK_SIZE;

 wimaxHdr_ack->header.cid = peer->getPrimary()->get_cid();
 peer->getPrimary()->enqueue (ack);
 }  /// fim if CAC

}
/***************************************************************************/

/**
 * process a flow request
 * @param p The received response
 */
void ServiceFlowHandler::processDSA_ack (Packet *p)
```

```
{
  mac_->debug ("At %f in Mac %d received DSA ack\n", NOW, mac_->addr());
}

/// Added by Aymen
  /**
   * Return The list of current flows
   * @return The list of current flows
   */
  struct serviceflow * ServiceFlowHandler::get_flow_head_()
  {
    return &flow_head_;
  }

/**
 * Send a flow request to the given node
 * @param index The node address
 * @param uplink The flow direction
 * @param serviceflow The service flow
 */

void ServiceFlowHandler::sendFlowRequest (int index, bool uplink, ServiceFlow*
serviceflow)
{
  /// Was token from code written by Richard
  Packet *p;
  struct hdr_cmn *ch;
  hdr_mac802_16 *wimaxHdr;
  mac802_16_dsa_req_frame *dsa_frame;
  PeerNode *peer;

  //create packet for request
  peer = mac_->getPeerNode(index);
  p = mac_->getPacket ();
  ch = HDR_CMN(p);
  wimaxHdr = HDR_MAC802_16(p);
  p->allocdata (sizeof (struct mac802_16_dsa_req_frame));
  dsa_frame = (mac802_16_dsa_req_frame*) p->accessdata();
  dsa_frame->type = MAC_DSA_REQ;
  dsa_frame->uplink = uplink;
  dsa_frame->transaction_id = TransactionID++;


  /// Added by Aymen
  // Add the service flow parameter to the dsa-req packet
  dsa_frame->serviceflow = serviceflow;


  /// Was token from code written by Richard
  if (mac_->getScheduler()->getNodeType()==STA_MN)
```

```
  ch->size() += GET_DSA_REQ_SIZE1;// (0); // change the data size
 else {
  //assign a CID and include it in the message
  Connection *data = new Connection (CONN_DATA);
  mac_->getCManager()->add_connection (data, uplink);
  if (uplink)
   peer->setInData (data);
  else
   peer->setOutData (data);
  dsa_frame->cid = data->get_cid();
  ch->size() += GET_DSA_REQ_SIZE1;// (1); // change the data size
 }

 wimaxHdr->header.cid = peer->getPrimary()->get_cid();
 peer->getPrimary()->enqueue (p);

}

/*************************** Adicionado por Claiton ************************/

/**
 * Mecanismo de CAC implementado por claiton
*/
int ServiceFlowHandler:: wimaxcac (int banda_solicitada, int fluxo){

      // UGS
                if (fluxo == 0){
                        if (banda_solicitada <= threshold-UGS){
                                threshold-UGS = threshold-UGS - banda_solicitada;
                                return 0;
                }
                else
                                return 1;
                }

      //rtPS
                if (fluxo == 1){
                         if (banda_solicitada <= threshold-rtps){
                                threshold-rtps = threshold-rtps - banda_solicitada;
                                return 0;
                }
                else
                                return 1;
                }

      //ntPS
                if (fluxo == 2){
                         if (banda_solicitada <= threshold-nrtps){
                                threshold-nrtps = threshold-nrtps - banda_solicitada;
                                return 0;
```

```
                }
                else
                        return 1;
                }

        //BE
                if (fluxo == 3){
                        return 0;
                }

}

/*****************************************************************************/
```

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo