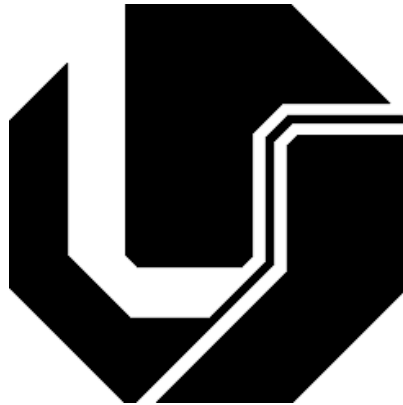


UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



ALGORITMO DE ESCALONAMENTO ADAPTATIVO PARA O TRÁFEGO
***UPLINK* EM REDES IEEE 802.16 COM GERENCIAMENTO DINÂMICO DE**
POLLING

Márcio Andrey Teixeira

Agosto

2012

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALGORITMO DE ESCALONAMENTO ADAPTATIVO PARA O TRÁFEGO
***UPLINK* EM REDES IEEE 802.16 COM GERENCIAMENTO DINÂMICO DE**
POLLING

Márcio Andrey Teixeira

Tese apresentada por Márcio Andrey Teixeira à
Universidade Federal de Uberlândia para obtenção do
título de Doutor em Ciências, avaliado em 09/08/2012
pela banca examinadora:

Paulo Roberto Guardieiro, Dr. (UFU) – Orientador

Juliana Freitag Borin, Dra. (UNICAMP)

Cesar Augusto Viana Melo, Dr. (UFAM)

Pedro Frosi Rosa, Dr. (UFU)

Éderson Rosa da Silva, Dr. (UFU)

Uberlândia (MG), Agosto de 2012.

**ALGORITMO DE ESCALONAMENTO ADAPTATIVO PARA O TRÁFEGO
UPLINK EM REDES IEEE 802.16 COM GERENCIAMENTO DINÂMICO DE
*POLLING***

Márcio Andrey Teixeira

Tese apresentada à Universidade Federal de Uberlândia para obtenção do título de
Doutor em Ciências.

Prof. Paulo Roberto Guardieiro, Dr.
Orientador

Prof. Alexandre Cardoso, Dr.
Coordenador do curso de Pós-Graduação

Dedicatória

Dedico este trabalho a todas as pessoas presentes em
minha vida e que tanto amo, cada uma com sua
parcela de contribuição, pelo carinho e incentivo
recíprocos em qualquer fase de minha vida.

Agradecimentos

Agradeço, em primeiro lugar, a Deus por me dar força, perseverança, teimosia e iluminar meu caminho, tornando possível a realização de mais essa etapa de minha vida.

Aos meus familiares por todo apoio, carinho e confiança irrestritos em todos os momentos. Especialmente, aos meus pais Dorival e Cleusa, por todas as lições e valores passados, que não foram poucos, e que sempre me auxiliaram e continuarão presentes comigo, além dos meus irmãos Marcos e Ana Paula, pela paciência e incentivo.

A minha esposa Aline, por todo o carinho, paciência e compreensão. Sempre me incentivando em todos os momentos da minha vida pessoal e profissional.

Ao meu filho Bruno, uma grande benção de Deus.

Ao prof. Dr. Paulo Roberto Guardieiro por toda orientação e experiência concedida, pela dedicação e incentivo para realização deste trabalho, e pelo exemplo de profissionalismo e conduta.

Aos colegas da Pós Graduação em Engenharia Elétrica da UFU (Universidade Federal de Uberlândia) pela colaboração no aprendizado.

Aos meus amigos pela amizade e companheirismo sempre presentes.

Resumo

Teixeira, Márcio. A. *Algoritmo de Escalonamento Adaptativo para o Tráfego Uplink em Redes IEEE 802.16 com Gerenciamento Dinâmico de Polling*, Uberlândia, Faculdade de Engenharia Elétrica – UFU, 2012.

A tecnologia *Worldwide Interoperability for Microwave Access* (WiMAX), baseada no padrão IEEE 802.16, é uma solução para redes de acesso sem fio de banda larga desenvolvida para dar suporte a uma grande variedade de aplicações de tempo real e não tempo real. Diferente das redes sem fio tradicionais, o padrão IEEE 802.16 define, na camada de controle de acesso ao meio, mecanismos para dar suporte à Qualidade de Serviço (*Quality of Service - QoS*) para as aplicações. Dentre tais mecanismos, destacam-se o escalonamento e o controle de admissão de conexões (*Connection Admission Control - CAC*). Entretanto, o padrão IEEE 802.16 não define as políticas que devem ser utilizadas na implementação de tais mecanismos.

O mecanismo de escalonamento tem como objetivo garantir a utilização eficiente do recurso largura de banda e, desta forma, promover o uso eficaz do enlace sem fio. O mecanismo de CAC tem como objetivo restringir o número de conexões existentes simultaneamente na rede, a fim de evitar que o enlace sem fio seja saturado.

Esta tese apresenta um novo e eficiente algoritmo de escalonamento para o tráfego *uplink*, a ser utilizado no escalonador *uplink* localizado na estação base (*Base Station – BS*). O algoritmo proposto foi desenvolvido para ser totalmente dinâmico, principalmente em redes que utilizam as funções de modulação adaptativa. Utilizando uma abordagem *cross-layer*, um esquema baseado em *deadlines* foi desenvolvido. Seu objetivo é minimizar o atraso máximo fim a fim para as aplicações de tempo real. Além disso, o algoritmo proposto interage com o

mecanismo de gerenciamento de *polling* da estação base, e controla a periodicidade do envio do *polling unicast* para as aplicações de tempo real e não tempo real, de acordo com os requisitos de QoS das aplicações. Ademais, para evitar que o enlace sem fio seja saturado por um número excessivo de conexões, desenvolveu-se um mecanismo de CAC que interage com o algoritmo de escalonamento proposto, o qual também utiliza a abordagem *cross-layer*. Resultados de simulação mostraram a eficiência do algoritmo de escalonamento proposto, bem como do mecanismo de CAC associado, principalmente em ambientes onde utilizou-se modulação adaptativa.

Palavras-chave: IEEE 802.16, WiMAX , QoS, Escalonamento.

Abstract

Teixeira, Márcio A. *An Adaptive Scheduler Algorithm for Uplink Traffic in WiMAX Networks with Dynamic Polling Management*, Uberlândia, Faculty of Electrical Engineering – UFU, 2012.

The Worldwide Interoperability for Microwave Access (WiMAX) technology, based on the IEEE 802.16 standard, is a solution for broadband wireless access metropolitan networks, developed to support a wide variability of real-time and non-real time applications. Different from the traditional wireless networks, the IEEE 802.16 standard defines, in the medium access layer, mechanisms to support the Quality of Service (QoS) for the applications. Among these mechanisms, we highlight the scheduling and the Connection Admission Control (CAC). However, the IEEE 802.16 does not define the policies that must be used in the implementation of the scheduling and CAC mechanisms.

The scheduling mechanism aims at guarantying the efficient utilization of the bandwidth resources, and thus, promotes the effective use of the wireless link. The CAC mechanism aims at restricting the number of existing connections simultaneously in order to avoid that the wireless link is saturated.

This thesis shows a new and efficient scheduling algorithm to uplink traffic in the Base Station (BS). The proposed algorithm is developed to be totally dynamic, mainly in networks that use adaptive modulation functions. Using a cross-layer approach, a deadline based scheme was developed, aiming at minimizing the end-to-end delay for the real-time applications. Moreover, the proposed algorithm interacts with the polling mechanism of the BS, and controls the periodicity of unicast polling to real-time and non-real-time applications, in accordance with the QoS requirements of the applications. Moreover, to avoid the wireless

link being saturated for an excessive number of connections, a CAC mechanism that interacts with the proposed scheduling algorithm was developed. The CAC mechanism was also developed using a cross-layer approach. Simulations results show the efficiency of the proposed scheduling algorithm and of the CAC mechanism, mainly in environments where an adaptive modulation was used.

Keywords: IEEE 802.16, WiMAX, QoS, Scheduling.

Publicações

A seguir são apresentadas as publicações resultantes das pesquisas realizadas no decorrer deste trabalho:

Artigos em periódicos

- TEIXEIRA, M. A & GUARDIEIRO, P. R. A new and efficient adaptive scheduling algorithm packets for the uplink traffic in WiMAX Networks. EURASIP Journal on Wireless Communications and Networking. September, 2011:113.
- TEIXEIRA, M. A & GUARDIEIRO, P. R. Adaptive Packet Scheduling for the Uplink Traffic in IEEE 802.16e Networks, International Journal of Communications Systems. January, 2012.

Capítulo de Livro

- TEIXEIRA, M. A & GUARDIEIRO, P. R. Scheduling Mechanims. Quality of Service and Resources Allocation in WiMAX, editado por Roberto Hincapie e Javier E. Sierra, Editora Intech, Fevereiro, 2012. ISBN 979-953-307-675-0. Disponível em: <http://www.intechopen.com/books/quality-of-service-and-resource-allocation-in-wimax/scheduling-mechanisms>.

Artigos em Congressos

- TEIXEIRA, M. A & GUARDIEIRO, P. R. Escalonamento de pacotes para o tráfego uplink em redes IEEE 802.16. 8th International Information and Telecommunication Technologies Symposium, I2TS 2009, Florianopolis, Santa Catarina, 09 – 11 December 2009.
- TEIXEIRA, M. A & GUARDIEIRO, P. R. A predictive scheduling algorithm for the uplink traffic in IEEE 802.16 networks. 12th International Conference On Advanced

Communications Technology, ICACT 2010, Phoenix Park, Korea, 7 – 10 February 2010, pp. 651 – 656.

- TEIXEIRA, M. A & GUARDIEIRO, P. R. Uplink scheduling algorithm with dynamic polling management in IEEE 802.16 broadband wireless networks. 25th Annual ACM Symposium on Applied Computing, SAC2010, Mobile Computer and Applications Track, Sierre, Switzerland, 22 – 26 March 2010.

- TEIXEIRA, M. A & GUARDIEIRO, P. R. Algoritmo de Escalonamento *Cross-layer* para o tráfego *Uplink* em redes IEEE 802.16. Aceito para ser publicado no XXIX Simpósio Brasileiro de Telecomunicações, SBrT'2011, Curitiba, Paraná, 02 – 05 Outubro 2011.

- TEIXEIRA, M. A & GUARDIEIRO, P. R. Integrated CAC and Scheduling Cross-layer Algorithms in WiMAX Networks with Dynamic Polling Management. IEEE Globecom Workshop: The 8th Broadband Wireless Access Workshop (BWA'2012), IEEE Globecom 2012, Anaheim, CA, USA, 134 – 139 December 2012.

Sumário

1. INTRODUÇÃO.....	25
1.1. Motivação.....	27
1.2. Objetivos.....	28
1.3. Contribuições da tese.....	29
1.4. Estrutura da tese.....	30
2. O PADRÃO IEEE 802.16 PARA REDES DE ACESSO DE BANDA LARGA SEM FIO	32
2.1. Introdução.....	32
2.2. O Padrão IEEE 802.16.....	33
2.3. Arquitetura de rede do padrão IEEE 802.16.....	35
2.3.1. Topologia.....	35
2.4. A Camada física.....	36
2.4.1. Subframe <i>downlink</i>	39
2.4.2. Subframe <i>uplink</i>	40
2.4.3. Modulação adaptativa.....	40
2.5. A Camada de acesso ao meio (MAC).....	43
2.5.1. Mecanismos para a provisão de qualidade de serviço.....	44
2.5.2. Mecanismos de requisição e alocação de banda.....	47
2.6. Considerações finais.....	49
3. MECANISMOS DE ESCALONAMENTO EM REDES IEEE 802.16.....	50
3.1. Introdução.....	50
3.2. O estado da arte dos algoritmos de escalonamento para redes IEEE 802.16.....	51

3.2.1. Classificação dos algoritmos de escalonamento.....	51
3.2.2. Algoritmos homogêneos.....	57
3.2.3. Algoritmos híbridos ou hierárquicos	59
3.2.4. Algoritmos <i>cross-layer</i>	64
3.2.5. Algoritmos de escalonamento que combinam CAC.....	68
3.3. Considerações finais	71
4. O CONTROLE DE ADMISSÃO DE CONEXÕES EM REDES IEEE 802.16.....	72
4.1. Introdução.....	72
4.2. O Mecanismo de CAC.....	73
4.3. Abordagens existentes sobre mecanismos de CAC.....	75
4.3.1. Mecanismos de CAC baseados em estratégia de degradação	75
4.3.2. Mecanismos de CAC que não utilizam estratégia de degradação	78
4.3.3. Mecanismos de CAC que utilizam abordagem <i>cross-layer</i>	81
4.4. Considerações finais.....	83
5. ALGORITMO DE ESCALONAMENTO ADAPTATIVO COM GERENCIAMENTO DINÂMICO DE <i>POLLING</i> PARA O TRÁFEGO <i>UPLINK</i> EM REDES IEEE 802.16 ...	84
5.1. Introdução.....	84
5.2. Desafios de projeto de algoritmos de escalonamento em redes IEEE 802.16.....	86
5.3. Definição do problema	87
5.4. Proposta de algoritmo de escalonamento adaptativo com gerenciamento dinâmico de <i>polling</i> para o tráfego <i>uplink</i> em redes IEEE 802.16.....	89
5.4.1. A arquitetura do escalonador proposto.....	89
5.4.1.1 Escalonamento da classe UGS	91
5.4.1.2 Escalonamento da classe ertPS.....	92

5.4.1.3 Escalonamento da classe rtPS	93
5.4.1.4. Escalonamento da classe nrtPS e BE.....	99
5.4.1.5. Gerenciamento dinâmico do <i>polling</i>	101
5.4.1.6. O mecanismo de CAC	103
5.5. Considerações finais.....	109
6. AVALIAÇÃO DA PROPOSTA DE ALGORITMO DE ESCALONAMENTO ADAPTATIVO COM GERENCIAMENTO DINÂMICO DE <i>POLLING</i> PARA O TRÁFEGO <i>UPLINK</i> EM REDES IEEE 802.16	110
6.1. Introdução.....	110
6.2. Modelagem e simulação	111
6.2.1. Características das fontes de tráfego utilizadas na simulação	113
6.3. Apresentação e análise dos resultados obtidos.....	114
6.3.1. Comparação entre os algoritmos de escalonamento RR, WRR e o algoritmo proposto	114
6.3.2. Comparação entre os algoritmos de escalonamento EDF e o algoritmo proposto	118
6.3.3. Comportamento das classes UGS e rtPS	121
6.3.4. Comportamento das classes UGS, rtPS e ertPS	124
6.3.5. Análise do comportamento do algoritmo de escalonamento proposto em função da MCS utilizada.....	125
6.3.6. Comparação entre o algoritmo de escalonamento proposto, e os algoritmos <i>cross-</i> <i>layers</i> mSIR e mmSIR.....	126
6.3.7. Análise do comportamento do algoritmo proposto para a classe nrtPS	127
6.3.8. Análise do comportamento do algoritmo proposto para as classes nrtPS e BE ...	129

6.3.9. Análise do comportamento da classe de serviço rtPS em função da utilização do mecanismo de CAC.....	130
6.3.10. Análise do comportamento das classes de serviço UGS, ertPS e rtPS em função da utilização do mecanismo de CAC.....	131
6.3.11. Análise do comportamento da classe de serviço nrtPS em função da utilização do mecanismo de CAC.....	133
6.4. Trabalhos relacionados.....	135
6.5. Considerações finais.....	139
7. CONCLUSÕES GERAIS E DESENVOLVIMENTOS FUTUROS.....	140
7.1. Conclusões.....	140
7.2. Desenvolvimentos futuros.....	142
REFERÊNCIAS BIBLIOGRÁFICAS.....	144

Lista de Figuras

Figura 2.1. Modelo de referência do padrão IEEE 802.16.....	33
Figura 2.2. Arquitetura de rede que utiliza o padrão IEEE 802.16.	35
Figura 2.3. Topologia PMP. As transmissões iniciam ou finalizam na BS.....	36
Figura 2.4. Topologia <i>Mesh</i> . A BS não é o centro da comunicação como na topologia PMP. As SSs podem se comunicar diretamente.....	36
Figura 2.5. Estrutura do Frame TDD.....	39
Figura 2.6. Formato do <i>subframe downlink</i>	39
Figura 2.7. Formato do <i>subframe uplink</i>	40
Figura 2.8. Esquema de modulação adaptativa adotada pelo WiMAX.....	41
Figura 2.9. Exemplo da utilização de limiar para a definição da MCS utilizada na transmissão. 42	
Figura 3.1. Classificação geral dos algoritmos de escalonamento	52
Figura 3.2. Classificação dos algoritmos de escalonamento definidas em [15].....	52
Figura 3.3. Classificação dos algoritmos de escalonamento definida em [16].....	53
Figura 3.4. Classificação dos algoritmos de escalonamento definida em [17].....	55
Figura 3.5. Estrutura de escalonamento hierárquico.	60
Figura 3.6. Estrutura de escalonamento utilizada em [27].	61
Figura 3.7. Estrutura de escalonamento utilizada em [29].	62
Figura 3.8. Estrutura de escalonamento utilizada em [30].	62
Figura 4.1. Estabelecimento de conexão definido pelo padrão IEEE 802.16.	74
Figura 5.1: Arquitetura de escalonamento proposta.....	90
Figura 5.2: Garantia de atraso e <i>jitter</i> limitados para a classe UGS.....	92
Figura 5.3: Algoritmo de escalonamento do serviço rtPS.....	95

Figura 5.4: Fluxograma do algoritmo de escalonamento rtPS.	97
Figura 5.5: Algoritmo do módulo de monitoramento de QoS.....	97
Figura 5.6: Algoritmo de ajuste do intervalo de <i>polling</i>	98
Figura 5.7: Atraso estimado vs. atraso amostrado.....	99
Figura 5.8: Algoritmo de escalonamento nrtPS.	99
Figura 5.9: Fluxograma do algoritmo de escalonamento nrtPS.	101
Figura 5.10: Largura de banda total do enlace <i>uplink</i>	104
Figura 5.11: Divisão da largura de banda.	105
Figura 5.12: Limiares definidos para as classes de serviço rtPS e nrtPS.	107
Figura 5.13: Algoritmo de CAC.....	108
Figura 6.1: O cenário utilizado na simulação.	111
Figura 6.2: Atraso médio vs. número de SSs com conexões rtPS.....	114
Figura 6.3: Atraso médio vs. número de SSs com conexões rtPS. Algoritmo utilizados RR, WRR e algoritmo proposto.	116
Figura 6.4: <i>Jitter</i> médio vs. número de SSs com conexões rtPS. Algoritmo utilizados RR, WRR e algoritmo proposto.	117
Figura 6.5: Vazão das classes nrtPS e BE com o algoritmo proposto vs. carga de tráfego rtPS. 118	
Figura 6.6: Atraso médio vs. número de SSs com uma MCS na rede de acesso.	119
Figura 6.7: Atraso médio vs. número de SSs (rtPS) com várias MCSs na rede de acesso. ...	120
Figura 6.8: Atraso médio das classes UGS e rtPS.....	121
Figura 6.9: Vazão média total dos serviços rtPS e UGS vs. a carga de tráfego rtPS.	122
Figura 6.10: Atraso médio dos serviços UGS e rtPS em função do número de SSs que geram tráfego rtPS.....	123

Figura 6.11: Atraso médio dos serviços UGS, rtPS, ertPS vs. o número de SSs que geram tráfego rtPS.....	124
Figura 6.12: Atraso médio em função da modulação utilizada.	125
Figura 6.13: Atraso médio vs. carga de tráfego rtPS.....	126
Figura 6.14: Vazão média total vs. carga de tráfego rtPS.	128
Figura 6.15: Vazão média das classes de serviço nrtPS e BE vs. a carga de tráfego nrtPS. ...	129
Figura 6.16: Atraso médio das conexões nrtPS e rtPS vs. a carga de tráfego nrtPS.	130
Figura 6.17: Atraso médio das conexões rtPS vs. o número de conexões rtPS.	131
Figura 6.18: Atraso médio das conexões rtPS vs. o aumento do número de conexões UGS e ertPS.	132
Figura 6.19: Vazão média total das conexões UGS, ertPS e rtPS vs. número de conexões rtPS.	133

Lista de Tabelas

Tabela 2.1. Taxas de transmissão na interface aérea em função da modulação, codificação e largura de canal.....	43
Tabela 2.2. Classes de Serviço e os parâmetros de QoS.	47
Tabela 3.1. Classificação dos algoritmos de escalonamento.....	56
Tabela 3.2. Comparação entre as categorias de algoritmos de escalonamento.	56
Tabela 3.3. Vantagens e desvantagens dos algoritmos discutidos na Seção 3.2.2.....	59
Tabela 3.4. Vantagens e desvantagens dos algoritmos discutidos na Seção 3.2.3.....	63
Tabela 3.5. Vantagens e desvantagens dos algoritmos discutidos na Seção 3.2.4.....	67
Tabela 3.6. Vantagens e desvantagens dos algoritmos discutidos na Seção 3.2.5.....	70
Tabela 4.1. Vantagens e desvantagens das propostas de CAC discutidas na Seção 4.3.1.....	77
Tabela 4.2. Vantagens e desvantagens das propostas de CAC discutidas na Seção 4.3.2.....	80
Tabela 4.3. Vantagens e desvantagens dos mecanismos de CAC discutidos na Seção 4.3.3...	82
Tabela 6.1. Níveis de modulação, codificação e SNR.....	112
Tabela 6.2 Principais parâmetros utilizados na simulação.	112

Lista de Abreviaturas

4G	Quarta Geração
AP	<i>Access Point</i>
ATM	<i>Asynchronous Transfer Mode</i>
BE	<i>Best Effort</i>
bps	bit por segundo
BR	<i>Bandwidth Request</i>
BS	<i>Base Station</i>
BW	<i>Bandwidth</i>
BWA	<i>Broadband Wireless Access</i>
b_{rtPS}	<i>bandwidth rtPS</i>
b_{ong}	<i>bandwidth ongoing</i>
b_{ugs}	<i>bandwidth UGS</i>
CAC	<i>Connection Admission Control</i>
CBR	<i>Constant Bit Rate</i>
CI	<i>CRC Indicator</i>
CID	<i>Connection IDentifier</i>
CPS	<i>Commom Part sublayer</i>
CRC	<i>Cyclic Redundancy Check</i>
CS	<i>Convegence Sublayer</i>
DFPQ	<i>Deficit Fair Priority Queue</i>
DFS	<i>Dynamic Frequency Selection</i>

DHCP	<i>Dynamic Host Configuration</i>
DL-MAP	<i>downlink map</i>
DSA	<i>Dinamic Service Addition</i>
DSL	<i>Digital Subscriber Lines</i>
E1	Nível Primário de Hierarquia Digital Plesiócrons Europeia
EC	<i>Encryption Control</i>
EDF	<i>Earliest Deadline First</i>
EKS	<i>Encryption Key Sequence</i>
ertPS	<i>extended real time Polling Service</i>
ETSI	<i>European Telecommunications Standards Institute</i>
FBWA	<i>Fixed Broadband Wireless Access</i>
FCH	<i>Frame Control Header</i>
FDD	<i>Frequency Division Duplexing</i>
FEC	<i>Forward Error Correction</i>
FIFO	<i>First In First Out</i>
FTP	<i>File Transfer Protocol</i>
Gbps	Gigabits por segundo
GHz	Gigahertz
GPC	<i>Grant Per Connection</i>
GPS	<i>Generalized Processor Sharing</i>
GPSS	<i>Grant Per Subscriber Station</i>
HCS	<i>Header Check Sequence</i>
HEC	<i>Header Error Check</i>
HT	<i>Header Type</i>

HTTP	<i>HyperText Transfer Protocol</i>
IE	<i>Information Element</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Internet Protocol</i>
IPv4	<i>Internet Protocol version 4</i>
ISP	<i>Internet Service Provider</i>
LAN	<i>Local Area Network</i>
L(rtPS)	<i>Limiar (rtPS)</i>
L(nrtPS)	<i>Limiar (nrtPS)</i>
LEN	<i>Lenght</i>
LL	<i>Link Layer</i>
LOS	<i>Line of Sight</i>
LTE	<i>Long Term Evolution</i>
kbps	Quilobit por segundo
MAC	<i>Medium Access Control</i>
Mbps	Megabit por segundo
MCS	<i>Modulation and Coding Scheme</i>
MIMO	<i>Multiple-Input Multiple-Output</i>
MMEP	<i>Média Móvel Exponencial Ponderada</i>
MPEG	<i>Motion Picture Expert Group</i>
MRTR	<i>Minimal Reserved Traffic Rate</i>
MSTR	<i>Maximum Sustained Traffic Rate</i>
ms	<i>milissegundo</i>
NDSL	<i>Networks & Distributed Systems Laboratory</i>

NLOS	<i>Non Line Of Sight</i>
nrtPS	<i>non-real time Polling Service</i>
NS-2	<i>Network Simulator-2</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
OFDMA	<i>Orthogonal Frequency Division Multiplexing Access</i>
PDA	<i>Portable Digital Assistants</i>
PDU	<i>Protocol Data Unit</i>
PHS	<i>Payload Header Suppression</i>
PHSI	<i>Payload Header Suppression Index</i>
PHY	<i>Physical Layer</i>
PM	<i>Poll-me</i>
PMP	<i>Point-to-Multipoint</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QoS	<i>Quality of Service</i>
QPSK	<i>Quadrature Phase Shift Keying</i>
RA	<i>Radio Access</i>
RLC	<i>Radio Link Control</i>
RR	<i>Round-Robin</i>
rtPS	<i>real time Polling Service</i>
SAP	<i>Service Access Point</i>
SC	<i>Single Carrier</i>
SCa	<i>Single Carrier access</i>
SDU	<i>Service Data Unit</i>
SFID	<i>Service Flow Identifier</i>

SNR	<i>Signal to Noise Ratio</i>
SS	<i>Subscriber Station</i>
T1	<i>Nível Primário de Hierarquia Digital Plesiócrona</i>
TDD	<i>Time Division Duplexing</i>
TDM	<i>Time Division Multiplexing</i>
TDMA	<i>Time Division Multiple Access</i>
UGS	<i>Unsolicited Grant Service</i>
UID	<i>Unique packet Identifier</i>
UL-MAP	<i>Uplink Map</i>
UNI	<i>User Network Interface</i>
VoIP	<i>Voice over Internet Protocol</i>
Wi-Fi	<i>Wireless Fidelity</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
WirelessHUMAN	<i>Wireless High-Speed Unlicensed Metropolitan Area Networks</i>
WLAN	<i>Wireless Local Area Network</i>
WMAN	<i>Wireless Metropolitan Area Network</i>
WRR	<i>Weighted Round-Robin</i>
xDSL	<i>Various Digital Subscriber Line Technologies</i>

Capítulo 1

INTRODUÇÃO

O desenvolvimento das tecnologias de redes sem fio tem se dado em ritmo acelerado nos últimos anos, principalmente no que diz respeito ao desenvolvimento das redes de banda larga sem fio. Tal desenvolvimento proporciona um cenário no qual usuários acessam a *Internet* a partir de dispositivos portáteis, a qualquer hora, em qualquer lugar, e executam diversos tipos de aplicações, incluindo aplicações multimídia de tempo real, que possuem uma variedade de requisitos de QoS.

Prover QoS em uma rede sem fio, sendo esta composta por estações móveis, nomádicas ou fixas, é atualmente um dos grandes desafios tecnológicos. Tais desafios são gerados a partir das próprias limitações do meio físico sem fio, como por exemplo, forte atenuação em função da distância, desvanecimento de Rayleigh, escalabilidade restrita, entre outros [1]. Por outro lado, novas aplicações estão surgindo cada vez mais exigentes, sendo necessário prover QoS nas redes sem fio de forma a garantir níveis de qualidade adequados e compatíveis a cada aplicação.

Assim sendo, o *Institute of Electrical and Electronics Engineers* (IEEE), que desenvolve padrões para atender diferentes tipos de redes sem fio, definiu o padrão IEEE 802.16 [2]. Este padrão define uma nova tecnologia de rede de banda larga sem fio, que tem como característica, aliar a flexibilidade existente nas redes sem fio, com a capacidade de realizar

transmissões que atendam aos requisitos das aplicações atuais, tais como: largura de banda mínima, atraso fim a fim e variação de atraso (*jitter*) máximo limitado. Todavia, este padrão não define como deve ser feito o escalonamento de recursos entre a estação base (*Base Station - BS*) e a estação cliente (*Subscriber Station - SS*), como também não define: o mecanismo de controle de admissão, um mecanismo eficiente de adaptação de enlace e um mecanismo eficiente para efetuar o gerenciamento de *polling*¹ para as estações clientes, sendo tais questões muito importantes na provisão da QoS.

Nesta tese propõe-se um algoritmo de escalonamento *uplink* para a provisão de QoS em redes baseadas no padrão IEEE 802.16. O algoritmo proposto interage com o mecanismo de gerenciamento de *polling* da BS e utiliza informações da camada física, mais precisamente, do mecanismo de adaptação de enlace, a fim de obter um melhor desempenho na rede de acesso e prover QoS. Existem vários trabalhos na literatura que tratam do escalonamento de pacotes como também do mecanismo de adaptação de enlace [3-6], entretanto, a maioria desses trabalhos considera tais mecanismos de forma isolada. Diferentemente desses trabalhos, o algoritmo proposto é aplicado diretamente nas filas de requisição de largura de banda da BS, e possui duas características fundamentais que o faz bastante eficiente num ambiente de comunicação sem fio:

- 1) Utiliza uma abordagem *cross-layer*: o algoritmo faz uso de informações oriundas da camada física sobre a modulação e a codificação que está sendo utilizada (*Modulation and Coding Schemes – MCSs*) e, com isso, calcula o atraso de transmissão com base nas mensagens de requisição de largura de banda (*Bandwidth Request - BR*) enviadas pelas SSs.
- 2) É preditivo: o algoritmo faz estimativas de atraso médio e monitora a quantidade de largura de banda que está sendo atribuída para as classes de serviço. Tais

¹ O *polling* é um processo pelo qual a BS aloca banda para as SSs realizarem suas requisições de banda. A requisição de banda é necessária para todos os tipos de serviços, exceto para o serviço UGS (*Unsolicited Grant Service*).

informações são utilizadas pelo escalonador, tendo como objetivo atender os requisitos de QoS das aplicações, bem como distribuir, de maneira justa e eficiente, os recursos disponíveis entre as várias conexões existentes.

Além disso, para evitar que o enlace sem fio seja saturado por um número excessivo de conexões, desenvolveu-se um mecanismo de CAC que interage com o algoritmo de escalonamento proposto. Tal mecanismo também foi desenvolvido utilizando uma abordagem *cross-layer*. Quando uma SS tenta estabelecer uma conexão, e assim, definir um tipo de serviço, a mesma envia para a BS uma mensagem denominada *Dynamic Service Addition* (DSA). O mecanismo de CAC, uma vez recebendo a mensagem DSA, verifica o requisito de largura de banda (para aplicações de tempo real e não tempo real) e o requisito de atraso máximo limitado (para as aplicações de tempo real). Baseando-se na disponibilidade dos recursos, o mecanismo de CAC verifica a possibilidade de aceitar ou não tal pedido de conexão.

1.1. Motivação

O padrão IEEE 802.16 é considerado atualmente como uma das tecnologias mais promissoras para oferecer comunicação de rede de banda larga sem fio para ambientes metropolitanos, bem como suburbanos e rurais. Diferente de outras tecnologias de redes sem fio, o padrão IEEE 802.16 foi desenvolvido com mecanismos incorporados na camada MAC para as aplicações com requisitos de QoS. Assim sendo, uma das maiores vantagens dessa tecnologia é a capacidade de viabilizar a transmissão de uma grande variedade de dados, com diferentes características de tráfego, mas com garantias de QoS. Além disso, o padrão proporciona conexão de banda larga em regiões onde não existe infraestrutura de cabeamento telefônico ou de TV a cabo, tornando-se uma ótima solução para a chamada última milha.

Uma das principais questões deixadas em aberto pelo padrão IEEE 802.16 é o escalonamento. As SSs ativas na rede compartilham um mesmo canal de comunicação *uplink*,

sendo de responsabilidade da BS alocar recursos e decidir em que momento uma dada SS terá direito de transmitir no canal de comunicação. O algoritmo de escalonamento, que é responsável por esse controle, não foi definido pelo padrão, sendo tal algoritmo de fundamental importância para a provisão de QoS para as aplicações.

Outra questão importante, que deve ser considerada na provisão de QoS, é o controle do número de conexões existente na rede. Uma conexão somente deve ser admitida na rede se os requisitos de QoS, definidos para tal conexão, puderem ser garantidos. Este é o papel fundamental do mecanismo de CAC. Entretanto, o padrão também não define o algoritmo de CAC.

As questões anteriormente descritas estão abertas para pesquisas e desenvolvimento, uma vez que o processo de alocação de banda e o controle de admissão de conexões são pontos fundamentais para a provisão de QoS na camada MAC. Além disso, os sistemas de comunicação sem fio utilizam, em geral, canais cujas características de propagação variam com o tempo. Nesse contexto, é de fundamental importância o desenvolvimento de algoritmos de escalonamento *cross-layer*, pois esses algoritmos podem utilizar informações de estado do canal de comunicação no escalonamento. A utilização desses algoritmos possibilita uma melhor utilização dos recursos em ambientes de comunicação sem fio. Tal questão é assunto de pesquisas e desenvolvimento, pois grande parte dos trabalhos que envolvem escalonamento assume condições ideais de propagação, o que não acontece num ambiente de comunicação real.

1.2. Objetivos

Esta tese tem por objetivo o desenvolvimento de um algoritmo de escalonamento *cross-layer* para o tráfego *uplink*. O algoritmo de escalonamento deverá garantir a QoS para as aplicações de tempo real, e também os recursos mínimos para aplicações não tempo real. Este algoritmo leva em consideração a modulação utilizada pelas SSs, visando utilizar o enlace

sem fio de forma eficiente. Além disso, o algoritmo de escalonamento interage com um mecanismo de CAC que também utiliza uma abordagem *cross-layer*. Para atingir tais objetivos, as seguintes ações foram executadas:

- Definição da política a ser utilizada no mecanismo de escalonamento. São várias as políticas de escalonamento existentes, sendo estas classificadas como políticas tradicionais, otimistas e *cross-layer*;

- Definição da arquitetura de escalonamento. Uma vez definida a política de escalonamento, elaborou-se uma arquitetura de escalonamento que permitiu integrar as classes de serviço definidas pelo padrão com o mecanismo de escalonamento;

- Definição do algoritmo de gerenciamento do *polling*. O presente trabalho utiliza um mecanismo de *polling* adaptativo;

- Definição do algoritmo de CAC. Tal algoritmo interage com o algoritmo de escalonamento proposto;

- Apresentação da proposta do algoritmo de escalonamento;

- Apresentação do algoritmo de gerenciamento de *polling* e também do algoritmo de CAC;

- Avaliação do algoritmo de escalonamento proposto em vários cenários e desafios, além de compará-lo com outras abordagens, a fim de demonstrar sua viabilidade;

- Avaliação do mecanismo de CAC em conjunto com o algoritmo de escalonamento proposto.

1.3. Contribuições da tese

O desenvolvimento deste trabalho tem como principais contribuições os seguintes itens:

- A proposição e o desenvolvimento do algoritmo de escalonamento para o tráfego *uplink*, bem como seu estudo baseado em modelagem e simulação por meio da ferramenta de simulação NS-2 [7];

- O estudo da interação do algoritmo de escalonamento com o mecanismo de gerenciamento de *polling* da BS. O mecanismo de *polling* da BS, mais precisamente o parâmetro intervalo de *polling*, é gerenciado pelo mecanismo de escalonamento, sendo esta uma questão completamente nova. A pesquisa bibliográfica realizada demonstrou que, até o presente momento, o gerenciamento do mecanismo de *polling* não foi considerado em outros trabalhos, e os intervalos de envio de *polling* são considerados estáticos;

- A definição do módulo de monitoramento de QoS da BS. Tal módulo faz estimativas sobre o atraso médio para as aplicações multimídia e também armazena informações sobre a quantidade de largura de banda atribuída para as classes de serviço;

- A definição de um algoritmo de CAC que utiliza uma abordagem *cross-layer*. Tal algoritmo interage com o algoritmo de escalonamento proposto.

1.4. Estrutura da tese

O Capítulo 2 apresenta os conceitos sobre a tecnologia de rede de acesso de banda larga sem fio, tendo como foco principal o padrão IEEE 802.16. Nesse capítulo são descritas as principais características do padrão IEEE 802.16, tais como a sua arquitetura e o seu funcionamento. Além disso, questões consideradas importantes para a obtenção da QoS são descritas.

O Capítulo 3 apresenta um levantamento dos principais algoritmos de escalonamento para redes IEEE 802.16 existentes na literatura.

O Capítulo 4 contém um resumo dos principais mecanismos de CAC para redes IEEE 802.16 existentes na literatura.

O Capítulo 5 descreve o algoritmo de escalonamento proposto para o tráfego *uplink* em redes IEEE 802.16.

O Capítulo 6 analisa o algoritmo de escalonamento proposto utilizando o recurso de modelagem e simulação.

O Capítulo 7 traz as considerações finais do trabalho como também as considerações sobre trabalhos futuros.

Capítulo 2

O PADRÃO IEEE 802.16 PARA REDES DE ACESSO DE BANDA LARGA SEM FIO

2.1. Introdução

O padrão IEEE 802.16 [2], comercialmente conhecido como WiMAX, apresenta uma solução promissora para as redes de acesso de banda larga sem fio [8][9], podendo ser utilizada como uma plataforma alternativa para a criação de redes MANs (*Metropolitan Area Networks*) ou WANs (*Wide Area Networks*) sem fio. Assim sendo, além de ser uma tecnologia de rede de acesso sem fio, o padrão IEEE 802.16 pode ser utilizado na construção de redes de dados, públicas ou privadas, em áreas abrangentes, competindo assim, com as tradicionais redes de acesso via cabo, tais como o DSL (*Digital Subscriber Line*) ou cable modem. Além disso, a tecnologia WiMAX pode ser utilizada como uma solução para a questão da chamada “última milha”, pois esta tecnologia pode ser adotada, rapidamente, em muitas áreas não servidas por banda larga cabeada.

Uma das grandes vantagens do padrão IEEE 802.16 é a especificação de mecanismos para provisão da QoS. Conforme será visto na Seção 2.5.1, o padrão define os meios para fazer a diferenciação e a priorização do tráfego, objetivando garantir os requisitos de QoS, tais como o atraso máximo limitado, a largura de banda mínima requerida. Este capítulo tem

como objetivo apresentar as principais características e o funcionamento do padrão IEEE 802.16.

2.2. O Padrão IEEE 802.16

O padrão IEEE 802.16 define as características da camada física (PHY) e as características da camada de controle de acesso ao meio (MAC) para as redes metropolitanas de banda larga sem fio. As características da camada física (PHY) são descritas na Seção 2.4, e as características da camada MAC são descritas na Seção 2.5. A Figura 2.1 ilustra o modelo de referência do padrão IEEE 802.16.

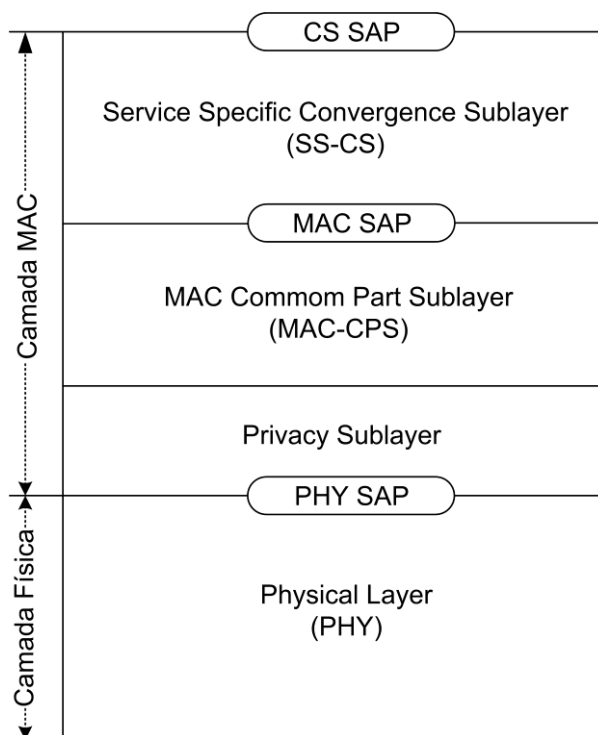


Figura 2.1. Modelo de referência do padrão IEEE 802.16 [2].

Como pode ser visto na Figura 2.1, a camada MAC é subdividida em três subcamadas distintas:

- **Service Specific Convergence Sublayer (SS-CS)**: a principal função da SS-CS é de fazer o mapeamento do tráfego proveniente da camada superior para a camada MAC. A SS-CS recebe as PDUs (*Protocol Data Unit*) da camada superior, realiza sua classificação e as

envia ao SAP (*Service Access Point*) apropriado, transformando os dados provenientes da rede externa em MAC SDUs (*Service Data Unit*). Durante a classificação, as SDUs são associadas a conexões. Cada conexão possui um identificador único denominado como CID (*Connection Identifier*), sendo este, relacionado com um determinado fluxo de serviço, identificado como SFID (*Service Flow Identifier*). O padrão especifica dois tipos de subcamada CS: *Packet CS*, definida para mapear serviços baseados em pacotes, tais como, o IPv4, IPv6 e a Ethernet; *ATM CS*, definida para serviços ATM;

– **MAC Common Part Sublayer (MAC CPS)**: a MAC CPS é responsável pelas funcionalidades comuns de acesso ao sistema, como também pela provisão da QoS. Fazem parte de tais funcionalidades: o estabelecimento e o gerenciamento das conexões entre a BS e a SS, o gerenciamento da largura de banda, o escalonamento dos fluxos de serviço, entre outros etc. A subcamada MAC CPS recebe as SDUs provenientes da camada superior e as encapsula em MAC PDUs, que serão enviadas para o destinatário. A MAC PDU é a unidade de dados trocada entre a MAC da BS e a MAC das SSs. A MAC PDU consiste em um cabeçalho de tamanho fixo, um campo de tamanho variável para dados (*payload*), e um campo CRC (*Cycle Redundancy Check*) opcional. As MAC PDUs são classificadas em conexões apropriadas, as quais são mapeadas em classes de serviço. Cada classe de serviço define seu conjunto de parâmetros de QoS, tais como, o atraso, a vazão e o *jitter*. O padrão IEEE 802.16 especifica cinco classes de serviço as quais serão descritas na Seção 2.5.1;

– **Security sublayer**: provê funcionalidades de segurança e privacidade dos dados. Esta subcamada é responsável pela autenticação para o acesso à rede, troca de chaves de segurança, criptografia dos dados dentre outras funcionalidades de segurança.

2.3. Arquitetura de rede do padrão IEEE 802.16

A arquitetura de rede, definida pelo padrão IEEE 802.16, é basicamente composta por dois elementos: BS e SS. A Figura 2.2 ilustra a arquitetura de uma rede que utiliza o padrão IEEE 802.16. A BS é o elemento responsável por fazer a interface entre uma infraestrutura de rede, como por exemplo, um ISP (*Internet Server Provider*), com a SS, possibilitando uma extensão dos serviços de Internet para seus usuários. A SS se conecta com a BS, tendo como objetivo oferecer serviços diferenciados para os usuários.

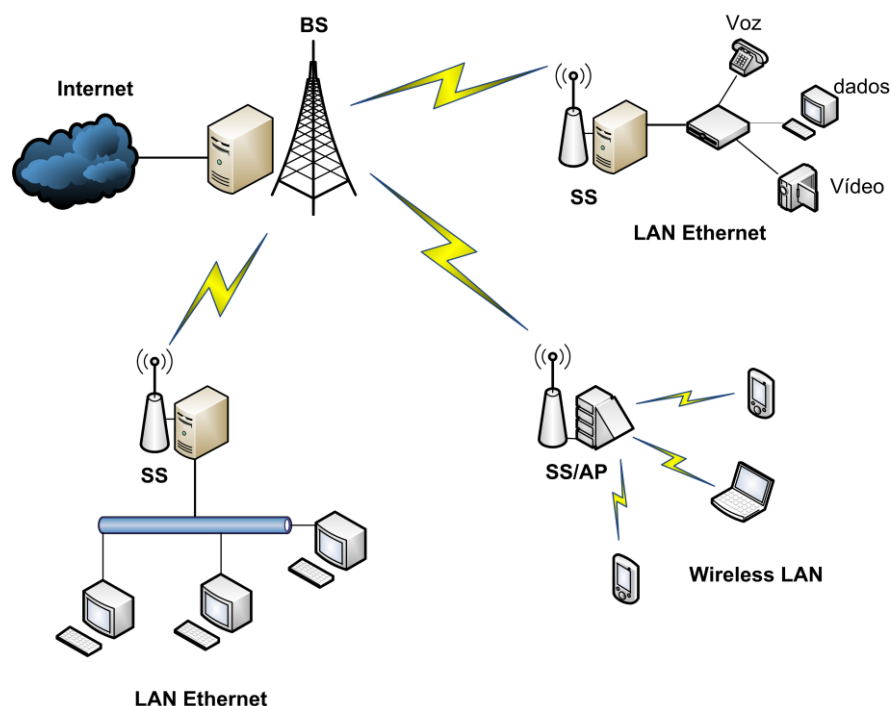


Figura 2.2. Arquitetura de rede que utiliza o padrão IEEE 802.16.

2.3.1. Topologia

O padrão IEEE 802.16 define dois tipos de topologias de comunicação [10][11]:

- **Ponto-multi-ponto** (*Point-to-Multipoint - PMP*): nesta topologia, a BS tem a função de coordenar e retransmitir todo o tráfego entre as SSs da rede. Toda e qualquer SS tem a obrigatoriedade de comunicar-se com a BS para transmitir qualquer tipo de dado para outra SS. A Figura 2.3 ilustra a topologia PMP;

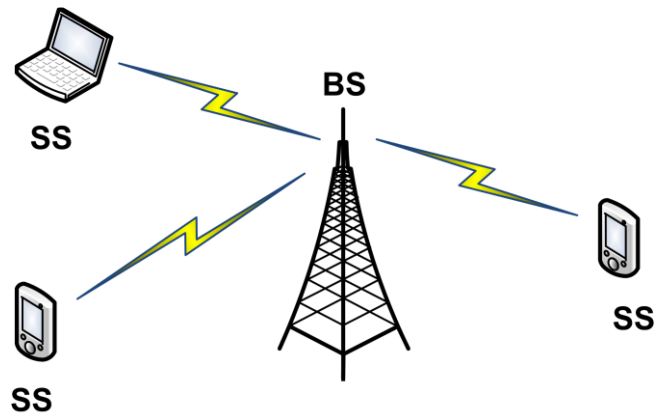


Figura 2.3. Topologia PMP. As transmissões iniciam ou finalizam na BS.

- **Topologia Mesh:** os nós são organizados em forma *ad-hoc*, isto é, todas as SSs são pares (*peers*), e cada par pode atuar como um roteador para repassar pacotes dos seus nós vizinhos. Neste caso, cada SS pode comunicar-se diretamente com as SSs vizinhas, sem qualquer coordenação com a BS. A Figura 2.4 ilustra a topologia *Mesh*;

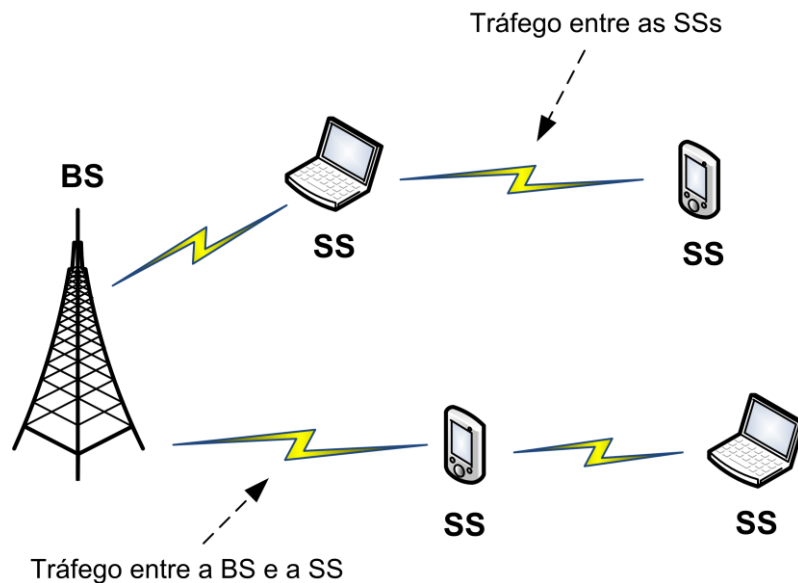


Figura 2.4. Topologia *Mesh*. A BS não é o centro da comunicação como na topologia PMP. As SSs podem se comunicar diretamente.

2.4. A Camada física

A camada física (PHY) segue as especificações do padrão IEEE 802.16, onde define-se: a estrutura de multiplexação, as técnicas de modulação, a sincronização entre os transmissores

e os receptores, as técnicas de correção de erro e a definição do espectro de frequência etc. Atualmente, o padrão IEEE 802.16 especifica cinco interfaces aéreas: uma desenvolvida para atuar com linha de visada (*Line of Sight - LOS*), operando nas frequências de 10-66 GHz, e quatro para atuar sem linha de visada (*Non Line of Sight - NLOS*), operando nas frequências de 2-11 GHz. Uma breve descrição das interfaces aéreas é feita a seguir:

- WirelessMAN-SC: utiliza a técnica de transmissão *Single Carrier*. Suporta técnicas de duplexação TDD e FDD. Esta interface é utilizada nas transmissões com linha de visada nas frequências de 10-66 GHz;
- WirelessMAN-SCa: utiliza o formato de portadora única (*SCa – Single Carrier access*) para frequências de até 11 GHz. Possui funcionalidades para suportar operações sem linha de visada. Dentre tais operações destacam-se: modulação adaptativa, múltiplos esquemas de codificação, estrutura de quadro robusta a multipercurso, diversidade de transmissão, *Automatic Repeat Request (ARQ)* e controle de potência;
- WirelessMAN-OFDM: utiliza modulação OFDM (*Orthogonal Frequency Division Multiplexing*) sem linha de visada direta (multiportadoras para frequências abaixo de 11 GHz). Suporta topologia *Mesh* e subcanalização no enlace *uplink* (da SS para BS);
- WirelessHUMAN: utilizada em bandas não licenciadas nas faixas de frequência de 5 e 6 GHz, com seleção dinâmica de frequências (*DFS – Dynamic Frequency Selection*);
- WirelessMAN-OFDMA: utiliza modulação OFDM com acesso múltiplo (OFDMA - *Orthogonal Frequency Division Multiplexing Access*) com uma transformada de 2048 subportadoras. O acesso múltiplo é disponibilizado por meio de um subconjunto de endereçamento de múltiplas portadoras para

destinatários individuais [11]. Suporta operações sem linha de visada em frequências abaixo de 11 GHz. Permite subcanalização nos canais *uplink* e *downlink* (da BS para a SS).

O padrão especifica dois modos de acesso ao meio físico: duplexação por divisão de frequência (*Frequency Division Duplexing - FDD*) e duplexação por divisão no tempo (*Time Division Duplexing - TDD*). O modo FDD opera simultaneamente com dois canais distintos, *downlink* e *uplink*. Já no modo TDD, os dados compartilham um mesmo canal de transmissão (mesma frequência), tanto no sentido *uplink* quanto no sentido *downlink*, porém, os dados são transmitidos em tempos diferentes.

Os fluxos de bits da camada física são estruturados em sequências de *frames* (quadros) de tamanhos iguais, onde os mesmos são transmitidos no sentido *uplink* e no sentido *downlink*. Os *frames* são constituídos por *slots físicos* (*Physical Slots - PS*). O número de *slots* físicos existente em um *frame* é uma função da taxa de símbolos e da duração do *frame* [2]. A duração de um *frame* pode variar de 2,5 a 20 ms, dependendo da tecnologia em uso na camada física.

Cada *frame* é dividido em dois *subframes*: um *subframe downlink* e um *subframe uplink*. O *subframe downlink* é utilizado pela BS para enviar dados e informações de controle para as SSs. O *subframe uplink* é utilizado pelas SSs para enviar dados e mensagens de requisição de largura de banda para a BS. Portanto, o *subframe uplink* é compartilhado entre todas as SSs. A Figura 2.5 ilustra a estrutura do *frame* TDD.

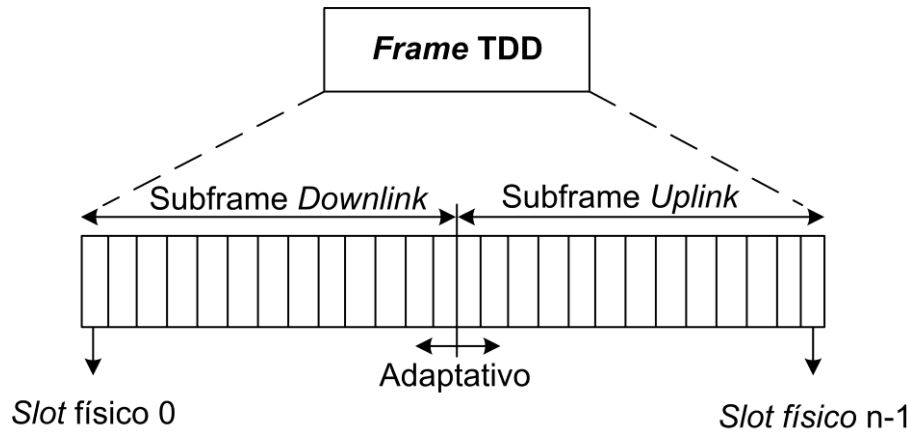


Figura 2.5. Estrutura do Frame TDD [2].

2.4.1. Subframe *downlink*

A transmissão no canal *downlink* é realizada em *broadcast* (ou difusão total). Desta forma, na comunicação *downlink*, somente a BS transmite, fazendo com que todas as SSs recebam os dados, mas uma SS apenas acessa os dados a que pertencem a ela. A Figura 2.6 ilustra o formato do *subframe downlink*.

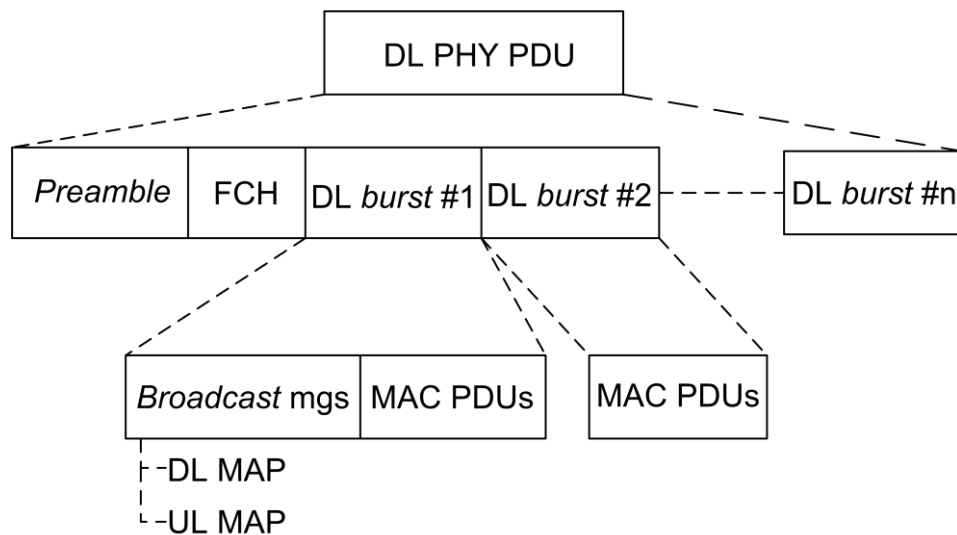


Figura 2.6. Formato do *subframe downlink* [2].

Como pode ser visto na Figura 2.6, o *subframe downlink* é iniciado com um preâmbulo, cuja função é realizar a sincronização da transmissão. Em seguida, tem início o FCH, onde são inseridas as informações de controle de transmissão DL-MAP e UL-MAP. A mensagem

DL-MAP define a transmissão no sentido *downlink*. Já a mensagem UL-MAP contém informações referentes às oportunidades de transmissão das SSs no sentido *uplink*.

2.4.2. Subframe *uplink*

O principal objetivo do canal *uplink* é fazer com que as SSs possam se comunicar com a BS para enviar seus dados. A estrutura de dados utilizada para o tráfego *uplink* é chamada de *subframe uplink*. A Figura 2.7 apresenta o formato do *subframe uplink*.

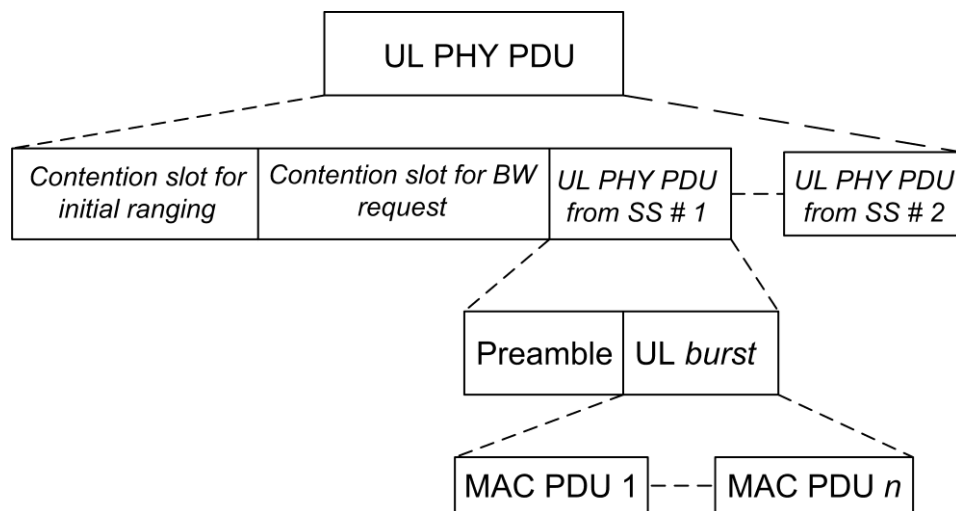


Figura 2.7. Formato do *subframe uplink* [2].

A primeira parte do *subframe uplink* corresponde aos intervalos de tempo (*time slots*) reservados para a manutenção inicial do sistema. Em seguida, inicia-se o período de contenção, no qual as SSs podem realizar solicitações de banda para realizar futuras transmissões no sentido *uplink*. A última parte do *subframe uplink* é composta pelos intervalos de transmissão das SSs, os quais foram previamente escalonados pela BS.

2.4.3. Modulação adaptativa

Diferentemente das tecnologias de redes sem fio tradicionais, o padrão IEEE 802.16 possui a capacidade de escolher, dinamicamente, dentre as técnicas de modulação definidas, a mais adequada. Em condições ruins de propagação ou enlaces muito longos, a BS aplica uma

técnica de modulação e codificação mais robusta (com menor quantidade de bits/símbolo), garantindo uma comunicação mais estável, porém, com uma taxa de transmissão mais reduzida. Em curtas distâncias, outras técnicas de modulação de alta eficiência espectral são utilizadas (maior quantidade de bits/símbolo), para garantir taxas de transmissão mais elevadas. Assim sendo, a técnica de modulação confere maior robustez e flexibilidade ao sistema.

A partir de negociação entre a BS e as SSs, a modulação a ser adotada é dinamicamente adaptada às condições do enlace de rádio. Por exemplo, utilizando informações de *feedback* do indicador de qualidade do canal, a SS pode prover para a BS informações sobre o canal de *downlink*. Já para o canal de *uplink*, a BS é capaz de fazer estimativas sobre o estado do canal baseando-se na qualidade do sinal recebido. A partir destas informações, a BS irá ajustar a modulação/codificação, tendo como objetivo, fazer um melhor uso do enlace de rádio. A Figura 2.8 ilustra o esquema de modulação adaptativa adotada pelo WiMAX.

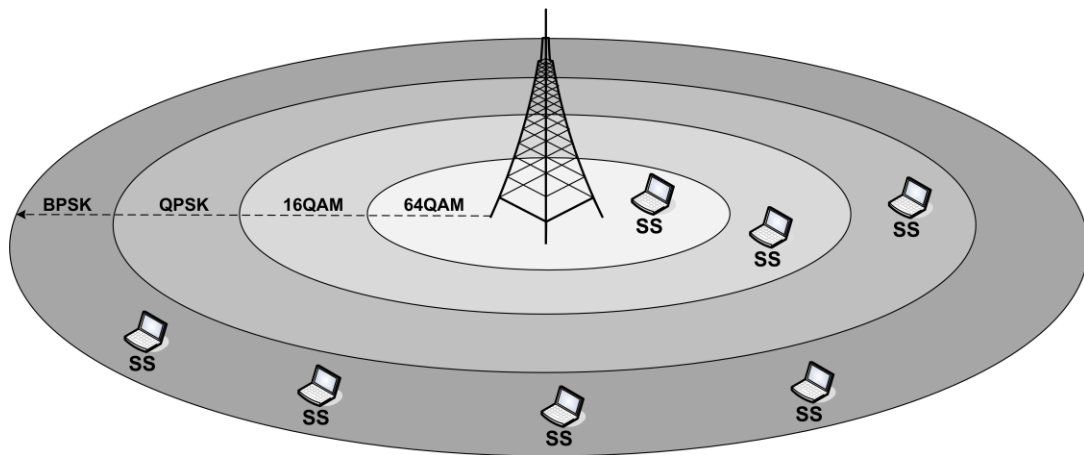


Figura 2.8. Esquema de modulação adaptativa adotada pelo WiMAX.

A BS executa a adaptação do enlace comparando o valor da relação sinal/ruído (SNR – *Signal-to-Noise Ratio*) das SSs com limiares, a fim de selecionar o esquema de modulação/codificação (*Modulation and Coding Scheme* – *MCS*) que será utilizada. Existem dois tipos de limiares:

- Limiar mínimo de entrada: representa o valor mínimo de SNR para iniciar a comunicação utilizando uma MCS mais eficiente;
- Limiar obrigatório de saída: representa o SNR abaixo do qual a MCS atual não pode ser utilizada, sendo necessário utilizar uma MCS mais robusta.

A Figura 2.9 ilustra a utilização do limiar para a definição da MCS que será utilizada em uma determinada transmissão.

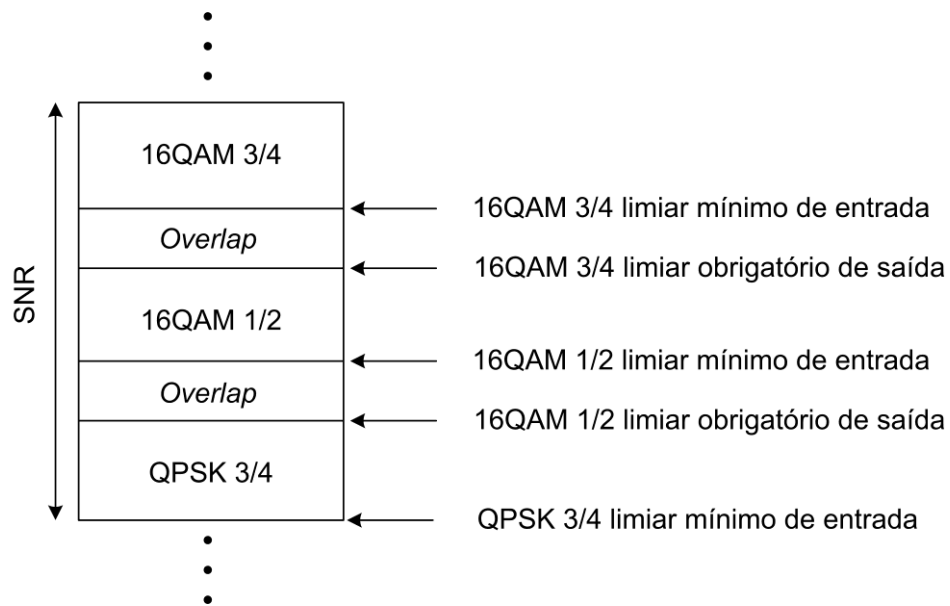


Figura 2.9. Exemplo da utilização de limiar para a definição da MCS utilizada na transmissão.

Quanto maior for a distância entre os pares BS e SS, menor será a taxa de transmissão. Para SSs próximas da BS, utiliza-se a modulação 64QAM (*Quadrature Amplitude Modulation*) que transmite a uma taxa de seis *bits/ baud*. Para SSs localizadas a uma distância média da BS, usa-se a modulação 16QAM que transmite a uma taxa de quatro *bits/ baud*. Para SSs mais distantes da BS, utiliza-se o QPSK (*Quadrature Phase Shift Keying*) que transmite a uma taxa de dois *bits/ baud* ou utiliza-se o BPSK (*Binary Phase Shift Keying*) que transmite a uma taxa de um *bit/ baud*. Assim sendo, existe uma grande variação na taxa de transmissão que irá depender diretamente da modulação utilizada, da taxa de codificação e da largura de banda de canal empregada. A Tabela 2.1 ilustra o resultado do esquema de modulação

adaptativa, na qual é possível observar a taxa de transmissão na interface aérea em função da modulação, codificação e largura de canal.

Tabela 2.1. Taxas de transmissão na interface aérea em função da modulação, codificação e largura de canal [12].

Modulação	64QAM		16QAM		QPSK		BPSK
Taxa de Codificação	3/4	2/3	3/4	1/2	3/4	1/2	1/2
Largura do Canal							
1,75 MHz	6,55	5,82	4,36	2,91	2,18	1,45	0,73
3,5 MHz	13,09	11,64	8,73	5,82	4,36	2,91	1,45
5 MHz	18,70	16,62	12,47	8,31	6,23	4,16	2,08
7 MHz	26,18	23,27	17,45	11,64	8,73	5,82	2,91
10 MHz	37,40	33,25	24,94	16,62	12,47	8,31	4,16
20 MHz	74,81	66,49	49,87	33,25	24,94	16,62	8,32

2.5. A Camada de acesso ao meio (MAC)

A camada MAC do padrão IEEE 802.16 tem como principal função fazer o controle de acesso ao meio (determinar quais estações podem acessar a rede) e garantir a QoS. O padrão IEEE 802.16 define na camada MAC mecanismos de alocação dinâmica de recursos e atribuição de prioridades de tráfego para tentar prover QoS. Como mostrado na Seção 2.2, a camada MAC é subdividida em três subcamadas. Entretanto, esta seção tem seu foco voltado para a subcamada MAC CPS, dado que as questões sobre a camada SS-CS e também as questões de segurança não fazem parte do escopo desta tese.

A subcamada MAC CPS é responsável por executar funcionalidades importantes, tais como o estabelecimento e o gerenciamento das conexões realizadas entre a BS e a SS, o suporte a QoS e o gerenciamento de largura de banda. Os principais serviços fornecidos por esta subcamada são:

- gerenciar o acesso das SSs no sentido *downlink*;
- gerenciar o acesso das SSs no sentido *uplink*;
- prover técnicas de QoS para conexões MAC;

- definir tipos de conexões, classes de fluxos de serviço e a estrutura dos quadros MAC;
- realizar o escalonamento dos fluxos de serviço e o controle de admissão de conexões (CAC).

A seguir são descritos os principais mecanismos utilizados para a provisão da QoS definidos pelo padrão IEEE 802.16.

2.5.1. Mecanismos para a provisão de qualidade de serviço

O padrão IEEE 802.16 provê mecanismos na camada MAC para fornecer QoS aos tráfegos *downlink* e *uplink*. Tais mecanismos estão relacionados aos seguintes conceitos: classificação de pacotes, estabelecimento de serviço e escalonamento por fluxo de serviço. Os pacotes que passam pela camada MAC são classificados e associados a um fluxo de serviço. O fluxo de serviço é um serviço da camada MAC que fornece transporte unidirecional aos pacotes. O fluxo de serviço é criado durante a fase de estabelecimento da conexão entre a BS e a SS. Cada fluxo de serviço possui um identificador único, sendo este relacionado com uma determinada classe de serviço. As classes de serviço definem um conjunto de parâmetros de QoS. A seguir é feita uma breve descrição de alguns parâmetros de QoS:

- *maximum sustained traffic rate* (MSTR): define a taxa de pico de informação de um determinado serviço. Essa taxa é informada em bits por segundo;
- *minimum reserved traffic rate* (MRTR): especifica a taxa mínima reservada para o fluxo de serviço. Essa taxa é informada em bits por segundo. Se este parâmetro for omitido, o valor padrão é zero, ou seja, não será garantida nenhuma largura de banda;

- *maximum latency* (ML): especifica o atraso máximo entre a recepção do pacote na interface da rede da BS ou da SS e a transmissão do pacote para a interface aérea;
- *tolerated jitter* (TJ): especifica a variação máxima permitida para o atraso da conexão;
- *traffic priority* (TP): o valor desse parâmetro especifica a prioridade atribuída para o fluxo de serviço. Dados dois fluxos de serviço idênticos em todos os parâmetros de QoS exceto na prioridade, o fluxo de serviço com maior prioridade deve ter atraso menor;
- *service flow scheduling* (SFS): o valor desse parâmetro determina o tipo de serviço ao qual o fluxo deve ser associado;
- *request/transmission policy* (R/TP): este parâmetro provê a capacidade de especificar certos atributos para o fluxo de serviço. Esses atributos incluem opções sobre a formatação da PDU e restrições sobre os tipos de requisição de banda que podem ser utilizado.

O padrão IEEE 802.16 define cinco classes de serviço. Cada classe de serviço é associada a um determinado conjunto de parâmetros de QoS. O escalonador *uplink* da BS deve alocar largura de banda para as SSs de acordo com o conjunto de regras definido para cada classe serviço.

O serviço UGS (*Unsolicited Grant Service*) suporta tráfegos de tempo real com taxa de transmissão constante (*CBR – Constant Bit Rate*), os quais requerem alocação fixa de banda. As conexões UGS recebem *grants*² periódicos de tamanho fixo sem a necessidade de requisitar largura de banda, eliminando a sobrecarga e o atraso da requisição de banda. Os

² Garantia de oportunidade de transmissão.

parâmetros de serviço dos fluxos UGS são negociados uma única vez, no momento em que a conexão é estabelecida. A partir de então, a banda concedida é garantida até o final da conexão.

O serviço rtPS (*real-time Polling Service*) é destinado às aplicações de tempo real com taxa de transmissão variável (*VBR – Variable Bit Rate*), como por exemplo, vídeo MPEG, teleconferência, jogos interativos, entre outros. Nesta classe, a SS recebe em intervalos periódicos, a oportunidade de solicitar largura de banda no canal *uplink* através de um mecanismo de *polling unicast* realizado pela BS.

O serviço ertPS (*extended real-time Polling Service*) suporta tráfego de tempo real com taxa de transmissão variável, tais como, serviços de voz sobre IP com supressão de silêncio. Este serviço combina as características do serviço UGS e do serviço rtPS. As conexões ertPS recebem *grants* periódicos como no serviço UGS, entretanto, como a taxa de transmissão é variável, a SS pode utilizar os *grants* concedidos para solicitar a alteração da largura de banda, como no serviço rtPS.

O serviço nrtPS (*non-real-time Polling Service*) é utilizado por fluxos sem requisitos de tráfego de tempo real, mas que necessitam de melhores condições do que os serviços de melhor esforço. O serviço oferece *polling unicast*, porém com menos frequência do que o serviço rtPS. Além disso, é permitida a utilização de *slots* de contenção reservados para requisição de banda.

O serviço BE (*Best Effort Service*) é utilizado pelo tráfego de melhor esforço, sem garantias de QoS, tais como as aplicações elásticas. As regras para requisição e transmissão devem permitir que as SSs, utilizando o serviço BE, participem de períodos de contenção para enviar as mensagens de requisição de largura de banda.

A Tabela 2.2 mostra as classes de serviço e alguns parâmetros de QoS associados a cada classe de serviço.

Tabela 2.2. Classes de Serviço e os parâmetros de QoS.

Classes de Serviço	Parâmetros de QoS						
	MRTR	MSTR	ML	TJ	TP	SFS	R/TP
UGS		X	X	X	X	X	X
ertPS	X	X	X	X	X	X	X
rtPS	X	X	X		X	X	X
nrtPS	X	X			X	X	X
BE						X	X

Uma vez estabelecido um fluxo de serviço, a BS e a SS devem garantir a QoS de acordo com o conjunto de parâmetros de QoS definidos para o fluxo de serviço. Os fluxos de serviço podem ser criados nas duas direções, *uplink* e *downlink*. Assim sendo, a abordagem básica para se garantir QoS, é fazer alocação de banda de maneira justa e eficiente em ambas as direções, *downlink* e *uplink*, responsabilidade esta atribuída para a BS.

O padrão IEEE 802.16 provê especificações para as classes de serviço, entretanto, não define o algoritmo de escalonamento que será utilizado. O algoritmo de escalonamento tem como objetivo fazer alocação de largura de banda entre as SSs e determinar a ordem de transmissão entre as mesmas, o que influencia diretamente na obtenção da QoS.

2.5.2. Mecanismos de requisição e alocação de banda

Inicialmente, para que as SSs possam entrar na rede de acesso, elas necessitam de obter uma autorização da BS. Isso é feito através do mecanismo de controle de admissão de conexões (CAC) da BS, que também não é definido pelo padrão IEEE 802.16. O mecanismo de CAC é responsável por negar ou autorizar a entrada de uma conexão na rede. Esta decisão deve ser tomada em função da capacidade da rede, pois uma vez admitida uma SS na rede, é necessário garantir a QoS por ela requerida, sem comprometer a QoS requerida pelas outras SSs que já estão na rede.

Uma vez autorizada a sua entrada na rede, a SS deve enviar sua requisição de largura de banda para a BS. Todavia, para que isso aconteça, primeiramente, a BS deve alocar largura de

banda para que as SSs possam enviar suas requisições de largura de banda. Esse processo é definido como mecanismo de *polling* [13].

O padrão IEEE 802.16 define dois mecanismos principais de *polling*: o mecanismo de *polling unicast* e o mecanismo de *polling* baseado em contenção [14]. O *polling unicast* é o mecanismo pelo qual a BS aloca largura de banda, de forma individual, para a SS enviar sua requisição de largura de banda. A BS efetua o *polling unicast* periodicamente. Após o *polling*, as SSs podem enviar suas mensagens de requisição de largura de banda (*Bandwidth Request - BR*) em resposta ao *polling*. Essas mensagens podem ser do tipo *stand-alone*, ou do tipo *piggyback*³. A Figura 2.10 ilustra o funcionamento do mecanismo de *polling unicast*.

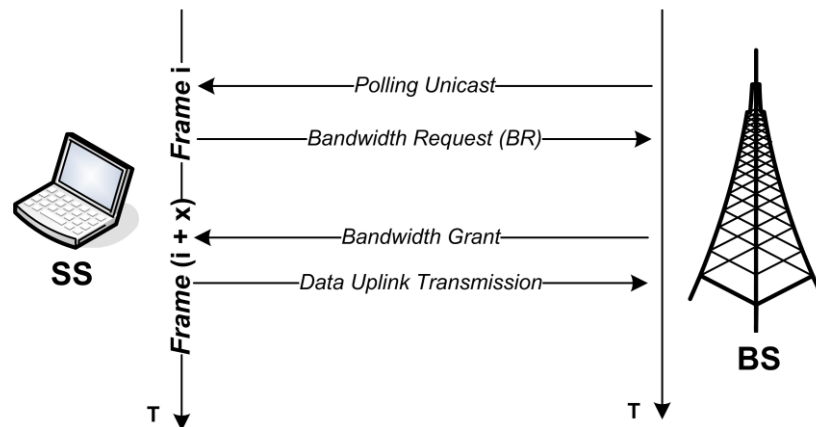


Figura 2.10. Mecanismo de *polling* definido pelo padrão IEEE 802.16 [2].

Como pode ser observado na Figura 2.10, a BS, durante o *frame i*, efetua o *polling* para a SS, que por sua vez envia sua mensagem de requisição de largura de banda. Uma vez recebida a mensagem de requisição de largura de banda proveniente da SS, a BS aloca a largura de banda no sentido *uplink* (*Bandwidth Grant*), para que a SS possa transmitir seus dados. Isso acontece em um determinado *frame* subsequente, ilustrado na Figura 2.10 como *frame (i + x)*.

O mecanismo de *polling* baseado em contenção geralmente é utilizado quando a BS não dispõe de recursos para efetuar o *polling* individualmente para todas as SSs. Neste caso, as

³ Esta forma de requisição de banda é feita aproveitando um *frame* de dados para enviar o pedido de requisição de largura de banda.

SSs disputam o envio da mensagem *BR* durante o período de contenção. Se múltiplas mensagens *BR* são enviadas ao mesmo tempo, colisões podem ocorrer. Neste caso, é utilizado um algoritmo de *backoff* exponencial para a resolução da contenção. Existem outros mecanismos que podem ser utilizados na requisição de largura de banda. Exemplos desses mecanismos são: o *polling multicast* e o método *Channel Quality Indicator Channel* (CQICH) [14]. Dependendo da QoS e dos parâmetros associados aos fluxos de serviço, um ou mais mecanismos de requisição de largura de banda podem ser utilizados.

2.6. Considerações finais

Este capítulo apresentou as principais características e o funcionamento do padrão IEEE 802.16. Mostrou-se que o padrão define dois tipos de topologias de comunicação, PMP e *Mesh*, e que o escopo principal do padrão envolve a camada física (PHY) e a camada de acesso ao meio (MAC).

Os tipos de camada física especificados pelo padrão foram descritos. Entretanto, maior destaque foi dado à descrição da camada MAC, uma vez que o foco principal de pesquisa desta tese é o algoritmo de escalonamento. Destacou-se que, diferentemente de outras tecnologias de rede sem fio, o padrão IEEE 802.16 foi desenvolvido com suporte à QoS na camada MAC. A partir disso, alguns dos principais mecanismos utilizados para a obtenção da QoS foram descritos. Todavia, o padrão deixa em aberto questões de suma importância para a obtenção da QoS. Tais questões envolvem o algoritmo de escalonamento de pacotes e o controle de admissão de chamadas, que atualmente são alvos de muitas pesquisas.

Capítulo 3

MECANISMOS DE ESCALONAMENTO EM REDES IEEE 802.16

3.1. Introdução

Os recursos existentes nas redes IEEE 802.16 devem ser distribuídos de maneira eficiente, tendo como objetivo oferecer garantias de desempenho para as aplicações e, ao mesmo tempo, permitir o compartilhamento dos recursos de acordo com critérios de equidade (*fairness*). É nesse contexto que são utilizados os algoritmos de escalonamento. Portanto, os algoritmos de escalonamento constituem o principal mecanismo responsável pelas garantias de desempenho para as diferentes aplicações existentes na rede.

Como descrito no Capítulo 2, o padrão IEEE 802.16 foi desenvolvido com mecanismos para dar suporte a uma grande variedade de tráfego, com características heterogêneas e diferentes requisitos de QoS. Entretanto, o padrão deixa em aberto questões relacionadas ao escalonamento dos diferentes fluxos de serviço.

Este capítulo tem como objetivo apresentar o estado da arte das propostas de algoritmos de escalonamento para redes IEEE 802.16. Para isso, na Seção 3.2 descreve-se o modo como esses algoritmos são classificados na literatura. As características desses algoritmos são descritas na Seção 3.2.1.

3.2. O estado da arte dos algoritmos de escalonamento para redes IEEE 802.16

Como já citado na Seção 2, o algoritmo de escalonamento é o principal mecanismo responsável pela obtenção da QoS nas redes IEEE 802.16. Assim sendo, o desempenho das aplicações, principalmente aquelas de tempo real, depende diretamente da disciplina de escalonamento utilizada. Atualmente, várias pesquisas vêm sendo desenvolvidas sobre escalonamento em redes IEEE 802.16. Entretanto, estudos mostram que o desenvolvimento de um escalonador justo e eficiente ainda é uma área de pesquisa em desenvolvimento [15][16][17].

Os algoritmos de escalonamento para redes IEEE 802.16 encontrados na literatura foram desenvolvidos em conformidade com algumas características definidas no padrão. Assim sendo, existem várias classificações de algoritmos de escalonamento para redes IEEE 802.16, as quais são descritas a seguir.

3.2.1. Classificação dos algoritmos de escalonamento

De um modo geral, os algoritmos de escalonamento são classificados da seguinte forma: algoritmos de escalonamento desenvolvidos para redes PMP e algoritmos de escalonamento desenvolvidos para redes *Mesh*. Dentro de tal classificação, os algoritmos de escalonamento também podem ser distinguidos em: algoritmos de escalonamento *downlink*, algoritmos de escalonamento *uplink*, ou algoritmos de escalonamento *downlink* e *uplink*. A Figura 3.1 ilustra a classificação geral dos algoritmos de escalonamento para redes IEEE 802.16.

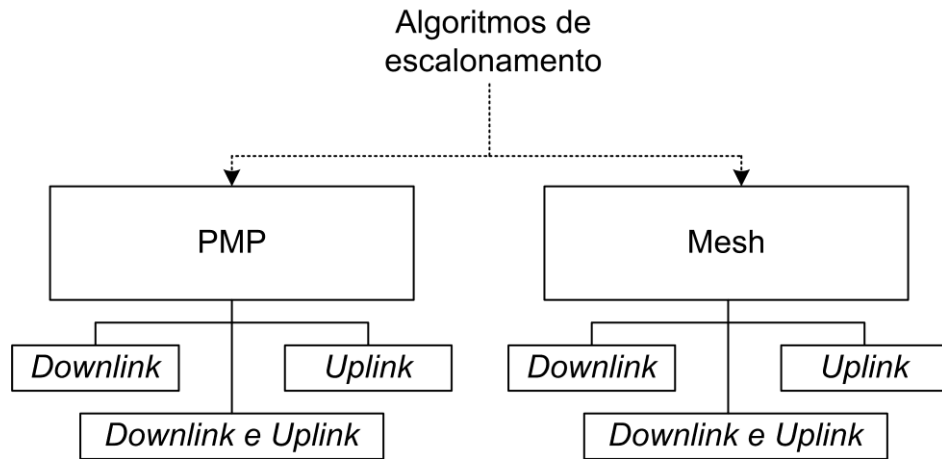


Figura 3.1. Classificação geral dos algoritmos de escalonamento [64].

De um modo mais específico, são várias as classificações dos algoritmos de escalonamento para redes IEEE 802.16 encontradas na literatura. Os autores em [15] classificam os algoritmos de escalonamento em três categorias: algoritmos homogêneos, algoritmos híbridos e algoritmos oportunistas. A Figura 3.2 ilustra essa classificação.

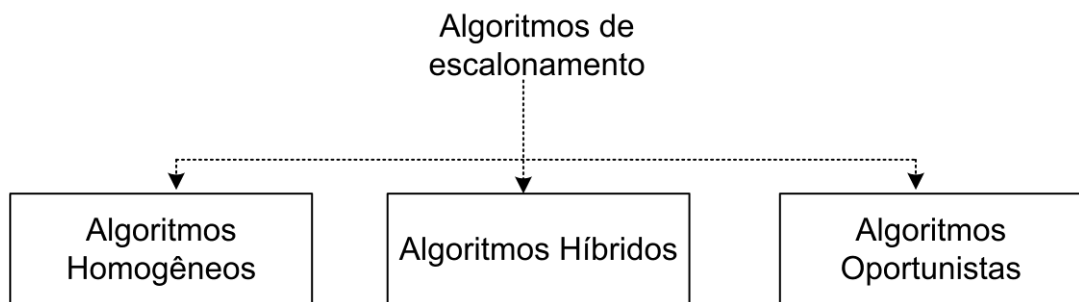


Figura 3.2. Classificação dos algoritmos de escalonamento definidas em [15].

As três categorias têm como objetivo comum satisfazer os requisitos de QoS das aplicações. O que difere uma categoria da outra são as características dos algoritmos de escalonamento utilizados e também o número de algoritmos empregados para garantir a QoS para as classes de serviço.

A categoria de algoritmos homogêneos consiste na utilização exclusiva de uma das políticas de escalonamento clássicas, propostas para redes cabeadas, mas que foram adaptadas para as redes IEEE 802.16. Exemplos de tais algoritmos são: *Round Robin* (RR) [18],

Weighted Round Robin (WRR) [19], *Deficit Round Robin (DRR)* [20], *Earliest Deadline First (EDF)* [21], *Weighted Fair Queuing (WFQ)* [22]. Já a categoria de algoritmos híbridos tem como característica combinar os vários algoritmos de escalonamento tradicionais, acima citados, a fim de criar um algoritmo que possa ser utilizado nas redes IEEE 802.16. Os algoritmos de escalonamento oportunistas são aqueles que exploram as variações na qualidade do canal de comunicação, dando prioridade de transmissão aos usuários com melhor qualidade no canal de comunicação. Esta técnica também é conhecida como *cross-layer*.

Os autores em [16] classificam os algoritmos de escalonamento em duas categorias: os algoritmos que utilizam informações provenientes da camada física e os algoritmos que não utilizam estas informações. A Figura 3.3 ilustra a classificação dos algoritmos de escalonamento para redes IEEE 802.16 definida em [16].

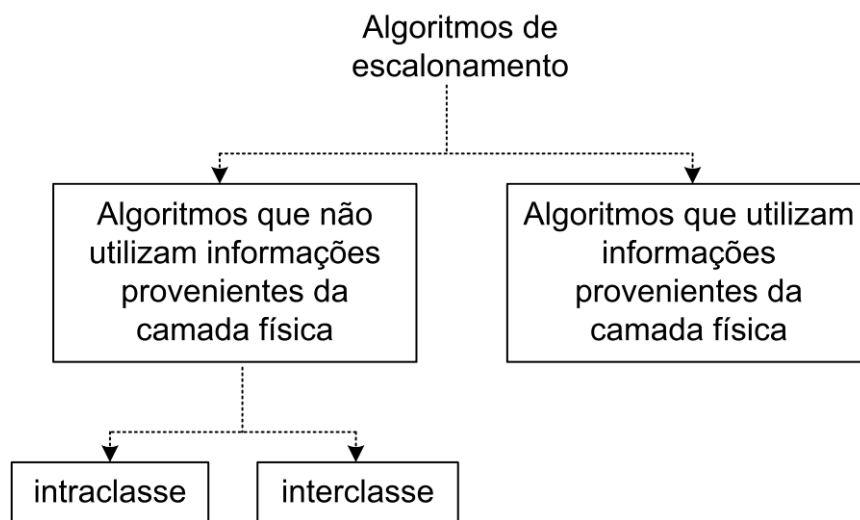


Figura 3.3. Classificação dos algoritmos de escalonamento definida em [16].

A categoria de algoritmos de escalonamento que não utilizam as informações provenientes da camada física é subdividida em dois grupos: intraclasse e interclasse. O algoritmo intraclasse é utilizado para alocar recursos para as conexões pertencentes a uma mesma classe de serviço. Por outro lado, o algoritmo interclasse determina a prioridade de atendimento entre as classes de serviço. As soluções baseadas nessa categoria possuem as

mesmas características existentes na categoria de escalonamento híbrido, classificada em [15], que consiste em combinar vários algoritmos de escalonamento (interclasse e intraclasse) a fim de criar um algoritmo que possa atender aos requisitos de QoS das aplicações. Além disso, o desenvolvimento de soluções de escalonamento baseadas nessa categoria é feito assumindo um canal de comunicação livre de erros. Entretanto, devido à natureza do meio de comunicação sem fio, essa premissa não é verdadeira, pois existe uma grande variação no enlace de comunicação sem fio.

A categoria de algoritmos de escalonamento que utilizam informações provenientes da camada física tem como objetivo fazer a alocação dos recursos de forma eficiente, levando em consideração a variação do canal de comunicação sem fio. Assim sendo, as decisões do escalonador sobre a alocação de recursos e prioridade de transmissão, geralmente são feitas baseadas em parâmetros, tais como, a qualidade do canal de comunicação e a taxa de erros.

Algumas das políticas de escalonamento classificadas em [15] também são estudadas e classificadas em [16]. Entretanto, o trabalho desenvolvido em [16] também avalia a capacidade desses algoritmos de distribuir os recursos entre várias conexões de uma mesma classe de serviço (intraclasse), como também em classes de serviço diferentes (interclasse). Os autores em [16] concluíram que alguns algoritmos de escalonamento podem ser aplicados para algumas classes de serviço de acordo com suas características, por exemplo, EDF para a classe rtPS, WFQ para as classes nrtPS e BE. Todavia, existem ainda algumas características da rede WiMAX que foram pouco exploradas no desenvolvimento de novos algoritmos de escalonamento, como por exemplo, o mecanismo de *polling*, otimização do *backoff*, entre outros.

A Figura 3.4 ilustra a classificação de algoritmos de escalonamento definida em [17]. Como pode ser observado na Figura 3.4, os autores em [17] também classificam os algoritmos

em três categorias: algoritmos baseados em enfileiramento de pacotes, algoritmos baseados em estratégias de otimização e algoritmos *cross-layer*.

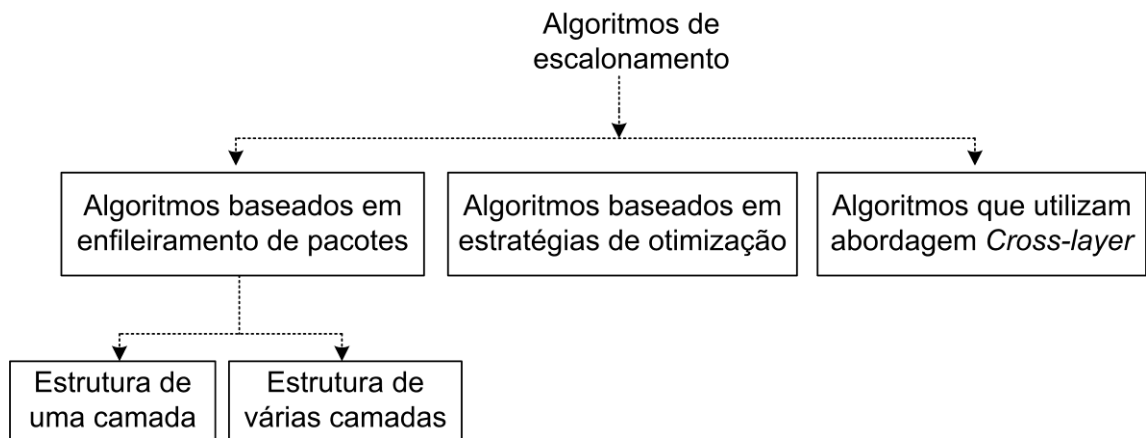


Figura 3.4. Classificação dos algoritmos de escalonamento definida em [17].

A estratégia de escalonamento baseada em enfileiramento de pacotes possui as mesmas características dos algoritmos desenvolvidos para redes cabeadas. Esta categoria é dividida em dois grupos: estruturas de uma camada, que consiste em utilizar apenas um único algoritmo de escalonamento para todas as classes de serviço; e estrutura de várias camadas, que consiste em dividir o escalonamento em duas ou mais etapas, definindo um escalonamento multinível. Todavia, os algoritmos pertencentes a essa categoria são os mesmos classificados em [15] como algoritmos homogêneos, e também os mesmos classificados por [16] como algoritmos que não utilizam informações provenientes da camada física.

A estratégia de escalonamento baseada em otimização consiste no desenvolvimento de algoritmos de escalonamento exclusivos para as redes baseadas no padrão IEEE 802.16. Neste caso, os algoritmos são desenvolvidos levando em consideração as características da rede definidas no padrão.

Os algoritmos de escalonamento que pertencem à categoria de escalonamento *cross-layer* possuem as mesmas características dos algoritmos definidos como oportunistas em [15]

e [16]. A Tabela 3.1 resume a classificação existente na literatura sobre os algoritmos de escalonamento e exemplifica alguns algoritmos de escalonamento que foram avaliados para redes WiMAX. A Tabela 3.2 mostra uma breve comparação das categorias de algoritmos de escalonamento.

Tabela 3.1. Classificação dos algoritmos de escalonamento.

		Propostas descritas em:				Exemplos de Algoritmos de escalonamento
		[15]	[16]	[17]		
Classificação	Homogêneo	Algoritmos que não utilizam informações provenientes da camada Física	Intraclasse	Algoritmos baseados em estratégia de enfileiramento	Estrutura de uma camada	RR, WRR, DRR, EDF, WFQ
					Estrutura de duas camadas	
	Híbrido		Interclasse	Algoritmos baseados em estratégia de otimização		EDF +WFQ+FIFO EDF + WFQ WRR+RR+PR DFPQ
Oportunista	Algoritmos que utilizam informações provenientes da camada física			Algoritmos que utilizam abordagem <i>cross-layer</i>		TRR, O-DRR, mSIR, mmSIR

Tabela 3.2. Comparação entre as categorias de algoritmos de escalonamento.

Categoria	Vantagens	Desvantagens
Algoritmos que não utilizam a camada física	Baixa complexidade de implementação	A maioria das soluções considera um canal de comunicação sem erro e sem perdas, o que não é real em se tratando de uma rede sem fio
Estrutura de uma camada	Implementação simples, trata os pacotes das diversas classes de serviço como se fosse uma única fila de escalonamento	Não garante que todos os requisitos de QoS das aplicações sejam garantidos. As aplicações de tempo real são as mais prejudicadas
Estrutura de várias camadas	Cada classe de serviço é tratada por uma política de escalonamento diferente, de acordo com suas características	Maior complexidade de implementação. O escalonamento é executado em vários estágios
Algoritmos que utilizam informações provenientes da camada física	Utilizam informações da camada física para otimizar os recursos e garantir a QoS para as diversas classes de serviço	Existe a necessidade de elaborar mecanismos para utilizar as informações da camada física de forma adequada e apropriada

É possível observar na Tabela 3.1 que muitos dos algoritmos de escalonamento classificados como homogêneos também são utilizados no desenvolvimento de soluções de algoritmos de escalonamento classificados como híbridos. O mesmo já não ocorre com os algoritmos de escalonamento classificados como oportunistas. Tais algoritmos têm como premissa utilizar as características da rede WiMAX para efetuar o escalonamento. Assim sendo, seu desenvolvimento se torna específico para as redes WiMAX.

A seguir serão descritos alguns algoritmos de escalonamento que foram avaliados em redes WiMAX e que fazem parte da classificação descrita nas Tabelas 3.1 e 3.2. Uma maior ênfase será dada para os algoritmos de escalonamento *uplink* na BS.

3.2.2. Algoritmos homogêneos

Como já descrito na Seção 3.2.1, a categoria de algoritmos homogêneos consiste na utilização de algoritmos de escalonamento que originalmente foram desenvolvidos para redes cabeadas, também conhecidos como algoritmos de escalonamento clássicos ou tradicionais. Neste caso, o termo homogêneo se justifica pelo fato de se utilizar apenas um algoritmo de escalonamento para a obtenção da QoS, não existindo nenhuma combinação entre outros algoritmos para tal propósito. Exemplos de tais algoritmos são: *Round Robin* (RR), *Weighted Round Robin* (WRR), *Earliest Deadline First* (EDF) e *Weighted Fair Queuing* (WFQ).

O algoritmo de escalonamento RR distribui igualmente os recursos do canal de comunicação entre todas as SSs. Esse algoritmo é simples e de fácil implementação. Todavia, esse algoritmo não é apropriado para sistemas que possuem vários tipos de tráfegos com diferentes níveis de prioridade. Entretanto, em [23] os autores consideraram que não há muito tempo disponível para efetuar o escalonamento, e devido a isso, o algoritmo de escalonamento deve ser simples, desenvolvido em apenas um nível de escalonamento. Por esse motivo, os autores em [23] propuseram uma solução de algoritmo de escalonamento baseada no algoritmo RR. Eles argumentam também que não há necessidade de se utilizar algoritmos

baseados em FQ (*Fair Queueing*), uma vez que os pesos utilizados por esses algoritmos são números de ponto flutuante, enquanto a quantidade de *slots* alocados nas redes IEEE 802.16 é um valor inteiro. A solução proposta compreende três estágios. No primeiro estágio, a BS calcula um número mínimo de *slots* que devem ser fornecidos para cada conexão de modo que os requisitos de QoS sejam satisfeitos. No segundo estágio, caso haja o número de *slots* necessários disponível, eles serão distribuídos para as classes de serviços rtPS, nrtPS e BE. No terceiro estágio, a BS faz a ordenação dos *slots* dentro da mensagem UL-MAP para que os requisitos de tempo sejam garantidos. Entretanto, o algoritmo não fornece garantias de atraso máximo limitado.

O algoritmo WRR, que é uma extensão do RR, foi avaliado no sistema WiMAX pelos autores em [24], onde considerou-se o escalonamento *uplink* na BS. Neste caso, a largura de banda foi distribuída entre as SSs de acordo com pesos previamente definidos. Discutiu-se em [24] que um peso pode ser atribuído para cada SS para representar sua prioridade de transmissão, de acordo com a classe de tráfego existente. Por exemplo, uma SS com um tráfego rtPS teria um peso maior do que uma SS com tráfego nrtPS ou BE. Entretanto, no caso de haver dois tipos de tráfego diferentes na mesma SS, tal premissa não seria eficaz. Além disso, através de modelagem e simulação, os autores concluíram que o algoritmo WRR não provê um bom desempenho quando utiliza-se tráfego com pacotes de tamanho variável.

A performance do algoritmo de escalonamento EDF foi avaliada em [25]. O EDF foi originalmente proposto para aplicações de tempo real em redes cabeadas. O algoritmo atribui para cada pacote um *deadline*, e a partir disso, ele aloca largura de banda para as SSs que possuem pacotes cujo *deadline* está prestes a expirar. Os autores afirmam que o algoritmo de escalonamento EDF é adequado para as classes de serviço UGS e rtPS, tendo em vista que essas classes de serviço possuem restrições temporais. Entretanto, o algoritmo EDF pode causar “inanição” para as classes de serviço nrtPS e BE, pois elas seriam atendidas apenas

quando não houvessem pacotes nas filas das classes de serviço UGS e rtPS. Além disso, os autores não demonstraram como o *deadline* é calculado para cada pacote existente nas filas das SSs.

O algoritmo de escalonamento WFQ também foi avaliado em [25] para efetuar o escalonamento *uplink* na BS. Os resultados mostraram que a performance do WFQ foi superior se comparado com o algoritmo WRR, quando utilizou-se pacotes de tamanho variável. A principal desvantagem do algoritmo WFQ é sua complexidade de implementação, pois, é necessário calcular o tempo final do serviço para cada pacote. A Tabela 3.3 resume as vantagens e desvantagens das propostas discutidas nesta seção.

Tabela 3.3. Vantagens e desvantagens dos algoritmos discutidos na Seção 3.2.2.

Algoritmos	Vantagens	Desvantagens
RR	Distribui largura de banda de forma equitativa para as classes de serviço	Não garante os requisitos de atraso e <i>jitter</i>
WRR	Garante largura de banda mínima para as conexões	Não tem uma boa performance para pacotes de tamanho variável
EDF	Adequado para aplicações de tempo real	Pode causar “inanição” para as aplicações não tempo real
WFQ	Possui bom desempenho na distribuição de recursos	Complexidade na implementação. É necessário calcular o tempo de serviço dos pacotes

3.2.3. Algoritmos híbridos ou hierárquicos

Os algoritmos híbridos, ou também chamados hierárquicos, têm como característica combinar vários algoritmos de escalonamento tradicionais a fim de criar um algoritmo que possa garantir diferentes requisitos de QoS.

Os autores em [26] foram os primeiros a introduzirem uma estrutura hierárquica para alocação de banda em sistemas 802.16. A Figura 3.5 ilustra tal estrutura de escalonamento.

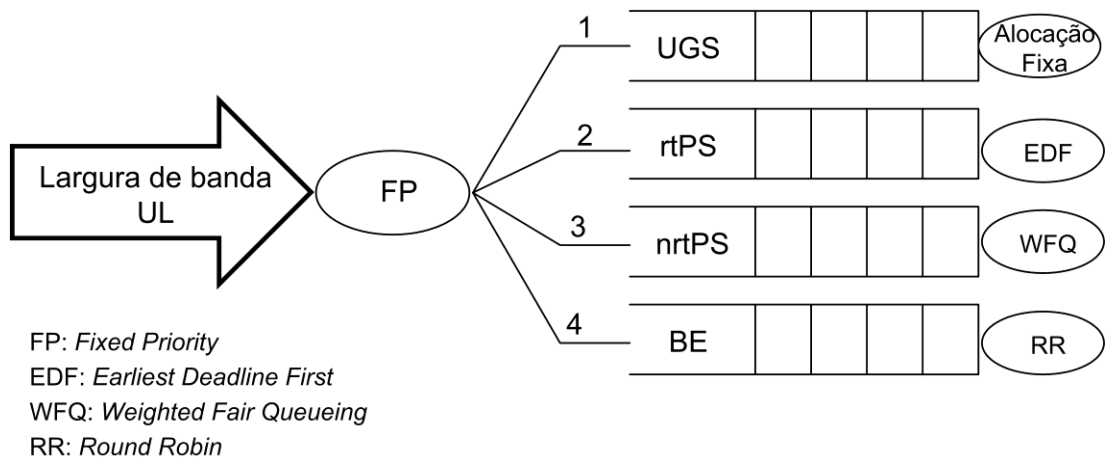


Figura 3.5. Estrutura de escalonamento hierárquico [26].

Como pode ser observado na Figura 3.5, a estrutura de escalonamento proposta em [26] combina as políticas de escalonamento FP, EDF, WFQ e RR. A alocação de largura de banda ocorre em duas fases: na primeira fase, utilizando a disciplina FP, o escalonador distribui a banda entre as diferentes classes de serviço. Na segunda fase, a banda recebida por cada classe de serviço é distribuída entre as suas conexões de acordo com as seguintes políticas: EDF para o serviço rtPS e WFQ para o serviço nrtPS. As conexões UGS recebem alocação de banda fixa. A largura de banda alocada para o serviço BE é igualmente distribuída entre as conexões. Uma das desvantagens de se utilizar a política de escalonamento por prioridades é a questão da “inanição”. Uma classe de menor prioridade pode não ser atendida pelo algoritmo de escalonamento enquanto houver dados de classes de alta prioridade para serem transmitidos. O algoritmo de escalonamento proposto em [26] é avaliado através de experimentos de simulação, entretanto, apenas o tráfego BE e nrtPS foram considerados.

Os autores em [27] propuseram um algoritmo de escalonamento híbrido que combina as disciplinas de escalonamento EDF, WFQ e FIFO. A alocação de largura de banda é feita da mesma forma como já descrito em [26]. Entretanto, os autores em [27] utilizaram a política de escalonamento FIFO para a classe de serviço BE. A Figura 3.6 ilustra a arquitetura de escalonamento utilizada em [27].

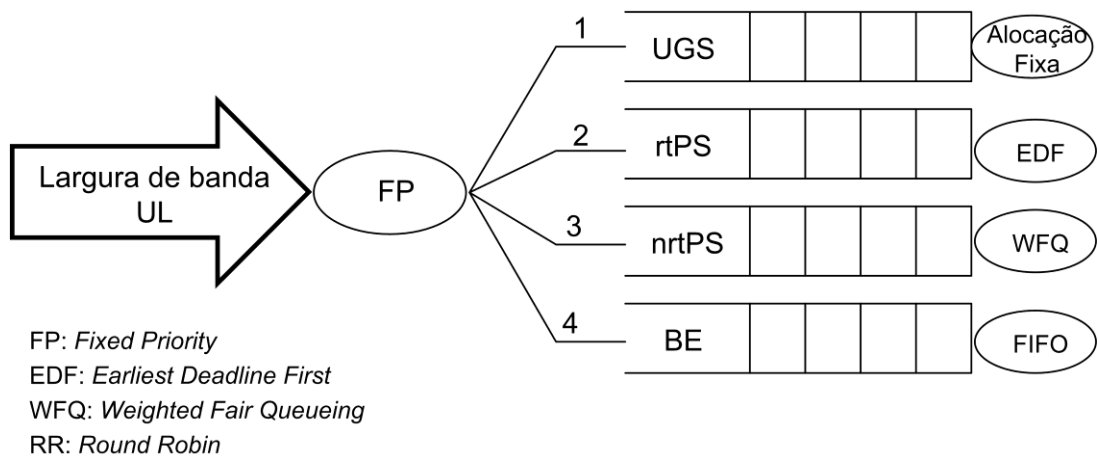


Figura 3.6. Estrutura de escalonamento utilizada em [27].

A mesma estrutura hierárquica desenvolvida em [26] é utilizada pelos autores do trabalho desenvolvido em [28]. Entretanto, os autores em [28] utilizam na primeira etapa do escalonamento a política *Deficit Fair Priority Queue* (DPFQ). Os resultados de simulação mostraram que o algoritmo DPFQ é mais justo do que o algoritmo *Fixed Priority*, e evita a “inanição” para o tráfego BE. Porém, nenhum resultado sobre o atraso foi mostrado pelos autores.

A estrutura hierárquica utilizada pelos autores em [29] é similar aos trabalhos propostos em [26 - 28]. Entretanto, os autores em [29] desenvolveram um algoritmo para ser utilizado na primeira etapa do escalonamento. Tal algoritmo foi desenvolvido baseado na política DRR, nomeado de *Adaptive Deficit Round Robin* (ADRR). Os autores alegam que o algoritmo ADRR faz a distribuição de largura de banda entre as classes de serviço de forma mais adequada se comparado com o trabalho desenvolvido em [28]. Além disso, os autores em [29] inseriram, na estrutura hierárquica, a classe de serviço ertPS, pois tal classe não fazia parte da estrutura hierárquica definida pelos autores em [26 - 28]. Já na segunda etapa, a banda recebida por cada classe de serviço é distribuída entre suas conexões, de acordo com as seguintes políticas: EDF para os serviços ertPS e rtPS, WFQ para o serviço nrtPS e RR para o serviço BE. A Figura 3.7 ilustra a arquitetura de escalonamento utilizada em [29].

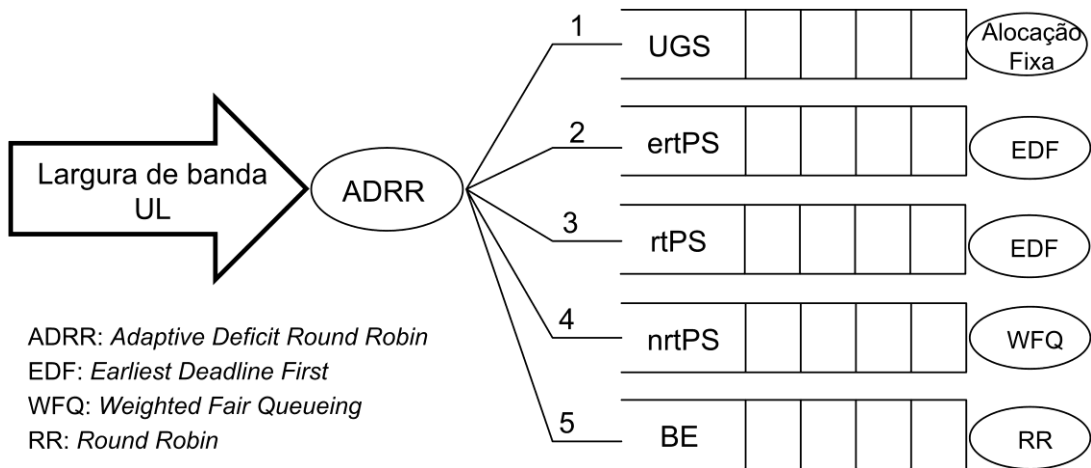


Figura 3.7. Estrutura de escalonamento utilizada em [29].

A estrutura de escalonamento desenvolvida pelos autores em [30] combina três algoritmos de escalonamento para garantir os requisitos de QoS para as aplicações. A Figura 3.8 ilustra a arquitetura proposta em [30].

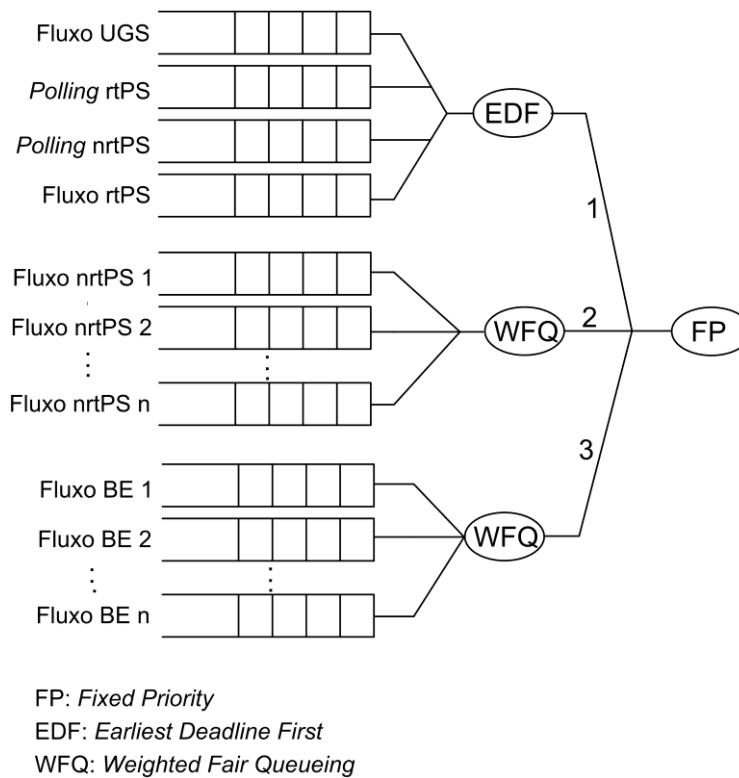


Figura 3.8. Estrutura de escalonamento utilizada em [30].

Como pode ser observado na Figura 3.8, os fluxos sensíveis ao atraso, nomeados na Figura 3.8 como fluxo UGS, fluxo rtPS, *polling* rtPS e *polling* nrtPS são servidos pelo

escalonador 1, de acordo com o algoritmo de escalonamento EDF. Os fluxos nrtPS, que possuem como requisito de QoS a largura de banda mínima, são servidos pelo escalonador 2, que utiliza o algoritmo de escalonamento WFQ. Neste caso, os pesos utilizados pelo algoritmo WFQ correspondem à proporção de largura de banda mínima requerida pelas conexões. O algoritmo WFQ também é aplicado pelo escalonador 3 para servir os fluxos da classe BE. Entretanto, neste caso, os pesos são utilizados para atender diferentes prioridades requeridas pelas conexões BE.

A Tabela 3.4 resume as vantagens e desvantagens das propostas discutidas nesta seção.

Tabela 3.4. Vantagens e desvantagens dos algoritmos discutidos na Seção 3.2.3.

Propostas descrita em:	Vantagens	Desvantagens
[26]	Garante atraso limitado para os serviços UGS e rtPS	Por utilizar a política <i>Fixed Priority</i> , é possível ocorrer “inanição” para os fluxos de serviço nrtPS e BE
[27]	Possui as mesmas vantagens do algoritmo desenvolvido em [26]	Por utilizar a política <i>Fixed Priority</i> , é possível ocorrer “inanição” para os fluxos de serviços nrtPS e BE
[28]	Utiliza a política DPFQ para distribuir banda entre as classes de serviço, sendo mais justo do que o algoritmo <i>Fixed Priority</i>	O escalonamento das conexões é feito utilizando <i>deficit counter</i> e <i>quantum</i> , o que aumenta a complexidade de implementação
[29]	Garante o atraso máximo limitado para as aplicações de tempo real, e garante largura de banda mínima para as aplicações	A utilização do algoritmo de escalonamento baseado no DRR introduz maior complexidade na implementação

	sem restrições temporais	
[30]	O algoritmo garante largura de banda mínima e atraso máximo limitado para as aplicações de tempo real	O escalonamento é efetuado em três estágios. Complexidade na implementação

3.2.4. Algoritmos *cross-layer*

A idéia principal existente no desenvolvimento de algoritmos que utilizam uma abordagem *cross-layer* é integrar os recursos e as informações disponíveis nas diferentes camadas (camadas superiores ou inferiores), criando um sistema de comunicação que seja altamente adaptativo e eficiente [31]. Tal eficiência está relacionada com a capacidade de oferecer garantias de QoS às aplicações. São várias as informações que podem ser utilizadas pelo algoritmo *cross-layer*, como exemplo, o estado do canal de comunicação, a taxa de erro de transmissão e o valor do atraso de uma determinada classe de serviço.

Em [32], os autores desenvolveram um algoritmo de escalonamento que aloca os recursos de rádio para as SSs que possuem os melhores níveis de SNR. A partir do valor da SNR, o algoritmo determina a MCS que será utilizada e atribui prioridades para as SSs. A SS que tem maior prioridade é aquela que possui o esquema de modulação e codificação de maior ordem, oferecendo assim, maior eficiência espectral. Contudo, as SSs que possuem um menor nível de SNR nunca serão atendidas. Os autores também propuseram um algoritmo de escalonamento denominado *Temporary Removal Scheduler* (TRS). Este algoritmo cria uma lista de SSs baseando-se nas condições do canal de comunicação. As SSs que possuem condições ruins de transmissão são bloqueadas temporariamente. O tempo em que as SSs ficam bloqueadas é pré-determinado e ajustável. Após expirar esse tempo, as SSs são inseridas na lista de escalonamento independentemente da qualidade do canal de

comunicação. Esse algoritmo de escalonamento pode ser combinado com outras políticas de escalonamento, tais como RR, WRR etc.

Os autores em [33] desenvolveram um algoritmo de escalonamento similar ao algoritmo apresentado em [32]. O algoritmo de escalonamento desenvolvido pelos autores em [33] também cria uma lista de SSs baseando-se nas condições do canal de comunicação. Entretanto, diferentemente do trabalho apresentado em [32], as SSs que possuem condições ruins de transmissão não são bloqueadas. Ao invés disso, o escalonador atende tais SSs após alocar recursos para as SSs que possuem melhores condições de comunicação. Todavia, para que esse atendimento seja realmente executado, é necessário que haja recursos disponíveis após o atendimento das SSs que possuem melhores condições de comunicação, o que não pode ser garantido. Utilizando o recurso de modelagem e simulação, os autores em [33] compararam a performance do algoritmo proposto com o algoritmo apresentado em [32]. Neste caso, houve uma melhora em relação ao atraso médio.

O algoritmo de escalonamento apresentado em [34] atribui para cada conexão admitida no sistema uma determinada prioridade. A conexão com maior prioridade é atendida em primeiro lugar, em cada *frame*. A prioridade é atualizada dinamicamente, de acordo com a qualidade atual do canal de comunicação. Além da prioridade, o algoritmo também considera a MCS utilizada de acordo com a SNR. O objetivo é maximizar a utilização do enlace ajustando o modo de transmissão de acordo com a SNR.

Os autores em [35] modificaram a estrutura do algoritmo de escalonamento apresentado em [34]. As principais modificações foram: a inserção da classe de serviço ertPS na estrutura do algoritmo de escalonamento (tal classe de serviço não era considerada pelos autores em [34]) e a definição de uma função para escolher, dentre as conexões que possuem a mesma prioridade, aquela cujo *deadline* está mais próximo de expirar.

Os autores em [36] apresentaram uma proposta de escalonamento denominada *Oportunistic Deficit Round Robin (O-DRR)*. O escalonamento acontece da seguinte forma: a BS efetua o *polling* para todas as SSs periodicamente, a cada k frames. Após cada período, a BS determina um grupo de SSs elegíveis para serem escalonadas. Entretanto, a SS será escalonada quando dois requisitos forem satisfeitos: a fila não estiver vazia e o nível de SNR não estiver abaixo de um *threshold* pré-definido. As SSs escalonadas são trocadas dinamicamente, de acordo com a variação do canal de comunicação. Entretanto, devido à natureza desta política de escalonamento, é necessário manter um *deficit count* para cada conexão que será escalonada, o que introduz uma maior sobrecarga, principalmente em se tratando do escalonamento *uplink*, no qual a BS não possui informações precisas das filas que estão nas SSs.

Os autores em [37] desenvolveram um algoritmo de escalonamento baseado na política *Strict Priority (SP)* e também na MCS em uso pela SS. Antes de iniciar uma transmissão, a SS envia para a BS, a mensagem de requisição de largura de banda. Após o recebimento de tal mensagem, a BS converte em *slots* físicos, a quantidade de largura de banda requisitada. Isso é feito levando em consideração a MCS utilizada na camada física. Para cada conexão, o escalonador verifica se possui *slots* suficientes no sistema para atender, em sua totalidade, a quantidade de *slots* requisitada. Caso não possua, a BS verifica se possui *slots* suficientes para atender os requisitos mínimos de largura de banda da aplicação. Após tais verificações, prioridades são atribuídas para as conexões. A conexão que possui maior prioridade é aquela que possui a MCS de maior ordem, de acordo com a seguinte seqüência: 64QAM > 16QAM > QPSK > BPSK. Além disso, os autores mencionam que o atraso dos pacotes é levado em consideração para fazer o escalonamento, entretanto, eles não descrevem como isso é feito. A Tabela 3.5 resume as vantagens e desvantagens das propostas discutidas nesta seção.

Tabela 3.5. Vantagens e desvantagens dos algoritmos discutidos na Seção 3.2.4.

Proposta descrita em:	Vantagem(ns):	Desvantagem(ns):
[32]	As SSs que possuem melhores condições de transmissão no canal de comunicação possuem maior prioridade	As SSs com qualidade inferior de transmissão nunca irão transmitir
[33]	Utiliza o mesmo princípio do trabalho desenvolvido em [32] com a vantagem de não bloquear as SSs com qualidade inferior de transmissão	Não há garantia no atendimento das SSs que possuem qualidade inferior de transmissão, uma vez que o atendimento continua sendo prioritário para as SSs com melhores, condições de transmissão
[34]	Maximiza a utilização do enlace ajustando o modo de transmissão com a variação do canal	As SSs com qualidade de sinal inferior podem sofrer “inanição”
[35]	Insere a classe de serviço ertPS na estrutura do algoritmo de escalonamento desenvolvido pelos autores em [34]	As SSs com qualidade de sinal inferior podem sofrer “inanição”
[36]	Considera o mecanismo de <i>polling</i> no escalonamento	É necessário manter um <i>deficit count</i> para cada conexão que será escalonada, com isso, aumenta a complexidade de implementação no

		sentido <i>uplink</i>
[37]	A prioridade de transmissão é definida utilizando a política <i>Strictly Priority</i> e a MCS	As SSs que utilizam MCSs de menor ordem podem ser prejudicadas devido ao atraso, pois elas sempre terão as menores prioridades no atendimento

3.2.5. Algoritmos de escalonamento que combinam CAC

Uma boa prática para garantir QoS é combinar a política de escalonamento com uma estratégia eficiente de controle de admissão de conexões (CAC). A regra principal de uma estratégia de CAC é decidir aceitar ou não novas conexões, após verificar se os recursos disponíveis serão suficientes para todas as conexões já estabelecidas. Os mecanismos existentes na literatura abordam duas estratégias de CAC. A primeira estratégia consiste em ser flexível, ou seja, degradar a QoS de alguma outra conexão momentaneamente em detrimento de aceitar uma nova conexão. A segunda estratégia é mais simples e conservativa, consistindo em manter a QoS para as conexões já estabelecidas e simplesmente rejeitar as novas conexões, caso não haja recurso disponível.

Os trabalhos apresentados em [38 – 40] são baseados na primeira estratégia descrita. Tais trabalhos utilizam a seguinte premissa: diminuir, o quanto possível, os recursos providos para as conexões existentes no sistema a fim de aceitar novas conexões. Essas estratégias são combinadas com limiares (*thresholds*) pré-estabelecidos [41], ou com mecanismos de reserva de recursos para conexões mais sensíveis ao atraso, como por exemplo, as conexões da classe UGS.

Em [38], os autores apresentaram uma proposta em que os fluxos de serviço são priorizados de acordo com suas classes de serviço (UGS > ertPS > rtPS > nrtPS > BE). As conexões pertencentes a mesma classe de serviço são priorizadas de acordo com os requisitos

de atraso e *jitter* das aplicações. Caso haja um pedido de uma nova conexão, o mecanismo de CAC verifica se existe largura de banda disponível para aceitar a nova conexão. Caso positivo, a conexão é aceita, caso contrário a nova conexão é rejeitada. Caso haja um pedido de uma nova conexão, e a nova conexão for oriunda de um processo de *handoff*, o mecanismo de CAC irá verificar se existe largura de banda disponível para aceitar a conexão. Se houver recurso disponível, a conexão é aceita, caso contrário, é aplicada a política de degradação das conexões que estão recebendo largura de banda maior do que o mínimo previsto.

O esquema de CAC proposto em [39] atribui a maior prioridade para os fluxos UGS visando maximizar a utilização de largura de banda aplicando a política de empréstimo de banda e degradação. Uma predeterminada quantidade de largura de banda L é exclusivamente reservada para conexões UGS. A partir disso, uma nova conexão UGS só é aceita se houver recurso suficiente para garantir seus requisitos de QoS, caso contrário, a conexão será rejeitada. O modelo de degradação é aplicado quando uma nova conexão rtPS é requisitada, ou quando a criação de uma nova conexão nrtPS é requisitada. Nesse caso, uma determinada quantidade de largura de banda das conexões nrtPS é emprestada para as novas conexões rtPS ou nrtPS, desde que a largura de banda mínima das conexões nrtPS existentes não seja degradada.

Em [40], os autores apresentaram uma proposta que combina o algoritmo de escalonamento *uplink* com o mecanismo de CAC, ambos baseados na abordagem *Token-bucket*. Nesta proposta de CAC, cada conexão é caracterizada por dois parâmetros denominados: *token rate* (r_i) e *bucket size* (b_i). As conexões rtPS possuem um parâmetro extra (d_i) correspondente aos seus requisitos de atraso. Para evitar a “inanição”, os autores definiram um limiar que determina a quantidade máxima de conexões por classe de serviço. Se uma SS tenta estabelecer um novo fluxo de serviço com a BS, o algoritmo de CAC verifica se a quantidade de largura de banda requisitada é menor do que a largura de banda total

restante. A conexão será aceita se houver largura de banda disponível, caso contrário a conexão não será aceita.

Diferente de muitos trabalhos onde o mecanismo de CAC é baseado na disponibilidade de largura de banda, o algoritmo de CAC proposto em [41] leva em consideração os requisitos de atraso e *jitter* dos fluxos de serviço. Como as conexões possuem diferentes requisitos de QoS, definiu-se um intervalo denominado *Hyper Interval* (HI). Esse intervalo é utilizado para testar a admissibilidade das novas conexões, representando o intervalo pelo qual o processo de admissão é executado. Os autores, entretanto, consideraram que os requisitos de atraso e *jitter* das classes de serviço UGS e rtPS podem causar o bloqueio momentâneo para as novas conexões nrtPS. Os autores também incluem em seu esquema de CAC um agente que é responsável por monitorar o tamanho das filas das conexões rtPS e nrtPS, e fazer estimativas da largura de banda baseadas nas trocas instantâneas de tamanho de fila. A Tabela 3.6 resume as vantagens e desvantagens das propostas discutidas nesta seção.

Tabela 3.6. Vantagens e desvantagens dos algoritmos discutidos na Seção 3.2.5.

Proposta descrita em:	Vantagem(ns):	Desvantagem(ns):
[38]	O escalonamento de fluxos da mesma classe de serviço é priorizado de acordo com os requisitos de atraso e <i>jitter</i>	Pode causar “inanição” para os fluxos de serviço não tempo real
[39]	A degradação é aplicada para os fluxos das classes de serviços rtPS e nrtPS	Maior complexidade de implementação
[40]	Utiliza o mecanismo de <i>Token Bucket</i> no CAC	Maior complexidade de implementação

[41]	Os requisitos de atraso e <i>jitter</i> são considerados pelo mecanismo de CAC.	Os requisitos de atraso e <i>jitter</i> das classes de serviço UGS e rtPS podem causar o bloqueio momentâneo para as novas conexões nrtPS.
------	---	--

3.3. Considerações finais

Este capítulo discutiu o escalonamento em redes IEEE 802.16. O padrão IEEE 802.16 define a utilização de mecanismos de escalonamento, como também a utilização do mecanismo de CAC em sua arquitetura. Entretanto, nenhum tipo de mecanismo é definido pelo padrão.

O estado da arte dos algoritmos de escalonamento para redes IEEE 802.16 foi apresentado. Os algoritmos de escalonamento para redes IEEE 802.16 encontrados na literatura foram desenvolvidos em conformidade com algumas características definidas pelo padrão. Assim sendo, existem várias classificações de algoritmos de escalonamento para redes IEEE 802.16. Baseando-se nessa classificação, alguns trabalhos sobre algoritmos de escalonamento foram discutidos. Um novo algoritmo de escalonamento para o tráfego *uplink* em redes IEEE 802.16 será apresentado no Capítulo 5.

Capítulo 4

O CONTROLE DE ADMISSÃO DE CONEXÕES EM REDES IEEE 802.16

4.1. Introdução

O mecanismo de controle de admissão de conexões é um componente muito importante na obtenção da QoS. Enquanto o mecanismo de escalonamento tem como objetivo garantir que os recursos requisitados sejam alocados de forma eficiente, o mecanismo de CAC tem como objetivo evitar que a rede seja saturada por uma quantidade excessiva de conexões, limitando o número de usuários na rede.

Assim sendo, o mecanismo de CAC é responsável por gerenciar a admissão ou a rejeição de conexões na rede. É neste mecanismo que se define o conjunto de ações que serão executadas pela rede durante a fase de estabelecimento de conexões. A decisão de aceitar uma conexão deve ser feita levando em consideração as garantias de QoS da própria conexão, como também das conexões já estabelecidas.

Como já descrito no Capítulo 2, o padrão IEEE 802.16 foi desenvolvido com mecanismos para dar suporte a QoS. Entretanto, o padrão deixa em aberto questões relacionadas aos dois principais mecanismos utilizados para a obtenção da QoS: o mecanismo de escalonamento e o mecanismo de CAC. O mecanismo de escalonamento foi descrito no

Capítulo 3. Assim sendo, este capítulo tem como objetivo apresentar o estado da arte sobre os mecanismos de CAC para redes IEEE 802.16. A Seção 4.2 descreve algumas questões importantes no desenvolvimento de mecanismos de CAC. A Seção 4.3 descreve as abordagens existentes na literatura sobre os mecanismos de CAC. Além disso, trabalhos que utilizam tais abordagens também são descritos. A Seção 4.4 traz as conclusões deste capítulo.

4.2. O Mecanismo de CAC

A principal função do mecanismo de CAC é decidir se aceita ou não uma nova conexão na rede. Esta decisão deve levar em conta questões, tais como: existem recursos suficiente na rede para aceitar uma nova conexão?; a QoS das conexões existentes não será afetada após a admissão da nova conexão? Assim sendo, o número de conexões estabelecidas na rede deve ser restringido pelo mecanismo de CAC. Desta forma, é possível evitar a saturação do enlace sem fio e, conseqüentemente, a violação das garantias de QoS das conexões já estabelecidas.

A eficiência de um mecanismo de CAC está relacionada com o atendimento de alguns requisitos, a fim de suportar toda a flexibilidade inerente aos diferentes tipos de tráfego e infraestrutura de rede. Dentre estes requisitos, podem ser citados:

- O baixo tempo de resposta entre o recebimento do pedido de conexão, e a decisão entre aceitar ou não a conexão;
- A existência de uma margem de segurança de modo a garantir que os parâmetros de QoS negociados sejam satisfeitos;
- Possuir um mecanismo de policiamento associado ao mecanismo de CAC. Dessa forma, é possível verificar a conformidade entre o tráfego negociado e o tráfego real, de modo a evitar que um tráfego excessivo na rede possa prejudicar a QoS das conexões admitidas na rede.

No padrão IEEE 802.16, quando uma SS tenta estabelecer uma conexão, e assim, definir um tipo de serviço, a mesma envia para a BS uma mensagem denominada *Dynamic Service Addition* (DSA). A Figura 4.1 ilustra o estabelecimento de conexão conforme definido pelo padrão IEEE 802.16. A mensagem DSA é utilizada para adicionar um novo serviço. Como já descrito no Capítulo 2, o padrão IEEE 802.16 define cinco classes de serviço e cada classe está associada a um conjunto de parâmetros de QoS. Além da mensagem DSA, o padrão define mais dois tipos de mensagens de gerenciamento: a mensagem *Dynamic Service Change* (DSC), utilizada para modificar os parâmetros dos fluxos de serviço, e a mensagem *Dynamic Service Delete* (DSD), utilizada para deletar um fluxo de serviço existente.

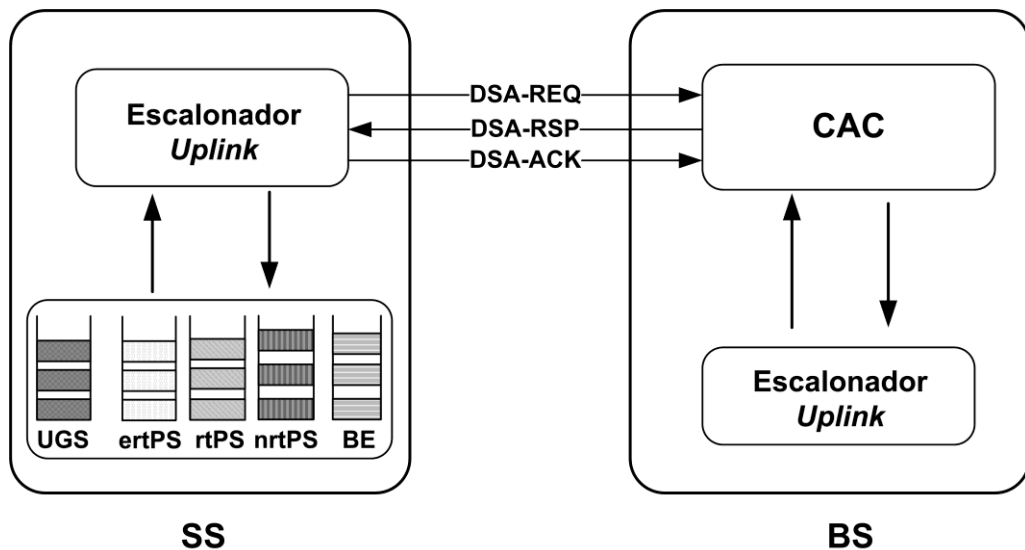


Figura 4.1. Estabelecimento de conexão definido pelo padrão IEEE 802.16 [2].

Quando uma SS deseja adicionar, modificar ou deletar uma conexão, ela envia as respectivas mensagens DSA, DSC ou DSD para o módulo de CAC da BS. Basicamente, o módulo de CAC obtém os requisitos de largura de banda e atraso definidos para cada conexão e verifica se o sistema possui recursos suficientes para admitir a conexão. Entretanto, existem várias questões que devem ser consideradas no desenvolvimento do mecanismo de CAC. Isso originou uma grande variedade de trabalhos sobre mecanismos de CAC, onde cada trabalho possui uma abordagem específica. As principais abordagens são descritas a seguir.

4.3. Abordagens existentes sobre mecanismos de CAC

São várias as abordagens existentes na literatura sobre o mecanismo de CAC. Tais abordagens possuem uma regra principal, que é decidir se aceita ou não uma nova conexão enquanto certifica-se de que os recursos existentes serão suficientes, tanto para as novas conexões, como também para as conexões já existentes.

Levando em consideração a não existência de recursos suficientes para atender a regra acima descrita, duas estratégias podem ser utilizadas. A primeira estratégia consiste em diminuir, gradativamente, os recursos das conexões existentes em função de aceitar uma nova conexão. A segunda estratégia é mais simples e conservadora, pois consiste em manter a QoS provida para as conexões já existentes e simplesmente rejeitar novas conexões. A seguir são descritos alguns trabalhos encontrados na literatura baseados nas estratégias acima descritas.

4.3.1. Mecanismos de CAC baseados em estratégia de degradação

A ideia principal desta estratégia é de decrementar, quando necessário e possível, os recursos providos para as conexões em andamento, a fim de poder aceitar uma nova conexão ou fluxo de serviço. Alguns trabalhos encontrados na literatura combinam esta estratégia com limiares que definem uma quantidade máxima de compartilhamento, a fim de evitar a “inanição”.

Os mecanismos de CAC que utilizam esta estratégia são baseados em algoritmos de “empréstimo” de largura de banda (*bandwidth borrowing*) [42], degradação de serviço (*service degradation*), ou “roubo” de largura de banda (*bandwidth stealing*) [39].

Os autores em [42] desenvolveram um mecanismo de CAC para garantir os requisitos de QoS para os fluxos de serviço das classes rtPS e nrtPS. Neste caso, utilizando a teoria dos jogos, os autores formularam um problema que consiste em buscar um ponto de equilíbrio entre os dois fluxos de serviço, a fim de oferecer largura de banda para as novas conexões,

como também, manter a QoS das conexões em andamento. Baseando-se em uma solução de jogo, os autores propuseram um esquema de CAC no qual, os jogadores são as conexões rtPS e nrtPS, que querem maximizar a performance em relação a QoS. Os autores utilizaram um algoritmo baseado em degradação de banda em um jogo não cooperativo.

O mecanismo de CAC previamente descrito na Seção 3.2.5 baseia-se no algoritmo de empréstimo de banda e degradação gradual [39]. Uma quantidade pré-determinada de largura de banda L é exclusivamente reservada para as conexões da classe de serviço UGS. Uma conexão UGS é aceita se existe largura de banda disponível para garantir seus requisitos de QoS. Caso contrário, a conexão é rejeitada. Seja B a largura de banda total, b_{ong} a largura de banda já atribuída para as conexões em andamento (UGS, rtPS e nrtPS), e b_{ugs} e b_{rtPS} o requerimento de largura de banda para uma nova conexão UGS e rtPS, respectivamente. O modelo de degradação é aplicado quando, por exemplo, uma nova conexão rtPS é requisitada e $b_{ong} + b_{rtPS} > B - L$. Neste esquema, os autores assumem que todas as conexões pertencentes ao mesmo tipo de serviço possuem os mesmos requerimentos de largura de banda, e que a largura de banda requisitada por uma conexão rtPS não varia entre o MRTR e o MSTR. Tais suposições simplificam o problema, entretanto, não contemplam os requisitos de QoS definidos pelo padrão.

Os autores em [40] combinam o algoritmo de escalonamento *uplink* com o mecanismo de CAC, baseando-se em uma abordagem *token-bucket*. Na proposta do CAC, cada conexão *uplink* é caracterizada por dois parâmetros denominados *token rate* (r_i) e *bucket size* (b_i). Os fluxos rtPS possuem um parâmetro extra, denominado (d_i), que corresponde ao requisito de atraso. Para evitar a “inanição”, os autores definiram um limiar máximo para a utilização de recursos para algumas classes de serviço. Desta forma, as classes de serviço que utilizarem uma maior quantidade de largura de banda do que foi definido em seu limiar terão menos chances de utilizar a capacidade restante do canal *uplink* (C_{remain}). Quando uma SS tenta

estabelecer uma nova conexão com a BS, o algoritmo proposto é aplicado da seguinte forma: se a largura de banda requerida for menor do que a capacidade restante do canal *uplink*, o fluxo é aceito. Caso contrário, a estratégia de “roubar” banda é aplicada para as classes de serviço de menor prioridade que estão utilizando maior quantidade de largura de banda do que foi definido em seus limiares.

Em [43], os autores utilizaram o algoritmo de empréstimo de largura de banda proporcional. Neste esquema, as conexões existentes que estão em processo de *handover* (*handover connections - HO*) são priorizadas sobre as novas conexões ($N_{_}$). As prioridades são definidas da seguinte forma: $HO_UGS > HO_rtPS \ \& \ HO_ertPS > N_UGS > N_rtPS \ \& \ N_ertPS > HO_nrtPS > N_nrtPS > HO_BE > N_BE$. Para as conexões UGS, é reservada uma quantidade de largura de banda que corresponde ao requisito de MSTR. Para as conexões da classe ertPS, rtPS e nrtPS é reservada a quantidade de largura de banda correspondente ao requisito mínimo de largura de banda. Nenhuma largura de banda é reservada para as conexões da classe BE. O algoritmo combina o empréstimo de largura de banda proporcional com uma determinada reserva de largura de banda para as conexões HO. Desta forma, uma nova conexão é bloqueada se a quantidade de largura de banda disponível for menor que a quantidade de largura de banda reservada, enquanto as conexões HO são bloqueadas somente se não houver largura de banda disponível.

A Tabela 4.1 resume as vantagens e desvantagens das propostas discutidas nesta seção.

Tabela 4.1. Vantagens e desvantagens das propostas de CAC discutidas na Seção 4.3.1.

Propostas descritas em:	Vantagens	Desvantagens
[42]	A solução busca um ponto de equilíbrio entre os fluxos de serviço rtPS e nrtPS, a fim de	A solução proposta considera somente duas classes de serviço rtPS e nrtPS

	oferecer largura de banda para as novas conexões. Os autores utilizam a teoria dos jogos	
[39]	Utiliza a largura de banda excedente das conexões ativas para aceitar novas conexões	A solução foi desenvolvida considerando que todos os fluxos em andamento da mesma classe possuem os mesmos requisitos de largura de banda máxima e mínima
[40]	A solução é baseada no algoritmo de empréstimo de banda e degradação gradual, objetivando maximizar a utilização da largura de banda	A solução foi desenvolvida considerando que todas as conexões pertencentes ao mesmo tipo de serviço possuem os mesmos requerimentos de largura de banda, e que a largura de banda requisitada por uma conexão rtPS não varia entre o MRTR e o MSTR
[43]	Os autores consideram o <i>handoff</i> na solução proposta e utilizam o algoritmo de empréstimo de largura de banda proporcional	Os resultados não mostraram que a solução proposta pode garantir limites para o atraso e <i>jitter</i>

4.3.2. Mecanismos de CAC que não utilizam estratégia de degradação

Os autores em [44] desenvolveram um mecanismo de CAC que utiliza como critério para admitir ou rejeitar as conexões a taxa média de transmissão, a taxa máxima sustentada e também a taxa mínima reservada. Estes autores alegam que, na maioria dos trabalhos

encontrados na literatura, apenas a taxa mínima reservada é utilizada como critério para admitir ou rejeitar as conexões. A proposta desenvolvida em [44] foi avaliada utilizando o recurso de modelagem e simulação, onde comparou-se o desempenho do mecanismo de CAC utilizando os três critérios citados anteriormente. Para todos os critérios, adotou-se a seguinte premissa: se a largura de banda disponível for maior que a taxa média do critério adotado, aceita-se a conexão. Caso contrário, rejeita-se a conexão. Os autores concluíram que, utilizando o critério de taxa média, é possível obter resultados melhores em relação à vazão e o atraso, se comparado com a taxa mínima reservada e a taxa máxima sustentada.

A solução de CAC apresentada em [45] consiste na utilização de três módulos denominados: módulo classificador de tráfego, módulo expedidor, e módulo de CAC. A principal função do módulo classificador de tráfego é de identificar as mensagens enviadas pelas SSs, verificar os parâmetros de QoS associados às mesmas e inserir estas mensagens em filas apropriadas. Feito isto, o módulo expedidor extrai as mensagens de suas determinadas filas e as envia para o módulo de CAC. O módulo de CAC recebe as mensagens e inicia o processo para verificar se aceita ou não as conexões na rede. Para tanto, o módulo de CAC utiliza como critério os parâmetros MSTR e MRTR. Caso haja recursos disponíveis, a conexão será aceita. Caso contrário, a conexão será rejeitada.

Os autores em [26] também desenvolveram um módulo de controle de admissão de conexões baseado na técnica *token bucket*. Entretanto, nenhum serviço de degradação das conexões existentes é previsto pelos autores para aceitar as conexões. Uma nova conexão é aceita se: (1) existir largura de banda suficiente para acomodar a nova conexão; (2) garantir que os requisitos de largura de banda e atraso sejam satisfeitos para as conexões das classes de serviço UGS e rtPS, e largura de banda mínima para as conexões da classe nrtPS; e (3) garantir que a QoS das conexões existentes serão mantidas. O mecanismo de CAC desenvolvido pelos autores em [26] é aplicado para as conexões das classes de serviço UGS,

rtPS e nrtPS. O CAC não é aplicado para as conexões da classe BE. Segundo os autores em [26], as conexões BE são sempre aceitas, pois tais conexões não possuem requisitos de QoS.

O algoritmo de CAC apresentado em [41] leva em consideração os parâmetros de atraso e *jitter* das conexões existentes para aceitar uma nova conexão. Como as conexões possuem diferentes requisitos de QoS, definiu-se um mecanismo para testar a admissibilidade das conexões. Este mecanismo é denominado pelos autores de *Hyper Interval* (HI). O HI representa o intervalo pelo qual o processo de admissão é executado. Além disso, os autores incluíram em seu esquema de CAC um agente para fazer uma estimativa de largura de banda e o monitoramento das filas das conexões pertencentes às classes de serviço rtPS e nrtPS. O objetivo do trabalho desenvolvido em [41] é de garantir QoS em termos de largura de banda, atraso e *jitter*. Entretanto, apenas a taxa de aceitação das conexões foi utilizada para avaliar a performance do algoritmo proposto. A Tabela 4.2 resume as vantagens e desvantagens das propostas discutidas nesta seção.

Tabela 4.2. Vantagens e desvantagens das propostas de CAC discutidas na Seção 4.3.2.

Propostas descritas em:	Vantagens	Desvantagens
[44]	Utiliza três critérios para admitir uma nova conexão: taxa média de transmissão, a taxa máxima sustentada e também a taxa mínima reservada	Quando adota-se o critério de taxa máxima sustentada, o número de SSS na rede diminui bastante em relação a taxa média e a taxa mínima reservada
[45]	Utiliza como critério para aceitar ou rejeitar as conexões os parâmetros MSTR e MRTR	Complexidade na implementação. A utilização de três módulos independentes para formar um único mecanismo de CAC pode inserir

		atrasos no mecanismo de CAC
[24]	A solução é baseada na técnica <i>token bucket</i>	Não possui flexibilidade em relação à distribuição de banda
[41]	Utilizam agente para fazer uma estimativa de largura de banda e o monitoramento das filas das conexões pertencentes às classes de serviço rtPS e nrtPS	Não apresenta resultados sobre o atraso e <i>jitter</i>

4.3.3. Mecanismos de CAC que utilizam abordagem *cross-layer*

O número de trabalhos sobre CAC, existentes na literatura, que utilizam abordagem *cross-layer* é bem menor se comparado com as estratégias de CAC que foram descritas nas seções anteriores. Em [46], foi apresentado um dos primeiros trabalhos sobre mecanismo de CAC que leva em consideração o tipo de modulação utilizado pelas SSs. Este trabalho foi então generalizado em [47]. O mecanismo de CAC proposto em [47] é baseado no modelo de *Markov*. Tal mecanismo leva em consideração o *handoff* das conexões, como também a troca de modulação das SSs para admitir ou não uma nova conexão. Entretanto, apenas dois tipos de modulação foram considerados no desenvolvimento do mecanismo de CAC. Além disso, os autores consideraram a suposição de que todas as conexões possuem os mesmos requisitos de QoS, o que limita a aplicabilidade do mecanismo.

Os autores em [48] propuseram um mecanismo de CAC denominado QoS CAC (Q-CAC). Este mecanismo foi desenvolvido com o objetivo de se adaptar a ambientes com várias MCSs. O controle de admissão das conexões é feito baseado na quantidade existente de *slots* físicos e no requisito de taxa mínima reservada das conexões admitidas. Entretanto, os autores redefinem o requisito de taxa mínima reservada para o número mínimo de *slots* reservados.

Para tanto, o mecanismo Q-CAC obtêm da mensagem DSA a taxa mínima reservada e também a modulação utilizada pela conexão. Com isso, é feito a tradução de quantidade de taxa mínima reservada para número de *slots* reservados.

O mecanismo de CAC proposto em [49] foi desenvolvido baseado em uma abordagem heurística. A largura de banda total é dividida entre as classes de serviço UGS, rtPS, nrtPS e BE. Uma nova conexão UGS ou rtPS é atendida se a somatória da largura de banda de todas as conexões UGS ou rtPS, mais os requisitos da nova conexão for menor ou igual à quantidade de largura de banda atribuída para a classe. Caso haja variações na demanda de largura de banda devido a SNR, o valor excedido é retirado da classe de serviço nrtPS. Assim sendo, a variação da SNR restringe o número de conexões nrtPS na rede, pois a largura de banda dessa classe de serviço pode ser utilizada pelas classes de serviço UGS e rtPS. A classe de serviço BE utiliza qualquer recurso que sobre das classes superiores. Os autores consideram a solução proposta apenas para o tráfego *Downlink*. A Tabela 4.3 resume as vantagens e desvantagens das propostas discutidas nesta seção.

Tabela 4.3. Vantagens e desvantagens dos mecanismos de CAC discutidos na Seção 4.3.3.

Proposta descrita em:	Vantagens	Desvantagens
[47]	Considera o <i>handoff</i> entre as conexões	Utiliza apenas dois tipos de modulação
[48]	O controle de admissão das conexões é feito baseado na quantidade existente de <i>slots</i> físicos e no requisito de taxa mínima reservada das conexões	Segundo os autores, as informações sobre a SNR são obtidas da mensagem DSA. Entretanto, como definido pelo padrão, tal mensagem não possui esta informação
[49]	Utiliza uma abordagem	A solução contempla apenas a

	heurística. A largura de banda total é dividida entre as classes de serviço	transmissão no sentido <i>Downlink</i>
--	---	--

4.4. Considerações finais

Este capítulo discutiu os mecanismos de CAC para redes IEEE 802.16. O padrão IEEE 802.16 define a utilização do mecanismo de CAC. Entretanto, o padrão não define como este mecanismo deve ser implementado.

A principal função do mecanismo de CAC é a de decidir se aceita ou não uma nova conexão na rede. Esta decisão deve ser feita baseada em questões tais como: verificar se a QoS da nova conexão será satisfeita e, garantir que a QoS das conexões existentes não seja afetada após a admissão da nova conexão.

Os requisitos que definem a eficiência do mecanismo de CAC foram destacados. Por exemplo, o baixo tempo de resposta para decidir a aceitação da conexão, e também a existência de uma margem de segurança para garantir que os requisitos negociados sejam satisfeitos são fatores importantes no desenvolvimento do mecanismo de CAC.

Este capítulo também mostrou como os mecanismos de CAC são classificados na literatura. Baseando-se nessa classificação, alguns trabalhos sobre CAC foram discutidos. Um mecanismo de CAC, que atua em conjunto com o algoritmo de escalonamento proposto nesta tese é apresentado no Capítulo 5.

Capítulo 5

ALGORITMO DE ESCALONAMENTO ADAPTATIVO COM GERENCIAMENTO DINÂMICO DE *POLLING* PARA O TRÁFEGO *UPLINK* EM REDES IEEE 802.16

5.1. Introdução

O mecanismo de escalonamento é um dos principais componentes responsáveis pela obtenção de QoS nas redes de computadores. Entretanto, o padrão IEEE 802.16 não define um algoritmo de escalonamento específico para as redes WiMAX. Além disso, o padrão define a utilização do mecanismo de *polling* para reservar largura de banda para as classes de serviço, porém, não especifica um mecanismo eficiente para tal propósito [50][51]. Para fazer um melhor uso do canal de comunicação, o padrão especifica a utilização de diferentes técnicas de codificação e modulação para adaptar o perfil de rajada de acordo com a SNR. Todavia, surge uma nova questão: como realizar eficientemente o escalonamento *uplink*, com garantias de QoS, das diversas SSs localizadas em diferentes pontos distantes da BS? Tais SSs transmitem dados com perfis de rajada diferentes, e o algoritmo de escalonamento deve garantir a QoS para as aplicações.

Além do mecanismo de escalonamento, existe outro mecanismo muito importante na obtenção de QoS, que é o mecanismo de CAC. Enquanto o mecanismo de escalonamento tem como objetivo garantir que os recursos requisitados sejam alocados de forma eficiente, o mecanismo de CAC tem como objetivo evitar que a rede seja saturada por uma quantidade excessiva de conexões, limitando o número de usuários na rede. Entretanto, como acontece com o mecanismo de escalonamento, o mecanismo de CAC também não é definido pelo padrão IEEE 802.16.

Tendo em vista as questões acima levantadas, este capítulo apresenta uma proposta de algoritmo de escalonamento adaptativo com gerenciamento dinâmico de *polling* para o tráfego *uplink* em redes IEEE 802.16. O algoritmo proposto é desenvolvido para ser totalmente dinâmico e adequado, portanto, às redes que utilizam várias MCSs. Utilizando uma abordagem *cross-layer* e as informações sobre as requisições de largura de banda enviadas pelas SSs, definiu-se um novo esquema de escalonamento baseado em *deadlines*, tendo como objetivo garantir atraso máximo limitado para as aplicações de tempo real. Além disso, o algoritmo de escalonamento interage com o mecanismo de gerenciamento de *polling* da BS, e controla a periodicidade do envio do *polling* para as classes de serviço de tempo real e não tempo real. Tal interação proporciona melhor atendimento aos requisitos das aplicações sensíveis ao atraso. Para evitar a saturação do enlace, um mecanismo de CAC foi desenvolvido para atuar em conjunto com o algoritmo de escalonamento proposto. O mecanismo de CAC também foi desenvolvido utilizando uma abordagem *cross-layer*.

Este capítulo está estruturado da seguinte forma: A Seção 5.2 descreve os desafios existentes no desenvolvimento de algoritmos de escalonamento para redes IEEE 802.16. A Seção 5.3 define o problema abordado nesta tese. Na Seção 5.4, a solução proposta é apresentada. A Seção 5.5 traz as considerações finais do capítulo.

5.2. Desafios de projeto de algoritmos de escalonamento em redes IEEE 802.16

O algoritmo de escalonamento em redes IEEE 802.16 é implementado para o tráfego *downlink* e para o tráfego *uplink*. O escalonamento do tráfego *downlink* é realizado na BS, e o escalonamento do tráfego *uplink* é realizado por dois escalonadores, um na SS e outro na BS.

Na SS, o escalonador *uplink* é responsável por distribuir a largura de banda concedida pela BS dentre as diversas conexões existentes. Neste caso, o escalonador *uplink* tem total acesso às filas das conexões, o que facilita no desenvolvimento do algoritmo de escalonamento e na provisão de QoS para as aplicações.

Na BS, o escalonamento do tráfego *uplink* é mais complexo do que o escalonamento do tráfego *downlink*. No escalonamento do tráfego *downlink*, a BS tem total acesso e conhecimento do estado das filas de cada conexão. Além disso, apenas a BS pode transmitir no sentido *downlink*. Os pacotes são enviados em *broadcast* para as SSs que, por sua vez, obtém apenas os pacotes destinados a ela. Já no escalonamento do tráfego *uplink*, as filas das conexões estão localizadas nas SSs, portanto, separadas da BS. Assim sendo, a BS não possui informações sobre o tempo de chegada dos pacotes nas filas localizadas nas SSs. Isso dificulta o desenvolvimento do algoritmo de escalonamento para o tráfego *uplink*, principalmente no que diz respeito à obtenção de QoS.

Como descrito na Seção 2.5.2, as SSs fazem acesso ao canal *uplink* através de mecanismos de requisição/concessão (*polling*). Primeiramente, a BS aloca recursos para que as SSs possam enviar suas requisições de largura de banda. O escalonador *uplink*, localizado na BS, depende das mensagens de requisição de banda enviadas pelas SSs para manter-se informado sobre o estado atual da fila de cada conexão. Através dessas mensagens, o escalonador *uplink* decide a ordem e quais requisições serão atendidas.

Todavia, o padrão define vários mecanismos de *polling*, e a escolha de qual mecanismo a ser adotado no desenvolvimento de um algoritmo de escalonamento é muito importante. O mecanismo de *polling* influencia diretamente no escalonamento, conseqüentemente, na provisão de QoS [52][53]. É responsabilidade do algoritmo de escalonamento minimizar o tempo entre o envio e o recebimento da resposta às requisições de largura de banda, como também à alocação dos recursos de forma eficiente, a fim de garantir os requisitos de atraso, *jitter* e vazão para as aplicações que possuem estas restrições.

O padrão IEEE 802.16 define a utilização do mecanismo de modulação adaptativa na camada PHY. Devido a este fato, é importante levar em consideração o perfil de rajada em uso na camada PHY durante a alocação dos recursos na camada MAC. Um usuário pode contratar um determinado serviço e, dependendo da localização desse usuário na rede, a BS poderá utilizar uma modulação mais robusta. Além disso, tal serviço poderá exigir uma alta demanda de recursos da rede, como por exemplo, a transmissão de um vídeo MPEG. O algoritmo de escalonamento deve ser capaz de garantir QoS para as aplicações em cada SS na rede de acesso, mesmo para aquelas SSs que utilizam modulação de baixa ordem. Todavia, para que tal consideração seja levada em conta, é necessário desenvolver um algoritmo de escalonamento que utilize uma abordagem *cross-layer*. Como já citado, um dos grandes desafios dos algoritmos de escalonamento é a questão da justiça (*fairness*), e no caso dos algoritmos de escalonamento *cross-layer*, tal questão se torna mais complexa.

5.3. Definição do problema

Muitos dos algoritmos de escalonamento propostos para ambientes de rede sem fio foram derivados de algoritmos de escalonamento já existentes, desenvolvidos para redes cabeadas. Entretanto, o canal de comunicação sem fio apresenta variações em suas características dependentes da atenuação no meio físico, condições de propagação etc. Assim sendo, surge a necessidade de adaptar os algoritmos de escalonamento para o ambiente de

rede sem fio, ou desenvolver novos algoritmos de escalonamento de acordo com as características do ambiente de rede sem fio.

Como descrito no Capítulo 3 muitas pesquisas foram feitas sobre o escalonamento em redes WiMAX. Entretanto, estudos revelam que o desenvolvimento de um escalonador justo, eficiente, que atenda a todos os requisitos de QoS das aplicações ainda é uma grande área de pesquisa [15 – 17] [54 – 61].

Tendo em vista o atendimento dos requisitos das aplicações sensíveis ao atraso, as pesquisas bibliográficas realizadas demonstram que, até o presente momento, não existe nenhum mecanismo eficiente de gerenciamento de *polling* que possa interagir com o mecanismo de escalonamento e contribuir para este objetivo. O padrão IEEE 802.16 especifica quais classes de serviço devem utilizar o mecanismo de *polling*, todavia, o padrão não define a periodicidade da utilização desse mecanismo. Tal questão é muito importante, pois esta periodicidade interfere diretamente no atraso máximo percebido pelas aplicações.

Para obter uma melhor utilização do enlace, o WiMAX utiliza o mecanismo de modulação adaptativa, porém, a utilização de tal mecanismo traz um novo desafio para o mecanismo de escalonamento: efetuar o escalonamento das SSs localizadas em pontos com diferentes distâncias da BS, com garantias de QoS, levando em consideração a MCS utilizada em cada SS. Tal questão é muito importante, não só para as redes WiMAX, mas também para todas as redes que utilizam o mecanismo de adaptação de enlace, pois tem um impacto direto na capacidade de prover QoS.

Levando em conta as questões acima descritas, propõe-se, nesta tese, um algoritmo de escalonamento para o tráfego *uplink* com o objetivo de atender aos requisitos de QoS das aplicações considerando: o mecanismo de adaptação de enlace definido pelo padrão e também, o gerenciamento do mecanismo de *polling* da BS, resultando no desenvolvimento de

um algoritmo de escalonamento totalmente dinâmico e adaptativo. A seguir é feita a descrição do algoritmo de escalonamento proposto nesta tese.

5.4. Proposta de algoritmo de escalonamento adaptativo com gerenciamento dinâmico de *polling* para o tráfego *uplink* em redes IEEE 802.16

Considerando os desafios de projeto apresentados na Seção 5.2 e os problemas descritos na Seção 5.3, propõe-se nesta tese, um algoritmo de escalonamento para o tráfego *uplink* em redes IEEE 802.16. O algoritmo proposto utiliza uma abordagem *cross-layer*, e tem como objetivo prover QoS para as diversas classes de serviço definidas pelo padrão. A seguir é feita a descrição da arquitetura do escalonador proposto.

5.4.1. A arquitetura do escalonador proposto

O padrão IEEE 802.16 define um método de adaptação de enlace. Esse método seleciona a modulação e a taxa de codificação de uma determinada conexão, baseando-se no estado do enlace sem fio. O padrão IEEE 802.16 também define quais modulações e taxas de codificação são suportadas pelo método de adaptação de enlace. Entretanto, o padrão não especifica com exatidão a melhor maneira de utilizá-lo. Assim sendo, o algoritmo proposto foi desenvolvido utilizando uma abordagem *cross-layer*, onde o escalonamento é executado levando em conta as informações sobre o esquema de modulação utilizado. A Figura 5.1 ilustra a arquitetura do escalonamento definida nesta tese.

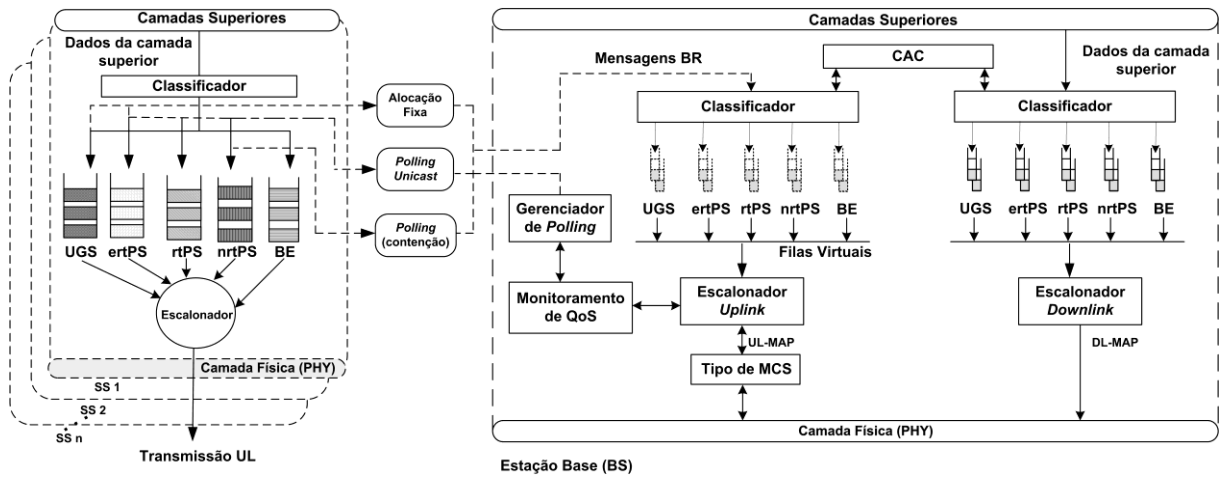


Figura 5.1: Arquitetura de escalonamento proposta.

Como pode ser visto na Figura 5.1, a arquitetura de escalonamento da BS inclui: as filas de requisição de largura de banda; o módulo de escalonamento da BS; o módulo de CAC; e dois novos componentes, um módulo denominado “Monitoramento de QoS” e um módulo denominado “Tipo de MCS”. Os módulos “Monitoramento de QoS” e “Tipo de MCS” provêm informações que são utilizadas no escalonamento das conexões. Além disso, é possível observar na Figura 5.1 o módulo de gerenciamento de *polling*, denominado “Gerenciador de *Polling*”. Esse módulo interage com o algoritmo de escalonamento proposto, e controla a periodicidade do envio do *polling unicast* para as conexões rtPS e nrtPS.

O módulo de monitoramento de QoS armazena informações sobre os recursos existentes na rede de acesso. Fazem parte dessas informações, a quantidade total de recursos disponíveis na rede, e a quantidade de recursos atribuídos para as classes de serviço. Além dessas informações, o módulo de monitoramento de QoS também mantém as informações relacionadas aos parâmetros de QoS de cada classe de serviço. Por exemplo, o atraso médio e o *jitter* médio.

O módulo “Tipo de MCS” faz a interface entre o algoritmo de escalonamento e a camada física (PHY). Esse módulo armazena informações relativas à modulação empregada na

camada física. Tais informações são utilizadas pelo algoritmo de escalonamento para efetuar o cálculo dos *deadlines* para as conexões da classe de serviço rtPS.

A BS recebe das SSs as mensagens *BR* contendo as requisições de largura de banda para a transmissão dos dados. A partir das informações sobre a modulação utilizada, o escalonador calcula os *deadlines* para as conexões rtPS, tendo em vista minimizar o atraso para os fluxos de serviço da classe rtPS. Os serviços UGS e ertPS recebem *grants* periódicos como definido pelo padrão. O serviço UGS possui alocação de banda fixa, e o serviço ertPS possui alocação de banda dinâmica. Entretanto, o serviço ertPS pode utilizar *grants* alocados para enviar mensagens *BR* indicando a quantidade de largura de banda necessária [62]. O serviço nrtPS possui garantias de largura de banda mínima, e o serviço BE não possui nenhuma garantia de QoS. A seguir é feita a descrição do escalonamento proposto.

5.4.1.1 Escalonamento da classe UGS

Para estar de acordo com o padrão IEEE 802.16, o serviço UGS recebe *grants* periódicos para o envio dos dados, possui alocação de banda fixa e maior prioridade dentre as outras classes de serviço. Assim sendo, primeiramente é feita a alocação dos recursos para o serviço UGS e depois para as classes de serviço ertPS, rtPS, nrtPS e BE. Devido à natureza periódica do tráfego UGS é possível, inicialmente, especificar a quantidade de largura de banda necessária para as transmissões. A partir disso, a BS concede *grants* periódicos para as SSs enviarem seus dados. Como ilustrado na Figura 5.2, os *grants* são enviados a cada *frame* para cada conexão UGS, evitando assim, atrasos desnecessários com requisições de largura de banda.

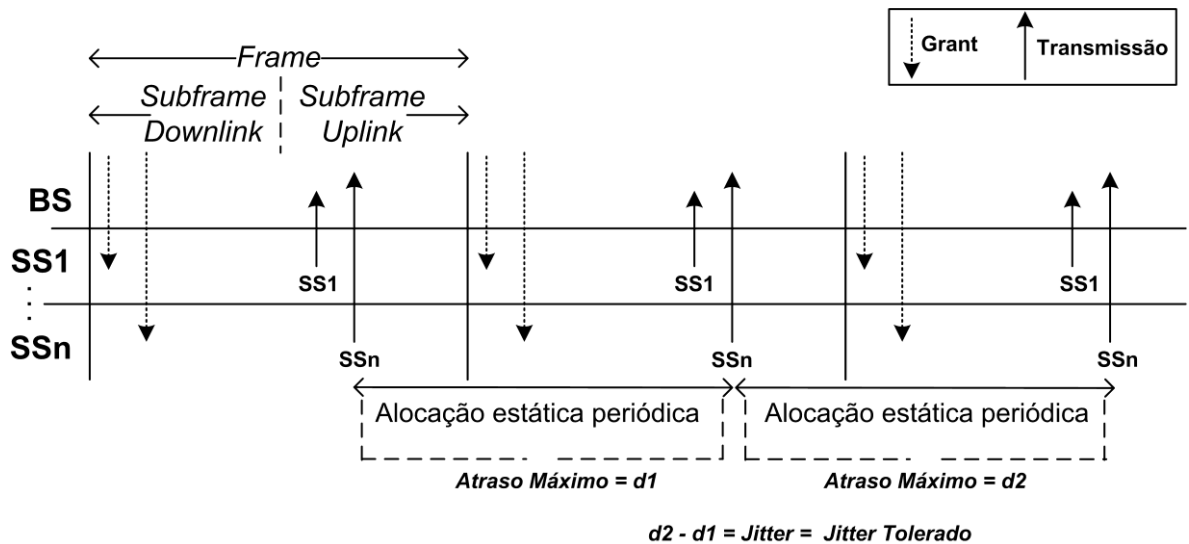


Figura 5.2: Garantia de atraso e *jitter* limitados para a classe UGS [63].

No caso do serviço UGS, assume-se que a modulação na camada física é fixa. Como descrito em [34], o modo de transmissão pode ser selecionado na fase de acesso inicial do serviço, e a partir de testes efetuados, é possível avaliar a melhor MCS a ser utilizada. Assim sendo, o modo de transmissão é fixo durante todo o tempo de serviço, o que significa que os *slots* alocados para o serviço UGS também são fixos.

O algoritmo proposto faz o escalonamento da classe de serviço UGS e depois faz o escalonamento das classes de serviço ertPS, rtPS, nrtPS e BE. Sendo assim, o algoritmo calcula a quantidade de *slots* alocados para o serviço UGS, e os subtrai da quantidade total de *slots* existentes por *frame*. A partir disso, o algoritmo distribui para as outras classes de serviço os *slots* restantes. A seguir é feita a descrição do algoritmo de escalonamento proposto para a classe de serviço ertPS.

5.4.1.2 Escalonamento da classe ertPS

De acordo com o padrão, o serviço ertPS combina algumas características dos serviços UGS e rtPS. Para evitar atrasos com requisições de largura de banda, a BS provê *grants* para o serviço ertPS da mesma forma como ocorre no serviço UGS. Entretanto, o serviço UGS

possui alocação de banda fixa, já o serviço ertPS possui alocação de banda dinâmica. Periodicamente, os *grants* alocados para o serviço ertPS podem ser utilizados para o envio de mensagens de requisição de largura de banda [12]. Assim sendo, a alocação de banda para este serviço ocorre de acordo com o recebimento das mensagens *BR*, e também de acordo com a MCS utilizada. O algoritmo calcula a quantidade de *slots* alocados para o serviço ertPS, e a subtrai da quantidade total de *slots* existentes por *frame*. A classe de serviço ertPS possui maior prioridade que as classes de serviço rtPS, nrtPS e BE.

5.4.1.3 Escalonamento da classe rtPS

A classe de serviço rtPS recebe da BS *grants* periódicos para enviar suas mensagens de requisição de largura de banda. Assim sendo, a BS deve reservar parte da largura de banda para o mecanismo de *polling unicast*. Todavia, o padrão não especifica a periodicidade do envio do *polling unicast* para a classe de serviço rtPS. Adicionalmente, o escalonador deve atender os requisitos de atraso máximo limitado e os requisitos de largura de banda mínima para esta classe de serviço.

5.4.1.3.1 Garantias de atraso máximo limitado e largura de banda mínima para a classe rtPS

O atraso máximo limitado, para a classe de serviço rtPS, é garantido pelo algoritmo de escalonamento proposto através do uso de um novo esquema baseado em *deadlines*, desenvolvido para as aplicações de tempo real. O escalonador atribui um *deadline* para cada mensagem de requisição de largura de banda recebida na fila da BS. Os *deadlines* são calculados levando em consideração os seguintes parâmetros: as informações sobre as MCSs utilizadas entre a SS e a BS; as informações sobre a quantidade de largura de banda requisitada pela SS; e o atraso de fila das mensagens de requisição de largura de banda na BS.

Devido à consideração destes parâmetros utilizados no cálculo dos *deadlines*, o algoritmo proposto é caracterizado como dinâmico.

Supondo que M_i represente a i -ésima mensagem de requisição de largura de banda de uma conexão rtPS existente na fila da BS, a expressão (5.1) permite calcular o i -ésimo *deadline*:

$$deadline_i = TT_i + QD_i, \quad \forall_i | BR_i \quad (5.1)$$

A descrição dos parâmetros utilizados em (5.1) é feita a seguir:

- TT_i : é o atraso de transmissão, o qual é calculado para cada mensagem de requisição de largura de banda enviada pelas SSs. Tal cálculo é feito baseado na MCS que está sendo utilizada na camada física e, de acordo com a expressão (5.2):

$$TT_i = \left(\frac{BR_size \times 8}{bpsymbol} \right) \times symbol_time \quad (5.2)$$

Onde: BR_size define a quantidade de bytes requisitada pelas aplicações para a transmissão dos dados no sentido *uplink*. Esta informação é obtida das mensagens de requisição de largura de banda enviadas pelas SSs, e reportam o tamanho de fila atual das conexões das SSs; $bpsymbol$ é a quantidade de bits/símbolo utilizada na transmissão. Este parâmetro é dependente da MCS utilizada e representa a eficiência da modulação. A BS seleciona a modulação de acordo com o valor da SNR; e $symbol_time$ é o tempo de duração de um símbolo OFDM ou OFDMA;

- QD_i : é o atraso de fila de cada requisição de largura de banda na fila da BS. Esse parâmetro é calculado a partir do instante de envio da mensagem BR pela SS.

Esse parâmetro é dependente do número de mensagens de requisição de largura de banda existente na fila de requisição da BS.

O serviço rtPS é desenvolvido para dar suporte ao tráfego de tempo real. Além das restrições temporais, esse tráfego tem como característica a geração periódica de pacotes de dados com tamanhos variáveis. O algoritmo de escalonamento para a classe rtPS é apresentado na Figura 5.3.

Algoritmo de Escalonamento do serviço rtPS

```
1: Verifique as mensagens rtPS de requisição de largura de banda na fila da BS;
2: Para cada mensagem  $BR_i$  na fila da BS faça
3:   Se ( $BR[i].length > availableBW$ ) então
4:     break;
5:    $deadline[i] = Deadline\_Calculation(BR[i].length)$ ;
6:   Se ( $deadline[i] > L_{SU}$ ) então
7:     descarte  $BR[i]$ ;
8:     break;
9:    $availableBW -= BR[i].length$ ;
10:   $BW[i] += BR[i].length$ ;
11: ordene o vetor  $deadline[i]$  pelo menor  $deadline$ 
12:  $QoSMonitoring(i)$ ;
```

Figura 5.3: Algoritmo de escalonamento do serviço rtPS.

Como pode ser observado na Figura 5.3, primeiramente, o algoritmo verifica se existe largura de banda disponível para ser alocada (linha 3). Se houver, o algoritmo calcula um *deadline* para cada mensagem de requisição de largura de banda (linha 5) como definido na expressão (5.1). Nas transmissões rtPS, as SSs enviam suas mensagens de requisição de largura de banda em um dado *frame*, e os dados são enviados no *frame* subsequente. Assim sendo, é necessário verificar se os *deadlines* das conexões rtPS não expirarão no próximo *frame*. Neste caso, foi definido no algoritmo proposto, um parâmetro denominado L_{SU} . Esse parâmetro representa o tamanho do *subframe uplink*, sendo utilizado para verificar se o *deadline* calculado para uma determinada conexão não expirará no próximo *frame* (linhas 6 e 7). Desta forma, é possível descartar, previamente, as mensagens de requisição de largura de

banda cujos *deadlines* expirarão no próximo *frame*. Caso o *deadline* não expire no próximo *frame*, o algoritmo atualiza a largura de banda disponível (linha 9), e armazena na variável $BW[i]$ (linha 10), a quantidade de largura de banda requisitada pela conexão.

Baseando-se na política de escalonamento do algoritmo EDF, o algoritmo de escalonamento do serviço rtPS cria uma lista contendo as mensagens de requisição de largura de banda ordenada pelo menor *deadline* (linha 11). As informações contidas nessa lista definem a ordem de transmissão das conexões do serviço rtPS. Essas informações são inseridas na mensagem de controle UL-MAP, a qual é enviada a cada *frame*, para as SSs pelo canal de comunicação *downlink*. A Figura 5.4 ilustra o fluxograma do algoritmo para a classe rtPS.

A cada *frame*, a BS executa o algoritmo de escalonamento e verifica, considerando uma determinada janela de duração T , se a quantidade de largura de banda atribuída para o serviço rtPS está dentro de limites mínimos pré-definidos. Isso é feito utilizando as seguintes informações do módulo de monitoramento: o requerimento mínimo de largura de banda, e a quantidade de largura de banda recebida na janela corrente de duração T . A Figura 5.5 ilustra o algoritmo do módulo de monitoramento da BS.

Como pode ser visto na Figura 5.5, é verificado, no módulo de monitoramento de QoS, se a largura de banda do serviço rtPS está dentro dos limites mínimos requisitados (linha 3). Se estiver, a BS mantém a configuração inicial do mecanismo de *polling*. Caso contrário, o módulo de monitoramento de QoS executará o método *Adjust_periodicity_polling()* (linha 4), que tem como objetivo fazer um balanceamento do *polling unicast* para ajustar a largura de banda mínima. Isso é feito diminuindo o intervalo de *polling* para a classe rtPS, com isso, reduz-se o intervalo de *polling unicast* enviado para a classe rtPS. A largura de banda mínima é definida pela variável *MinimumReservedTrafficRate* (linha 3), que expressa a taxa mínima requerida em bps.

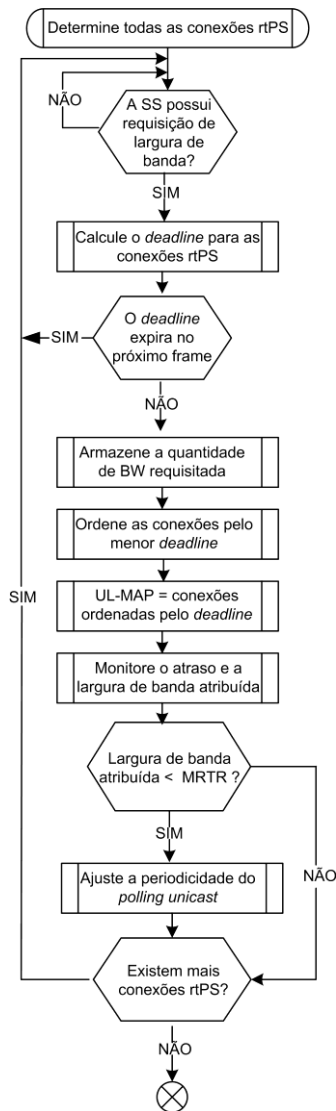


Figura 5.4: Fluxograma do algoritmo de escalonamento rtPS.

QoSMonitoring(CID)

- 1: **Para** cada conexão **faça**
 - 2: **se** (servico[cid] = rtPS ou nrtPS) **então**
 - 3: granted_tmp_bw[cid] = BW[cid];
 - 4: **se** (granted_tmp_bw[cid] < *MinimumReservedTrafficRate*) **então**
 - 5: *Adjust_periodicity_polling*(cid);
-

Figura 5.5: Algoritmo do módulo de monitoramento de QoS.

A Figura 5.6 ilustra o algoritmo de ajuste do intervalo de *polling* utilizado pelo módulo de monitoramento de QoS.

Adjust_periodicity_polling(cid)

```
1: Para cada conexão faça
2:   se (serviço[cid] = rtPS) então
3:     se ( $P_{overhead} < overhead\_threshold$ ) então
4:       polling_interval_rtPS -=  $\alpha$ ;
5:   se (serviço[cid] = nrtPS) então
6:     atraso_estimado = (atraso_estimado * 0.9) + (atraso_amostrado * 0.1);
7:     se (atraso_estimado <  $rtPS\_threshold$ ) então
8:       polling_interval_nrtPS -=  $\alpha$ ;
9:     senão
10:      polling_interval_nrtPS +=  $\alpha$ ;
11:      use_contention_polling();
```

Figura 5.6: Algoritmo de ajuste do intervalo de *polling*.

Para o serviço rtPS, o intervalo de *polling* é diminuído desde que a sobrecarga causada pela utilização do *polling unicast* esteja dentro de um limiar pré-definido (linha 2). O intervalo de *polling* é representado pelo símbolo “ α ”, sendo este configurado de forma dinâmica. A definição desse intervalo como também o cálculo da sobrecarga do *polling* ($P_{overhead}$) é mostrada na Seção 5.4.1.5. Além disso, como pode ser observado na Figura 5.6, o módulo de monitoramento utiliza uma média móvel exponencialmente ponderada (MMEP) para fazer a estimativa do atraso médio para o serviço rtPS (linha 7). A Figura 5.7 ilustra uma comparação entre o atraso estimado e o atraso amostrado em função do tempo de simulação das transmissões rtPS.

O atraso estimado é utilizado pelo módulo de monitoramento de QoS para ajustar o intervalo de *polling* para a classe de serviço nrtPS. O algoritmo da classe de serviço nrtPS e BE é descrito a seguir.

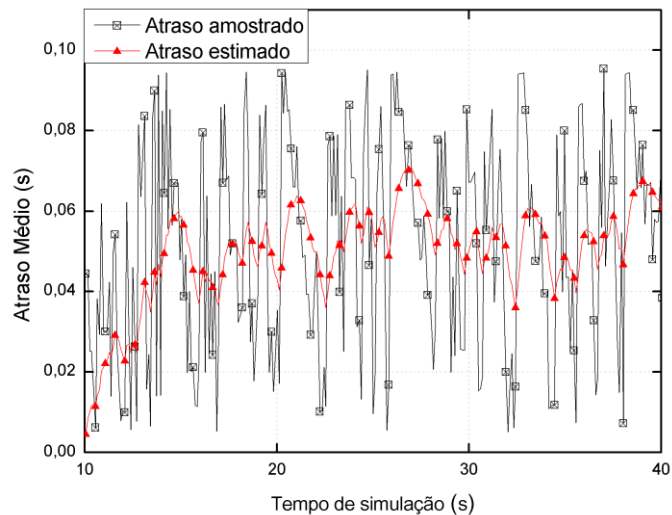


Figura 5.7: Atraso estimado vs. atraso amostrado [65].

5.4.1.4. Escalonamento da classe nrtPS e BE

O algoritmo de escalonamento da classe nrtPS é bem simples se comparado ao algoritmo de escalonamento da classe rtPS, uma vez que as aplicações dessa classe não possuem restrições temporais. A Figura 5.8 ilustra o algoritmo da classe nrtPS.

Algoritmo de escalonamento nrtPS

- 1: Verifique as mensagens nrtPS de requisição de largura de banda na fila da BS;
 - 2: **Para** cada mensagem BR_i na fila de BS **faça**
 - 3: **Se** $BR[i].length < availableBW$ **então**
 - 4: $BW[i] += BR[i].length$;
 - 5: $availableBW -= BR[i].length$;
 - 6: **Senão** $backlogged_tmp_BW[i] = BR[i].length$
 - 7: $QoSMonitoring(i)$;
-

Figura 5.8: Algoritmo de escalonamento nrtPS.

O serviço nrtPS oferece suporte para aplicações não tempo real, mas que necessitam de largura de banda mínima. De acordo com o padrão IEEE 802.16, o serviço nrtPS pode utilizar o mecanismo de *polling* baseado em contenção e também *polling unicast*. Entretanto, este último é utilizado com menor frequência para o serviço nrtPS se comparado ao serviço rtPS. Todavia, poucas pesquisas [15][16][17] focam o balanceamento do envio de *polling unicast*

entre os serviços rtPS e nrtPS, e o padrão não especifica a frequência de envio do *polling unicast* para tais serviços.

Primeiramente, o algoritmo verifica se existe largura de banda disponível (linha 3). Caso positivo, a largura de banda é atribuída (linha 4), e a largura de banda restante é atualizada (linha 5). Caso negativo, a requisição ficará na fila para ser escalonada no próximo *frame* (linha 6). Através do módulo de monitoramento de QoS (linha 7), o algoritmo verifica se a quantidade de largura de banda atribuída para a classe nrtPS está dentro dos limites pré-definidos (linha 4 da Figura 5.5). Se não estiver, o módulo de monitoramento aciona o método *Adjust_periodicity_polling()* para verificar a possibilidade de aumentar a frequência de envio do *polling unicast* para a classe nrtPS (linha 5 da Figura 5.5). O método *Adjust_periodicity_polling()* primeiramente verifica se o atraso estimado da classe rtPS está dentro dos limites pré-definidos (linha 7 da Figura 5.6). Se estiver, então é possível aumentar a frequência do *polling unicast* para a classe nrtPS. Desta forma, o controle da periodicidade do envio do *polling unicast* para a classe nrtPS é feito de forma dinâmica. Se o atraso estimado da classe rtPS estiver acima dos limites pré-estabelecido, então a classe nrtPS utiliza o *polling* baseado em contenção. A Figura 5.9 ilustra o fluxograma do algoritmo de escalonamento da classe de serviço nrtPS.

Em relação ao escalonamento das conexões do serviço BE, uma vez que este serviço não oferece garantias de QoS, enquanto houver recursos disponíveis, eles serão atribuídos ao serviço BE utilizando o algoritmo RR. No entanto, caso não haja recursos disponíveis, o serviço BE deixará de ser atendido, pois como definido pelo padrão, o serviço BE não oferece nenhuma garantia de QoS para as aplicações.

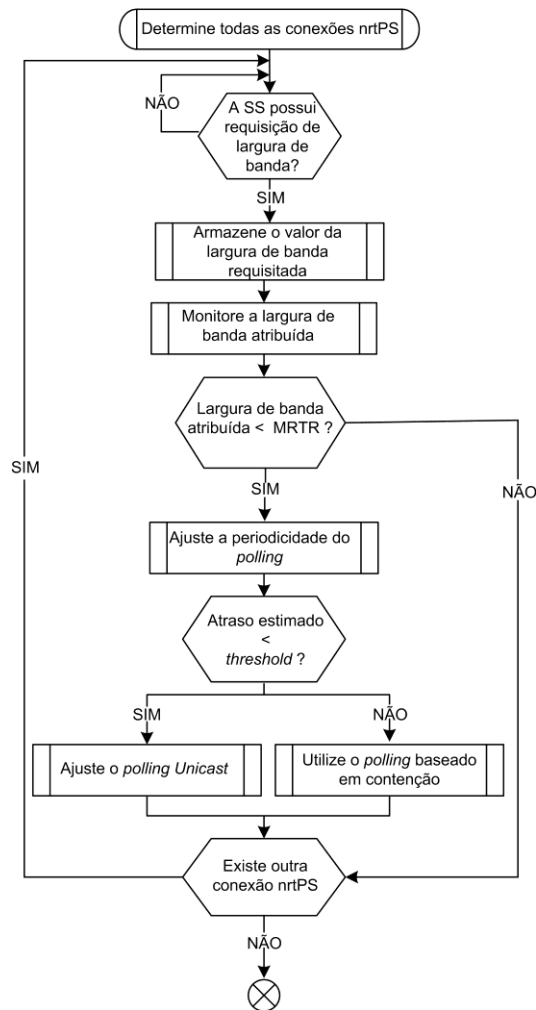


Figura 5.9: Fluxograma do algoritmo de escalonamento nrtPS.

5.4.1.5. Gerenciamento dinâmico do *polling*

Uma questão muito importante a ser considerada no escalonamento é o atraso de escalonamento decorrente da utilização do mecanismo de *polling* [17]. O gerenciamento do *polling* desenvolvido nesta tese, que busca a minimização deste atraso, é descrito a seguir.

5.4.1.5.1. O gerenciamento do mecanismo de *polling unicast*

O mecanismo de *polling unicast* define o intervalo de alocação consecutiva de *slots* para que as SSs possam enviar suas requisições de largura de banda. Esse mecanismo pode influenciar diretamente nos valores de atraso percebido pelas aplicações de tempo real, tendo

em vista que as SSs devem esperar por um intervalo de tempo que começa no momento que recebem os pacotes da camada superior, e termina quando transmitem para a camada física.

O mecanismo de gerenciamento de *polling* calcula o intervalo máximo de *polling* para as conexões da classe rtPS e nrtPS. Para a classe rtPS, o intervalo máximo de *polling* é calculado de acordo com o requisito de atraso máximo limitado da conexão rtPS. Já para as conexões nrtPS, o intervalo de *polling* é calculado de acordo com o requisito de largura de banda mínima que deve ser garantida. A expressão (5.3) é utilizada para calcular o intervalo máximo de *polling* para o serviço rtPS (IPu_i):

$$IPu_i = AtrasoMaximo - DuraçãoFrame \quad (5.3)$$

Onde: IPu_i representa o intervalo máximo de *polling unicast* em milissegundos (*ms*) calculado para o serviço rtPS; *AtrasoMaximo* é o parâmetro de QoS que representa o atraso máximo tolerado pela conexão rtPS em *ms*. Já o parâmetro *DuraçãoFrame*, representa a duração do *frame*, que pode variar entre 2,5 e 20 *ms*. Por exemplo, considerando-se um valor do atraso máximo igual 100 *ms*, e a duração do *frame* igual a 20 *ms*, utilizando a expressão (5.3), é definido um valor de intervalo máximo de *polling* igual a 80 *ms*. Neste caso, o *polling unicast* será efetuado para a conexão rtPS dentro de um intervalo máximo de 4 *frames* consecutivos.

A expressão (5.3) também é utilizada para calcular o intervalo máximo de *polling unicast* para as aplicações da classe de serviço nrtPS. Entretanto, como a classe de serviço nrtPS não possui restrição temporal, o intervalo de *polling* é inicialmente definido com um valor de aproximadamente 1 segundo.

Uma vez definido o intervalo de *polling* para as conexões rtPS e nrtPS, pode ser calculada (em porcentagem), a sobrecarga de *polling unicast*. Seja n o número total de

conexões rtPS ou nrtPS, a expressão (5.4) é utilizada para calcular a sobrecarga de *polling unicast*:

$$P_{overhead} = \sum_{i=1}^n \frac{1}{IP_{ui} * U_{i_{fps}} \left(\frac{slots}{frame} \right) * fps} \quad (5.4)$$

Onde: IP_{ui} é o intervalo de *polling* da i -ésima conexão rtPS ou nrtPS; $U_{i_{fps}}$ é a quantidade de *slots* por *frame* utilizado pelo i -ésimo usuário; e fps é a quantidade de *frames* por segundo. Com isso, o módulo de monitoramento verifica a sobrecarga causada pela utilização do *polling unicast*, e com isso, é capaz de fazer o gerenciamento do mecanismo de *polling unicast* de forma mais eficiente.

A variação mínima e máxima do intervalo do *polling unicast* é definida dentro do intervalo $[P_{i_{mínimo}}, P_{i_{máximo}}]$, onde $P_{i_{mínimo}}$ representa o intervalo mínimo, entre *frames*, para o envio do *polling unicast*, e $P_{i_{máximo}}$ define o valor máximo, entre *frames*, para o envio do *polling unicast*. O intervalo mínimo do *polling unicast* pode chegar a ser de até um *frame*. Entretanto, tal intervalo é controlado pela sobrecarga da utilização do *polling unicast*, pois, quanto menor for o intervalo do *polling unicast*, maior será a sobrecarga. O valor máximo é definido pela expressão (5.3) como previamente descrito. Além disso, considerando as conexões nrtPS, é possível utilizar o valor da sobrecarga do *polling unicast* para fazer um balanceamento entre a utilização dos mecanismos de *polling unicast* e *polling* baseado em contenção.

5.4.1.6. O mecanismo de CAC

É desejável que se utilize um mecanismo de CAC em conjunto com o escalonamento para garantir que a rede não seja saturada. Assim sendo, um mecanismo de CAC foi desenvolvido para ser utilizado em conjunto com o algoritmo de escalonamento proposto. O desenvolvimento do mecanismo de CAC é descrito a seguir.

5.4.1.6.1. Descrição do mecanismo de CAC

O mecanismo de CAC, desenvolvido para ser utilizado em conjunto com o algoritmo de escalonamento proposto, possui uma abordagem *cross-layer*, sendo capaz de se adaptar em ambientes de rede que utilizam várias MCSs. Além disso, o mecanismo de CAC utiliza a técnica de reserva de banda e degradação. Para tanto, a largura de banda total é dividida em três faixas, como ilustrado na Figura 5.10.

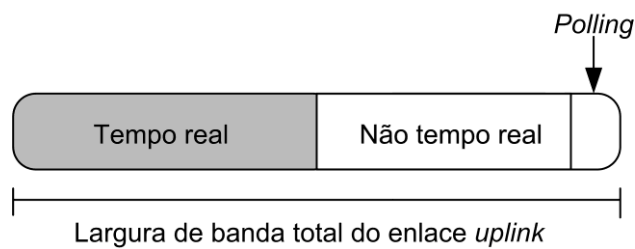


Figura 5.10: Largura de banda total do enlace *uplink*.

Como pode ser observado na Figura 5.10, a primeira faixa é destinada às aplicações de tempo real, a segunda faixa é destinada previamente às aplicações sem restrições temporais, e a última faixa é reservada para o *polling unicast*.

A Figura 5.10 ilustra como a largura de banda é dividida no sistema. Entretanto, este recurso é computado em termos de número de *slots* disponíveis. Assim sendo, o critério adotado para aceitar ou rejeitar as conexões é baseado na taxa mínima reservada (MRTR) para as conexões ertPS, rtPS e nrtPS, que é traduzida em número mínimo de *slots* reservados, e a taxa máxima sustentada (MSTR) para as conexões UGS.

Para estabelecer uma nova conexão *uplink*, as SSs enviam mensagens DSA para a BS. Uma vez recebida uma mensagem DSA, a BS aciona o mecanismo de CAC, que inicia o procedimento para verificar se aceita ou não tal conexão. Tal procedimento implica em verificar se a BS possui recursos suficientes para prover a QoS requerida e, ao mesmo tempo, garantir que a QoS das conexões já existentes será mantida. Utilizando a mensagem DSA, o mecanismo de CAC primeiramente verifica o tipo de conexão (UGS, ertPS, rtPS, nrtPS ou

BE) e, em seguida, obtém a informação sobre os requisitos de QoS que devem ser garantidos. Utilizando as informações da camada física, o mecanismo de CAC verifica a modulação utilizada e traduz o requisito MRTR, ou MSTR no caso das conexões UGS, em número mínimo de *slots* requerido. Isso é feito de acordo com a expressão 5.5:

$$NMslots_{ij} = \left(\frac{BR_size \times 8}{bpsymbol} \right) \quad (5.5)$$

Onde: $NMslots_{ij}$ é o “Número Mínimo” de *slots* requeridos pela conexão *i* associada ao tipo de serviço *j*; BR_size é a quantidade de bytes requisitada pelas SSs para a transmissão dos dados no sentido *uplink*; $bpsymbol$ é a quantidade de bits/símbolo utilizada na transmissão. Esse parâmetro é dependente da MCS utilizada e representa a eficiência da modulação. A BS seleciona a MCS de acordo com os valores de SNR recebidos das SSs.

O mecanismo de CAC proposto utiliza a técnica de reserva de banda e degradação. O esquema de reserva de banda proposto é ilustrado graficamente na Figura 5.11.

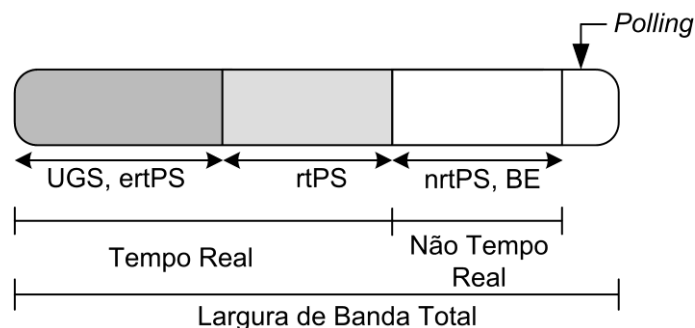


Figura 5.11: Divisão da largura de banda.

Como já descrito, a largura de banda total é traduzida em número de *slots*, os quais são divididos em *slots* reservados para aplicações de tempo real ($slots_{TR}$), *slots* reservados para aplicações sem restrições temporais ($slots_{NTR}$) e *slots* reservados para o mecanismo de *polling unicast*. Assim sendo, os *slots* são logicamente separados e pré-reservados para as conexões de tempo real (UGS, ertPS, rtPS), para as conexões não tempo real (nrtPS e BE), e também para o mecanismo de *polling*.

A reserva de largura de banda é utilizada para evitar a “inanição” dos fluxos de serviço que possuem menor prioridade, e também para evitar a subutilização dos recursos. A técnica de degradação é utilizada para diminuir a largura de banda alocada para as conexões já admitidas, tendo como objetivo, acomodar um número maior de conexões admitidas na rede. A degradação é aplicada somente para os serviços rtPS e nrtPS. Quando existem poucas conexões no sistema, essas conexões utilizam a taxa máxima de tráfego sustentado (MSTR). Entretanto, quando o número de conexões admitidas no sistema aumenta, a largura de banda alocada para as conexões diminui. A técnica de degradação é aplicada gradualmente para todas as conexões das classes de serviço (rtPS e nrtPS). Entretanto, a degradação é limitada pela somatória do requisito MRTR das conexões rtPS e nrtPS. Assim sendo, a largura de banda alocada para as conexões dos serviços rtPS e nrtPS varia de acordo com a carga de tráfego na rede.

Para garantir o requisito MRTR e controlar o número de *slots* pré-determinado para as conexões rtPS e nrtPS, dois limiares foram definidos, os quais são representados pelas expressões 5.6 e 5.7

$$L_{(rtPS)} = \sum_{i=0}^n r_{\min_rtPS}(i) \quad (5.6)$$

$$L_{(nrtPS)} = \sum_{j=0}^n r_{\min_nrtPS}(j) \quad (5.7)$$

Onde $L_{(rtPS)}$ e $L_{(nrtPS)}$ são os limiares definidos para os serviços rtPS e nrtPS; r_{\min_rtPS} e r_{\min_nrtPS} representam o número mínimo de *slots* requeridos pelos serviços rtPS e nrtPS. Assim, as expressões 5.6 e 5.7 representam o número mínimo de *slots* alocados para os serviços rtPS e nrtPS. A Figura 5.12 mostra os limiares definidos.

Uma vez calculado o número mínimo de *slots* requeridos, o mecanismo de CAC verifica se existem *slots* disponíveis para aceitar a nova conexão. A Figura 5.13 apresenta o algoritmo

de CAC proposto. Os critérios utilizados para aceitar ou rejeitar uma nova conexão são descritos a seguir.

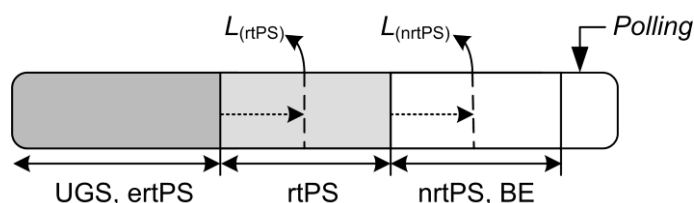


Figura 5.12: Limiares definidos para as classes de serviço rtPS e nrtPS.

Uma nova conexão UGS é aceita pela BS se haver largura de banda suficiente para garantir o requisito MSTR. Quando uma nova conexão UGS chega à BS, o mecanismo de CAC obtém o valor do requisito MSTR. Esse requisito é traduzido em número de *slots* (linha 3), e o mecanismo de CAC verifica se existem *slots* disponíveis para aceitar a conexão (linha 4). Se existirem *slots* disponíveis, a conexão UGS é aceita (linha 5), e o valor do número de *slots* disponíveis é atualizado (linha 6). Se não existirem *slots* disponíveis, a conexão UGS será rejeitada (linha 8). O procedimento utilizado para verificar se aceita ou não as conexões ertPS é similar ao procedimento utilizado para a classe UGS. Entretanto, no caso das conexões ertPS, é verificado pelo mecanismo de CAC se a BS possui recursos disponíveis para garantir o requisito MRTR (linhas 9 – 15).

Se uma nova requisição de conexão rtPS chega à BS, o mecanismo de CAC verifica se a BS possui recursos disponíveis para garantir o requisito MRTR. Para tanto, o mecanismo de CAC obtém o requisito MRTR, sendo o mesmo traduzido em números de *slots* (linha 17). Com isso, o mecanismo de CAC verifica se existem *slots* disponíveis para aceitar a conexão (linha 18). Entretanto, nesse caso, se a quantidade de *slots* definida para as aplicações de tempo real (*slotsTR*) não for suficiente, o mecanismo de CAC verifica a possibilidade de se utilizar os *slots* reservados para as aplicações sem restrições temporais (*slotsNRT*) (linha 23). Em ambas as verificações, se existir *slots* disponíveis, é avaliado se o atraso estimado das conexões rtPS está abaixo do limite predefinido (linhas 19 e 24).

Algoritmo de CAC

```
1: Para cada nova conexão faça
2:   Se serviço[i] = UGS então
3:      $r_{\max\_UGS}[i] = \text{getNumberSlots}(\text{MSTR});$ 
4:     Se  $r_{\max\_UGS}[i] < (\text{slotsTR} - L_{(\text{rtPS})})$  então
5:       aceita a conexão;
6:        $\text{slotsTR} = \text{slotsTR} - r_{\max\_UGS}[i];$ 
7:     Senão
8:       rejeita a conexão;

9:   Se serviço[i] = ertPS então
10:     $r_{\min\_ertPS}[i] = \text{getNumberSlots}(\text{MRTR});$ 
11:    Se  $r_{\min\_ertPS}[i] < (\text{slotsTR} - L_{(\text{rtPS})})$  então
12:      aceita a conexão;
13:       $\text{slotsTR} = \text{slotsTR} - r_{\min\_ertPS}[i]$ 
14:    Senão
15:      rejeita a conexão;

16:  Se serviço[i] = rtPS então
17:     $r_{\min\_rtPS}[i] = \text{getNumberSlots}(\text{MRTR});$ 
18:    Se  $r_{\min\_rtPS}[i] < (\text{slotsTR} - L_{(\text{rtPS})})$  então
19:      Se atraso estimado rtPS < threshold então
20:        aceita a conexão;
21:         $L_{(\text{rtPS})} += r_{\min\_rtPS}[i];$ 
22:      Senão
23:        Se  $r_{\min\_rtPS}[i] < (\text{slotsNTR} - L_{(\text{nrtPS})})$  então
24:          Se atraso estimado rtPS < threshold então
25:            aceita a conexão;
26:             $L_{(\text{rtPS})} += r_{\min\_rtPS}[i]$ 
27:          Senão
28:            rejeita a conexão;

29:  Se serviço[i] = nrtPS então
30:     $r_{\min\_nrtPS}[i] = \text{getNumberSlots}(\text{MRTR});$ 
31:    Se  $r_{\min\_nrtPS}[i] < (\text{slotsNTR} - L_{(\text{nrtPS})})$ 
32:      aceita a conexão;
33:       $L_{(\text{nrtPS})} += r_{\min\_nrtPS}[i];$ 
34:    Senão
35:      rejeita a conexão;
```

Figura 5.13: Algoritmo de CAC.

Se o atraso estimado estiver dentro dos limites pré-definidos, a conexão é aceita (linhas 20 e 25), e os parâmetros $L_{(\text{rtPS})}$ são atualizados (linhas 21 e 26). Caso contrário, e conexão rtPS será rejeitada (linha 28).

Se uma nova conexão nrtPS chega a BS, o mecanismo de CAC obtém o parâmetro MRTR e o traduz em número de *slots* (linha 30). Feito isso, o mecanismo de CAC verifica se existem *slots* disponíveis para aceitar a nova conexão (linha 31). Se existirem *slots* disponíveis, a conexão é aceita, e o parâmetro $L_{(nrtPS)}$ é atualizado. Caso contrário, a conexão será rejeitada. As conexões BE não possuem nenhum requisito de QoS. Assim sendo, todas as conexões BE são aceitas desde que haja recurso de largura de banda disponível. Entretanto, as conexões BE possuem menor prioridade dentre as outras classes de serviço.

5.5. Considerações finais

Este capítulo introduziu uma nova proposta de algoritmo de escalonamento de pacotes com gerenciamento dinâmico de *polling* para o tráfego *uplink* em redes IEEE 802.16. Primeiramente, alguns problemas relevantes ao desenvolvimento de algoritmos de escalonamento para redes sem fio foram destacados. Após isso, apresentou-se a proposta de algoritmo de escalonamento. A solução proposta segue as especificações do padrão IEEE 802.16.

O algoritmo de escalonamento proposto interage com o mecanismo de *polling* da BS. Assim sendo, apresentou-se o desenvolvimento do mecanismo de gerenciamento dinâmico do *polling* da BS. Tal mecanismo tem como objetivo definir, de forma dinâmica, o intervalo de alocação consecutiva de *slots* para que as SSs possam enviar suas requisições de largura de banda para a BS.

Para evitar que os recursos da rede de acesso não sejam saturados, apresentou-se o desenvolvimento de um mecanismo CAC para ser utilizado em conjunto com o algoritmo de escalonamento proposto. Tal mecanismo utiliza uma abordagem *cross-layer* e foi desenvolvido utilizando as técnicas de reserva de banda e degradação. Uma análise de desempenho dos algoritmos propostos é apresentada no Capítulo 6.

Capítulo 6

AVALIAÇÃO DA PROPOSTA DE ALGORITMO DE ESCALONAMENTO ADAPTATIVO COM GERENCIAMENTO DINÂMICO DE *POLLING* PARA O TRÁFEGO *UPLINK* EM REDES IEEE

802.16

6.1. Introdução

Neste capítulo, avalia-se por meio de modelagem e simulação o desempenho do algoritmo de escalonamento proposto. O algoritmo será avaliado em cenários que utilizam o mecanismo de modulação adaptativa, e também em cenários que utilizam apenas um mecanismo de modulação. Além disso, o algoritmo proposto será comparado com outros algoritmos de escalonamento encontrados na literatura, tais como o RR, WRR, EDF, e com os algoritmos de escalonamento que utilizam abordagem *cross-layer*, tais como o mSIR e o mmSIR [66][67][68].

Este capítulo está organizado da seguinte maneira: A Seção 6.2 apresenta as principais características sobre a modelagem e simulação. A Seção 6.3 faz uma análise do desempenho

do algoritmo proposto por meio de modelagem e simulação. A Seção 6.4 contém as considerações finais acerca deste capítulo.

6.2. Modelagem e simulação

A avaliação do algoritmo de escalonamento, proposto no Capítulo 5, é realizada por meio de modelagem e simulação. A ferramenta de simulação utilizada foi o NS-2 (*Network Simulator*) [7], em conjunto com o módulo que implementa a camada MAC do padrão IEEE 802.16 desenvolvido pelos autores em [67]. Este módulo foi originalmente desenvolvido com as seguintes classes de serviço: UGS, rtPS e BE. Assim sendo, foi necessário implementar as classes de serviço ertPS e nrtPS, o mecanismo de *polling unicast* que interage com o escalonador, o módulo de monitoramento e também o mecanismo de CAC descrito no Capítulo 5.

O cenário de simulação consiste em uma BS e várias SSs distribuídas ao redor da BS de maneira aleatória. A Figura 6.1 ilustra o cenário utilizado na simulação.

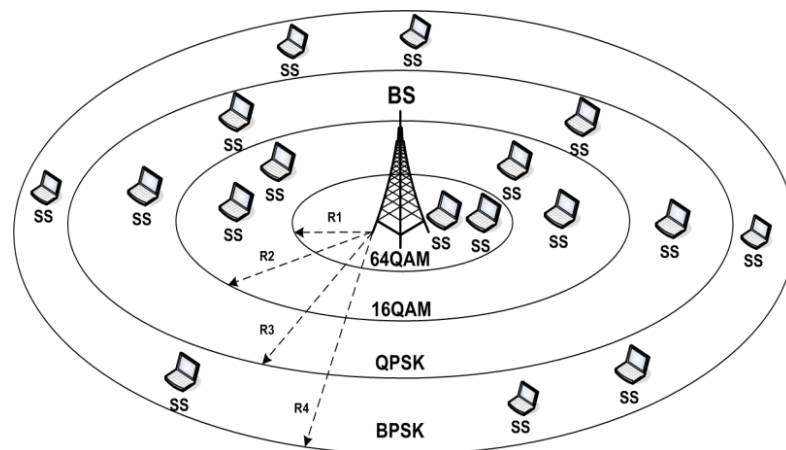


Figura 6.1: O cenário utilizado na simulação.

Como pode ser observado na Figura 6.1, o cenário de simulação foi dividido em círculos concêntricos que delimitam a área de convergência de cada esquema de modulação. Para calcular a área de convergência para cada esquema de modulação, é necessário determinar um raio máximo R_i entre a BS e as SSs que utilizam a modulação correspondente [69]. Os

cálculos utilizados para determinar a área de convergência como também os raios foram realizados como definido em [70] e [71]. Esses raios são determinados através do valor da relação sinal/ruído (SNR). Diferentes valores de SNR para diferentes técnicas de modulação foram calculados em [72] e são mostrados na Tabela 6.1. Tais valores foram utilizados no cenário de simulação.

Tabela 6.1. Níveis de modulação, codificação e SNR.

Modulação	Codificação	SNR(dB)
BPSK	1/2	3,0
QPSK	1/2	6,0
	3/4	8,5
16QAM	1/2	11,5
	3/4	15,0
64QAM	2/3	19,0
	3/4	21,0

Os experimentos de simulação têm como objetivo analisar o comportamento do algoritmo proposto em uma rede onde utiliza-se uma variedade de MCSs, como também, em uma rede onde utiliza-se apenas um tipo de modulação. Dessa forma, é possível verificar o efeito da utilização de vários perfis de rajadas na rede de acesso e verificar a eficiência do algoritmo proposto nos dois ambientes de comunicação. As técnicas de modulação utilizadas pelas SSs variam de acordo com a SNR, como definido na Tabela 6.1. A Tabela 6.2 mostra os principais parâmetros utilizados na simulação [67].

Tabela 6.2 Principais parâmetros utilizados na simulação.

Parâmetro	Valor
Frequência de Operação	3,5 GHz
Largura de Banda	5 MHz
Duplexação	TDD
Modelo de Propagação	<i>Two Ray Ground</i>
Antena	Omnidirecional
Duração do quadro	20 ms
<i>Cyclic Prefix (CP)</i>	0,25
Largura de banda <i>downlink</i>	6,16 Mbps
Largura de banda <i>uplink</i>	9,25 Mbps
Tempo de Simulação	60 s

6.2.1. Características das fontes de tráfego utilizadas na simulação

Os tipos de tráfego utilizados na simulação foram: voz sem supressão de silêncio, voz com supressão de silêncio, vídeo, FTP e Web, os quais foram mapeados, respectivamente, para as classes de serviço UGS, ertPS, rtPS, nrtPS e BE. Os tráfegos foram gerados a partir de um Agente Gerador de Tráfego (*Traffic Generating Agent - TGA*) desenvolvido para o módulo WiMAX [73]. O tráfego de voz sem supressão de silêncio foi gerado por uma fonte que gera periodicamente pacotes de 160 bytes a cada 20 ms, produzindo uma taxa média de 64kbps. O tráfego de voz com supressão de silêncio foi gerado por uma fonte *on/off*. O tempo médio de duração do período *on* considerado foi de 1,2 segundos e o período *off* considerado foi de 1,8 segundos. Durante os períodos *on* foram gerados pacotes de 66 bytes a cada 20 ms, seguindo uma distribuição exponencial. O tráfego de vídeo foi obtido por uma fonte de tráfego que gera periodicamente pacotes de tamanhos diferentes, simulando o tráfego MPEG. Os tamanhos dos pacotes variam entre 450 a 1500 bytes. O tráfego FTP foi gerado utilizando uma fonte com distribuição exponencial e comprimento médio de pacotes de 512 Kbytes. Para o tráfego Web, utilizou-se uma fonte de tráfego que gera rajadas em intervalos fixos produzindo uma taxa média de 380 kbps, e utilizou-se uma distribuição híbrida lognormal/pareto.

O intervalo entre os *grants* para os serviços UGS e ertPS é de 20 ms. O intervalo de *polling unicast* para as classes de serviço rtPS e nrtPS é controlado de forma dinâmica. Cada simulação foi executada dez vezes, cada qual com semente diferente para gerar o intervalo de confiança de 95% usando o método de replicação.

6.3. Apresentação e análise dos resultados obtidos

6.3.1. Comparação entre os algoritmos de escalonamento RR, WRR e o algoritmo proposto

Este experimento de simulação tem por objetivo avaliar o comportamento do algoritmo proposto num ambiente onde ocorrem várias transmissões com diferentes MCSs. Nesse caso, utilizou-se um cenário com várias SSs localizadas em pontos com diferentes distâncias da BS, o que realmente ocorre na rede de acesso de um ISP (*Internet Service Provider*) típico. Nesse experimento, não houve a interação do algoritmo proposto com o mecanismo de *polling* da BS. Com isso, pode-se avaliar o comportamento do algoritmo proposto sem essa funcionalidade. A rede simulada inclui uma BS e 20 SSs com uma conexão rtPS por SS. A MCS utilizada foi definida de acordo com distância entre a BS e as SSs. O tráfego de vídeo gerado na origem variou de 200 a 400 kbps por conexão rtPS, e o número de SSs ativas variou de 5 a 20. O gráfico da Figura 6.2 ilustra o comportamento do atraso médio entre as SSs e a BS em função do número de SSs que transmitiram dados utilizando diversas MCSs [74].

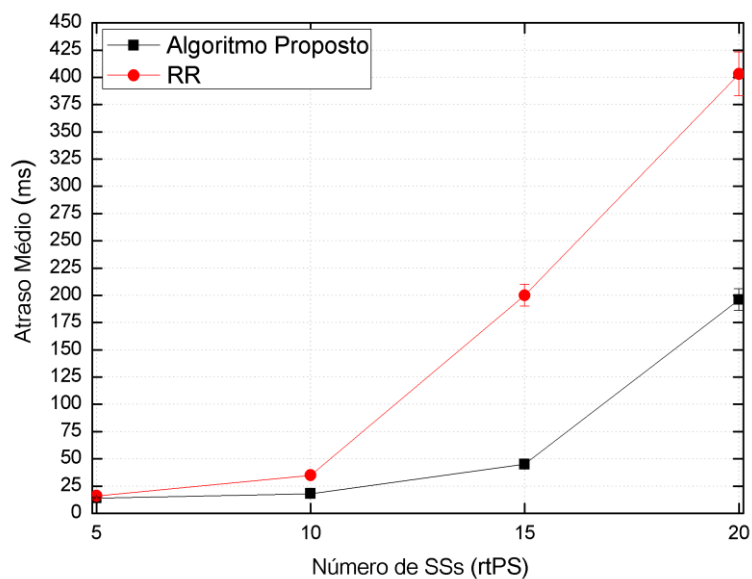


Figura 6.2: Atraso médio vs. número de SSs com conexões rtPS.

Executou-se esse experimento com o algoritmo de escalonamento *Round Robin* (RR) e também com o algoritmo proposto. Como pode ser observado no gráfico da Figura 6.2, em ambos os casos, houve um aumento expressivo no atraso médio com o aumento do número de SSs ativas na rede de acesso. Esse aumento se justifica pela saturação do enlace. Entretanto, é possível notar que o aumento do atraso médio com o aumento do número de SSs foi mais significativo quando se utilizou a disciplina de escalonamento RR. Isso ocorreu devido à característica dessa disciplina, que utiliza uma lista circular para definir a ordem das transmissões sem considerar as restrições temporais das aplicações. Em relação ao algoritmo proposto, o atraso médio menor se comparado ao algoritmo RR, se deve a sua característica de priorizar as transmissões em função do *deadline* atribuído para cada conexão, minimizando o atraso, e com isso, mostrando-se mais eficiente.

O objetivo do próximo experimento é avaliar o comportamento do algoritmo proposto num cenário com as mesmas características de tráfego de origem do experimento anterior, porém, considerando-se a interação do algoritmo proposto com o mecanismo de *polling*. Além disso, considerou-se o mecanismo de CAC para evitar a saturação do enlace. Dessa forma, é possível verificar como o algoritmo proposto aloca os recursos entre as SSs da mesma classe num ambiente com a sobrecarga controlada. Nesse experimento, comparou-se o desempenho do algoritmo de escalonamento proposto com dois algoritmos de escalonamento tradicionais, RR e WRR, os quais são frequentemente descritos nos estudos comparativos e trabalhos relacionados. A Figura 6.3 mostra o comportamento do atraso médio entre as SSs e a BS, em função do número de SSs que transmitiram dados [74].

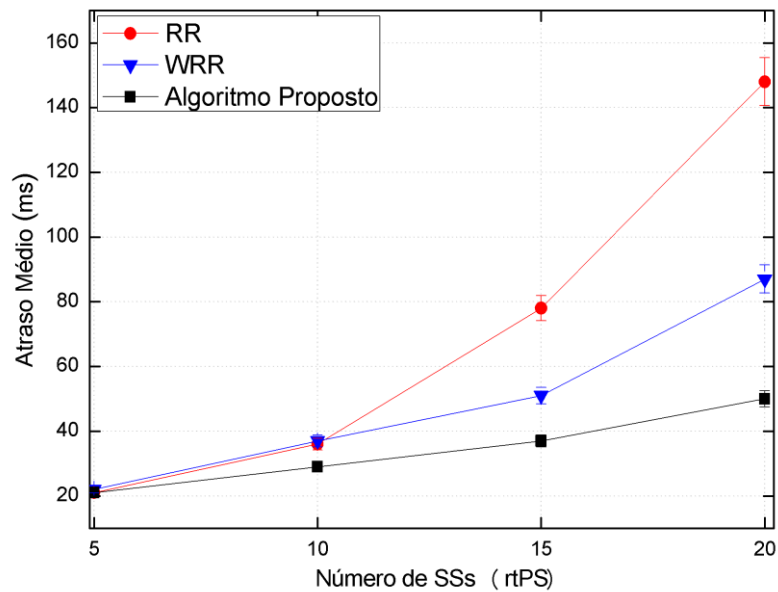


Figura 6.3: Atraso médio vs. número de SSs com conexões rtPS. Algoritmo utilizados RR, WRR e algoritmo proposto.

Primeiramente, é possível observar no gráfico da Figura 6.3 que o atraso médio permaneceu relativamente baixo se comparado com a Figura 6.2. Isso ocorreu porque, neste caso, utilizou-se o mecanismo de CAC para evitar que o enlace fosse totalmente saturado. Além disso, também pode ser observado na Figura 6.3 que o algoritmo de escalonamento proposto apresentou um melhor desempenho se comparado com os algoritmos RR e WRR. Isso ocorreu devido a sua característica de priorizar as transmissões de acordo com o menor *deadline*, levando em consideração a MCS utilizada, e também pela utilização do mecanismo de gerenciamento de *polling*, que faz o controle do intervalo de *polling* de acordo com os parâmetros de QoS. É possível observar também que o algoritmo proposto apresenta melhor desempenho num ambiente sem sobrecarga.

A Figura 6.4 ilustra o comportamento do *jitter* médio em relação ao aumento do número de SSs na rede de acesso, com conexões rtPS [74].

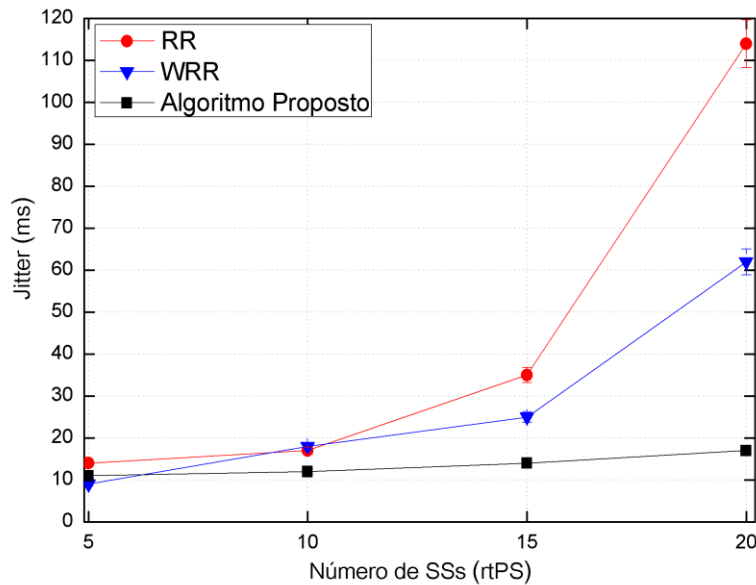


Figura 6.4: *Jitter* médio vs. número de SSs com conexões rtPS. Algoritmo utilizados RR, WRR e algoritmo proposto.

Como pode ser observado na Figura 6.4, o *jitter* médio do algoritmo proposto manteve-se baixo e com um comportamento mais estável, se comparado com os algoritmos WRR e RR, mesmo se tratando de um ambiente com várias MCSs. Tal comportamento se justifica pelo fato do algoritmo proposto fazer o escalonamento dos dados de acordo com os seus *deadlines*, como também pelo fato de fazer o controle do gerenciamento do *polling unicast*. Neste caso, o algoritmo proposto mantém o atraso estável, o que implica em manter o *jitter* também estável.

O objetivo do próximo experimento de simulação é verificar o comportamento da vazão do serviço nrtPS em função do aumento do número de SSs com conexões rtPS. Neste caso, o cenário é composto por 10 SSs com uma conexão UGS por SS, 10 SSs com uma conexão nrtPS por SS, 15 SSs com uma conexão BE por SS. O número de SSs ativas variou de 5 a 20 SSs com uma conexão rtPS por SS. O tráfego gerado na fonte variou entre 200 e 400 Kbps. Foi definida a largura de banda mínima de 55 Kbps para a classe nrtPS. A Figura 6.5 ilustra a variação da vazão em função da carga de tráfego para as classes de serviço nrtPS e BE obtida com o algoritmo proposto [74].

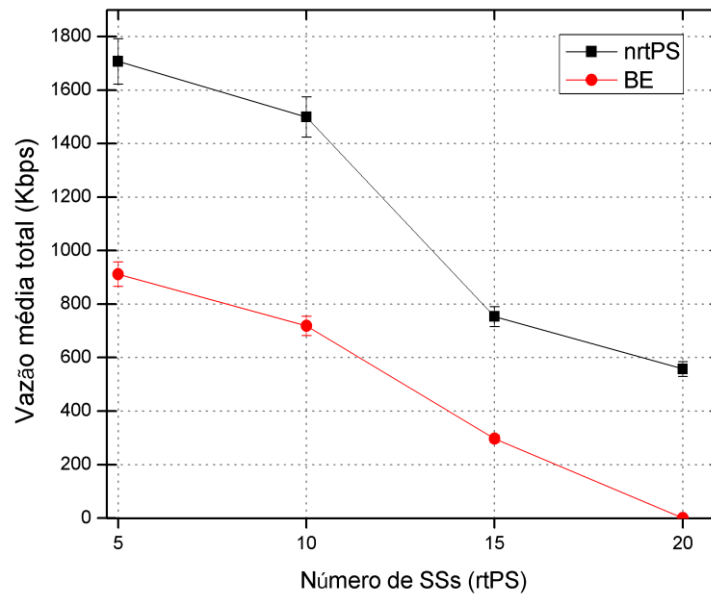


Figura 6.5: Vazão das classes nrtPS e BE com o algoritmo proposto vs. carga de tráfego rtPS.

Como pode ser visto na Figura 6.5, conforme a carga de tráfego rtPS foi aumentando, a vazão das classes de serviço nrtPS e BE foi diminuindo. Este comportamento justifica-se porque o algoritmo proposto distribui os recursos previamente alocados para as classes de serviço nrtPS e BE para as classes de serviço de tempo real, neste caso o serviço rtPS. Entretanto, para garantir a largura de banda mínima para a classe nrtPS, o algoritmo de escalonamento interagiu com o mecanismo de *polling* da BS. Com isso, as SSs com conexões nrtPS continuaram obtendo vazão da rede de acesso com a rede saturada, e as SSs com conexões BE não conseguiram transmitir mais dados. Isso ocorreu porque o recurso restante foi alocado para manter o serviço nrtPS.

6.3.2. Comparação entre os algoritmos de escalonamento EDF e o algoritmo proposto

Diferentemente dos experimentos da Seção 6.3.1, este experimento de simulação tem como objetivo verificar o desempenho do algoritmo proposto num ambiente com transmissões de dados utilizando apenas uma MCS e também transmissões de dados com

várias MCSs. Desta forma, é possível analisar se o *deadline* definido no algoritmo proposto é eficiente em ambos os ambientes. A rede de acesso simulada inclui uma BS e 30 SSs, com uma conexão rtPS por SS. As MCSs utilizadas pelas SSs variaram, de acordo com a distância entre a BS e as SSs. O número de SSs ativas variou de 5 a 30. Neste experimento utilizou-se o mecanismo de CAC para evitar que o enlace fosse totalmente saturado. A Figura 6.6 apresenta os resultados de simulação para apenas uma MCS (64QAM 1/2) dentro da área de cobertura da BS [75].

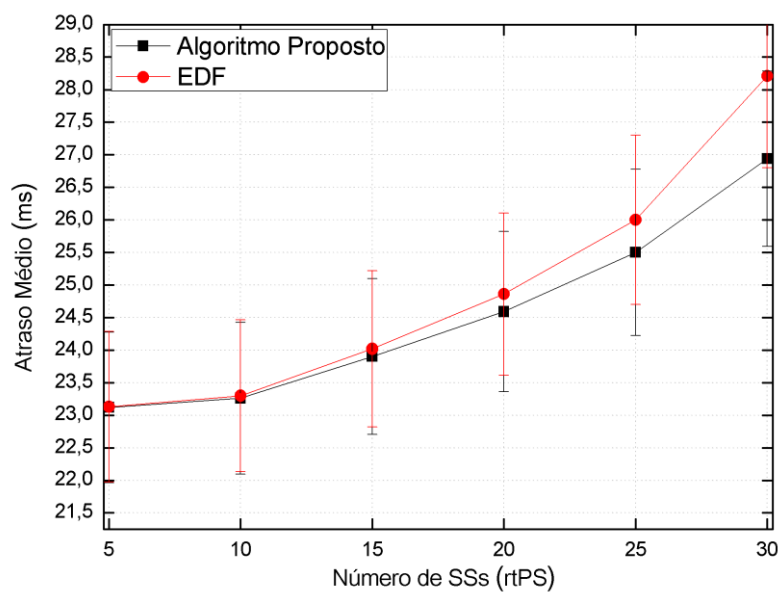


Figura 6.6: Atraso médio vs. número de SSs com uma MCS na rede de acesso.

Este experimento foi realizado com o algoritmo proposto e com o algoritmo EDF. Com isso, foi possível comparar o comportamento do algoritmo proposto com um algoritmo de escalonamento bastante referenciado nos trabalhos relacionados, desenvolvido para aplicações de tempo real, que também é baseado em *deadlines*. O algoritmo de escalonamento EDF seleciona, dentre os pacotes enfileirados, aquele com o menor *deadline*. Quanto mais tempo o pacote permanecer na fila, maior será a sua prioridade, pois o seu *deadline* estará mais próximo de expirar. Como a BS não possui informações sobre a chegada dos pacotes nas filas das SSs, considerou-se como *deadline* o tempo de chegada das requisições *BR* na fila da BS

somado com o requisito de atraso máximo das conexões. Como pode ser observado na Figura 6.6, o algoritmo proposto obteve um melhor desempenho, embora a diferença do atraso médio entre o algoritmo proposto e o algoritmo EDF seja pequena. Todavia, isso mostra que o *deadline* definido no algoritmo proposto é compatível com o algoritmo EDF original. Este *deadline* foi calculado utilizando apenas o tempo de transmissão das mensagens *BR* somado com o atraso de fila de requisição, uma vez que utilizou-se apenas uma MCS. A Figura 6.7 ilustra o atraso médio em um ambiente com várias MCSs na rede de acesso [75].

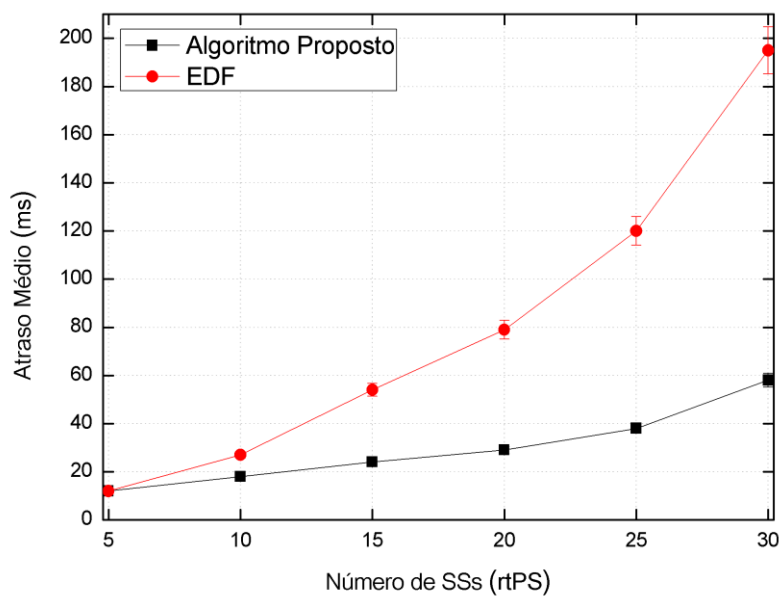


Figura 6.7: Atraso médio vs. número de SSs (rtPS) com várias MCSs na rede de acesso.

Primeiramente, é possível observar no gráfico da Figura 6.7 que o atraso médio ocorrido com o algoritmo EDF é muito superior se comparado ao atraso médio do algoritmo EDF da Figura 6.6. Isso ocorreu porque, no segundo caso, existem várias SSs na rede de acesso utilizando diferentes MCSs na transmissão dos dados. O atraso de transmissão de uma SS que utiliza, por exemplo, a modulação QPSK, é superior se comparado com outras modulações (16QAM, 64QAM). Em relação ao algoritmo proposto, é possível observar na Figura 6.7 que o atraso médio permaneceu relativamente baixo. Isso se deve ao fato de que o algoritmo proposto calculou um *deadline* para cada mensagem *BR*, de acordo com a MCS utilizada e o

atraso da fila de requisição. Dessa forma, obteve-se um desempenho superior quando comparado com o algoritmo EDF original, sendo possível minimizar o atraso na rede de acesso. A abordagem adotada no desenvolvimento do algoritmo proposto permite organizar o *subframe uplink* com os diferentes perfis de rajada de uma maneira mais eficiente. É possível observar, através dos resultados ilustrados nas Figuras 6.6 e 6.7, que o algoritmo proposto é apropriado em ambos os ambientes, especialmente quando são utilizadas várias MCSs na rede de acesso.

6.3.3. Comportamento das classes UGS e rtPS

O objetivo deste experimento é analisar o comportamento do escalonamento das classes UGS e rtPS de acordo com o aumento da carga de tráfego UGS. Para este propósito, o cenário de simulação consiste de 10 SSs com uma conexão rtPS por SS; 20 SSs com uma conexão UGS por SS, onde as conexões ativas da classe UGS variaram de 5 a 20; 10 SSs com uma conexão nrtPS e 10 SSs com uma conexão BE. Definiu-se um limiar de 100 ms para o atraso máximo da classe rtPS. A Figura 6.8 ilustra o atraso médio para as classes UGS e rtPS [65].

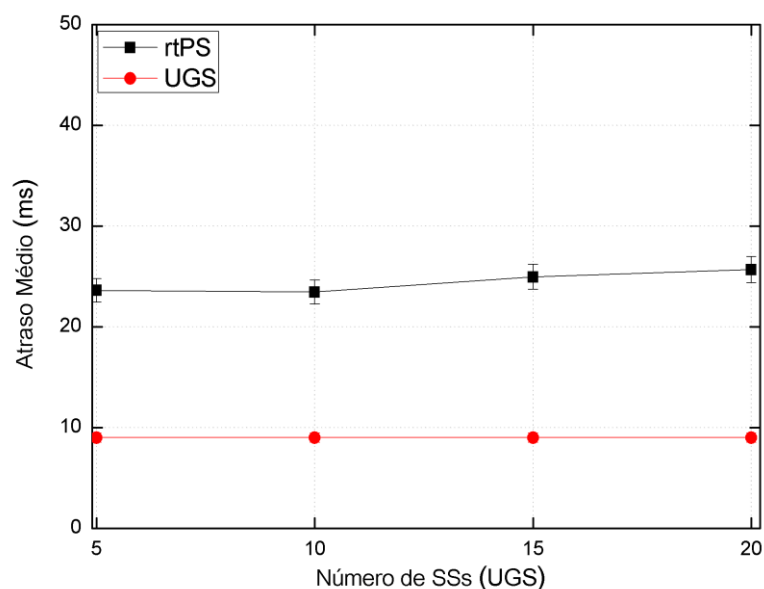


Figura 6.8: Atraso médio das classes UGS e rtPS.

Como pode ser observado na Figura 6.8, mesmo diante da variação na carga de tráfego da classe UGS, o atraso médio da classe rtPS permaneceu dentro dos limites estipulados pelo limiar. O atraso da classe UGS permaneceu praticamente estável, e não foi afetado pelo aumento da carga de tráfego. Este comportamento já era esperado, tendo em vista que o algoritmo proposto aloca os recursos para a classe UGS de forma fixa, e distribui o restante dos recursos para as outras classes de serviço.

O experimento descrito a seguir tem como objetivo avaliar a vazão e o atraso dos serviços UGS e rtPS. Entretanto, diferentemente do experimento anterior, os parâmetros de interesse foram analisados em função do crescimento da carga de tráfego rtPS. Para esse propósito, utilizou-se 15 SSs com uma conexão UGS por SS. Considerou-se também fontes de tráfego CBR onde, para cada conexão UGS, o tráfego na fonte é de 134 kbps. Além disso, utilizou-se 25 SSs com uma conexão rtPS por SS, com conexões ativas que variam entre 5 a 25 SSs, 10 SSs com uma conexão nrtPS por SS e 10 SSs com uma conexão BE por SS. Os serviços nrtPS e BE foram utilizados como tráfego de *background*, e definiu-se um limiar de 100 ms para a classe rtPS. A Figura 6.9 ilustra a vazão média total dos serviços rtPS e UGS em função da carga de tráfego rtPS [75].

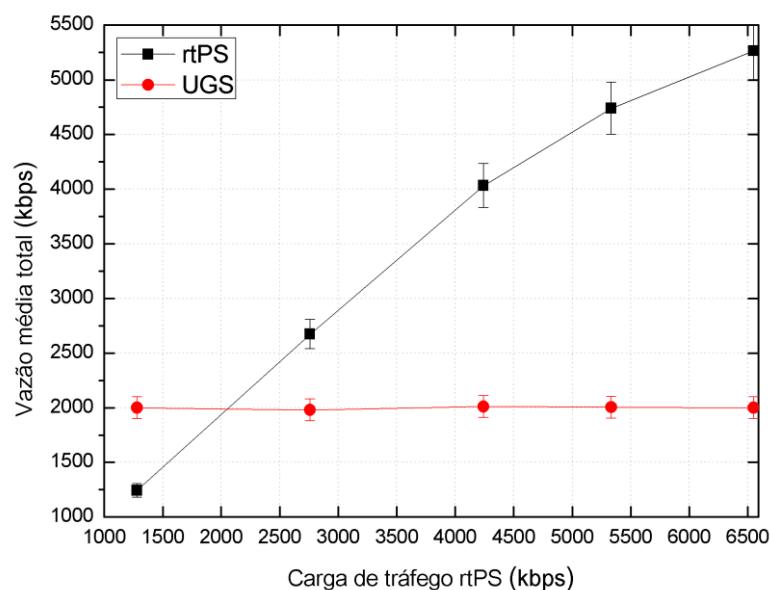


Figura 6.9: Vazão média total dos serviços rtPS e UGS vs. a carga de tráfego rtPS.

É possível observar na Figura 6.9 que o aumento da carga de tráfego rtPS não interferiu na vazão média UGS, que se manteve constante como definido pelo padrão. A vazão média para o serviço rtPS também apresentou resultados satisfatórios, mesmo com o enlace saturado em aproximadamente 70%. A diferença entre a carga de tráfego do serviço rtPS e a vazão é baixa. A Figura 6.10 ilustra o atraso médio dos serviços UGS e rtPS em função do número de SSs que geram tráfego rtPS. Como pode ser observado na Figura 6.10, o atraso médio do serviço rtPS apresentou um aumento de acordo com o crescimento da carga de tráfego rtPS. Entretanto, o valor do atraso médio permaneceu inferior em relação ao limiar definido. O atraso médio do serviço UGS não foi afetado com o aumento da carga de tráfego rtPS. Isso significa que o escalonador foi capaz de prover largura de banda para o serviço UGS em intervalos fixos, como requerido pelo padrão.

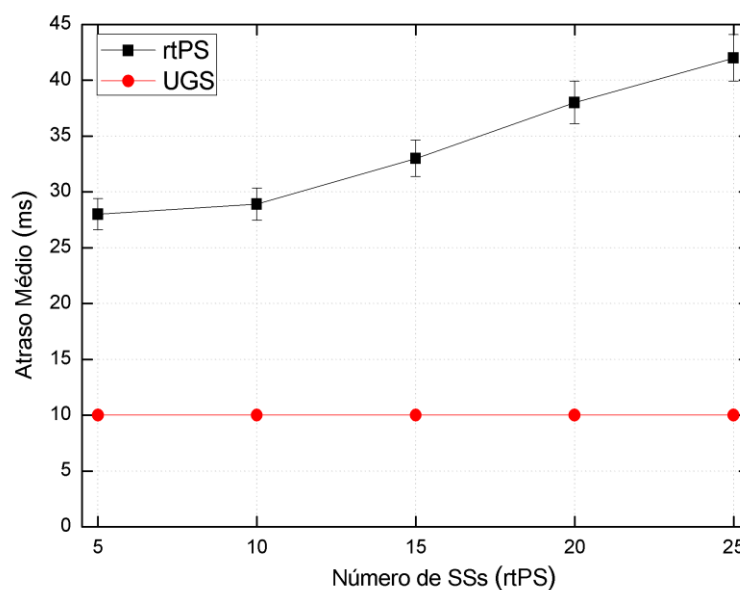


Figura 6.10: Atraso médio dos serviços UGS e rtPS em função do número de SSs que geram tráfego rtPS.

6.3.4. Comportamento das classes UGS, rtPS e ertPS

O objetivo deste experimento é investigar o comportamento das classes UGS, rtPS e ertPS de acordo com o aumento da carga de tráfego da classe rtPS. Para tanto, a rede simulada inclui 1 BS, 10 SSs com uma conexão UGS por SS, 10 SSs com uma conexão ertPS por SS, e 25 SSs com uma conexão rtPS por SS, nas quais o número de conexões ativas varia de 5 a 25. As classes nrtPS e BE foram utilizadas como tráfego de *background*. O atraso médio estimado foi definido com o valor de 100 ms para a classe rtPS. A Figura 6.11 ilustra o atraso médio das classes UGS, rtPS e ertPS [76]. Como pode ser observado na Figura 6.11, o atraso médio da classe rtPS apresentou uma pequena variação quando a carga de tráfego rtPS aumentou. Entretanto, o atraso médio dos valores da classe rtPS permaneceu menor em relação ao limiar de 100 ms definido. O atraso médio das classes UGS e ertPS permaneceu praticamente estáveis. Uma pequena diferença ocorreu entre a classe UGS e ertPS. O escalonador atribuiu recursos de forma fixa para a classe UGS. Isso também ocorreu para a classe ertPS, entretanto, neste caso, os recursos são atribuídos mediante a requisição de largura de banda.

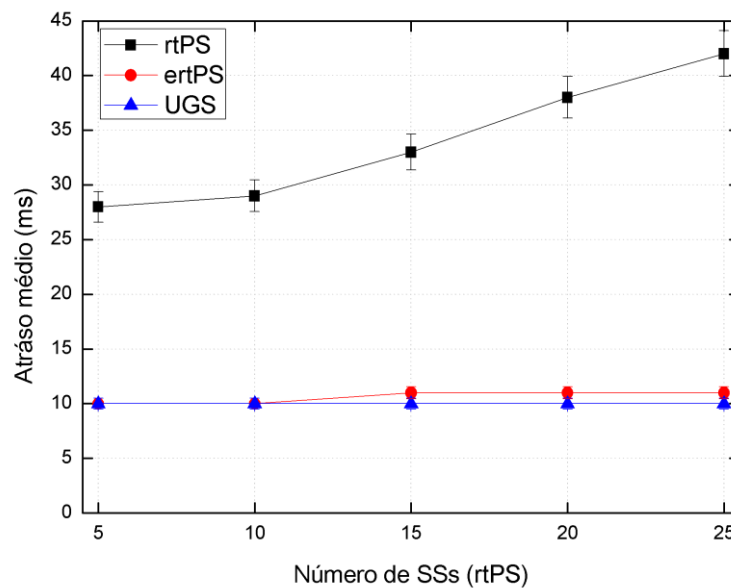


Figura 6.11: Atraso médio dos serviços UGS, rtPS, ertPS vs. o número de SSs que geram tráfego rtPS.

6.3.5. Análise do comportamento do algoritmo de escalonamento proposto em função da MCS utilizada

Este experimento tem como objetivo analisar o comportamento do algoritmo proposto de acordo com o esquema de modulação utilizado. As características do tráfego utilizado e o número de SSs utilizadas foram as mesmas do primeiro experimento definido na Seção 6.3.2. Entretanto, as SSs utilizaram as seguintes MCSs: 64QAM 3/4, 64QAM 1/2, 16QAM 3/4, 16QAM 1/2, QPSK 3/4, QPSK 1/2. A modulação BPSK foi utilizada apenas para enviar mensagens de gerenciamento entre a BS e a SS. A Figura 6.12 ilustra o atraso médio em função do tipo de MCS utilizada [76].

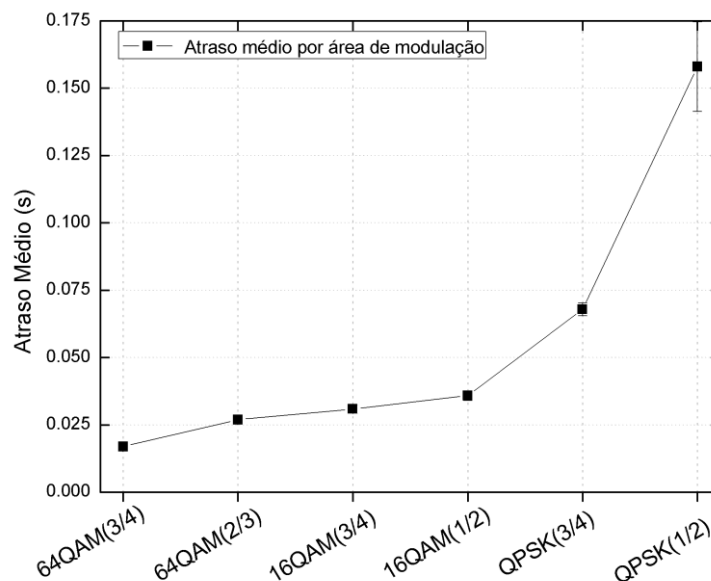


Figura 6.12: Atraso médio em função da modulação utilizada.

Como pode ser observado na Figura 6.12, considerando as modulações 64QAM e 16 QAM, houve pouca variação no atraso médio em função do tipo de MCS utilizada. Isso significa que o algoritmo proposto conseguiu distribuir os recursos de forma justa dentro das diferentes áreas de modulação. A maior diferença ocorreu quando utilizou-se a técnica de

modulação QPSK(1/2). Neste caso, o atraso de transmissão das SSs é maior devido à necessidade de utilizar maior quantidade de símbolos para transmitir a informação, influenciando diretamente no atraso médio deste grupo de SSs.

6.3.6. Comparação entre o algoritmo de escalonamento proposto, e os algoritmos *cross-layers* mSIR e mmSIR

O algoritmo de escalonamento proposto utiliza informações da camada física para efetuar o escalonamento. Este experimento tem como objetivo comparar o algoritmo proposto com os algoritmos mSIR, e mmSIR, que também utilizam informações da camada física para efetuar o escalonamento. As características dos algoritmos de escalonamento mSIR e mmSIR foram descritas no Capítulo 3.

Neste experimento, analisou-se o atraso médio em função da carga de tráfego inserida na rede de acesso para a classe de serviço rtPS. O atraso médio estimado foi definido com o valor de 100 ms para a classe rtPS. O enlace foi carregado em aproximadamente 60% da intensidade máxima de tráfego. A Figura 6.13 apresenta os resultados de simulação obtidos [76].

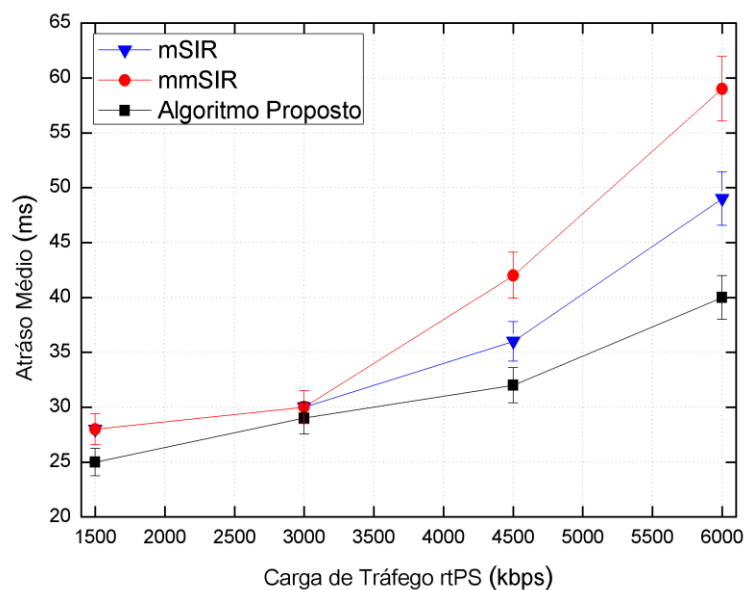


Figura 6.13: Atraso médio vs. carga de tráfego rtPS.

Como pode ser observado na Figura 6.13, o algoritmo proposto apresentou um melhor desempenho se comparado com os algoritmos mSIR e mmSIR. Baseando-se no valor da SNR e nas mensagens *BR*, o algoritmo proposto gera um *deadline* específico para cada conexão. As conexões são escalonadas de acordo com estes *deadlines*. O tráfego rtPS é composto por pacotes de tamanho variável, gerando assim, uma carga de tráfego variável, e neste caso o *deadline* calculado também se torna variável. Logo, a prioridade atribuída a cada conexão também varia de forma totalmente dinâmica, dando chances para todas as SSs transmitirem seus dados.

Os algoritmos mSIR e mmSIR têm como premissa atribuir prioridades para as SSs que possuem canais de comunicação com melhores condições de transmissão. Neste caso, as SSs que possuem canais de comunicação com desempenhos inferiores nunca serão atendidas, influenciando diretamente na garantia da QoS para as aplicações. Como pode ser observado na Figura 6.13, o atraso médio do algoritmo proposto foi inferior ao atraso médio dos algoritmos mSIR e mmSIR. Isso se deve ao fato de que o algoritmo proposto ordena a lista das SSs que serão atendidas pelo *deadline*, dando oportunidade para todas as SSs transmitirem seus dados. Como o tráfego da classe rtPS é variável, a prioridade de transmissão das SSs muda dinamicamente. Isso não ocorre nos algoritmos mSIR e mmSIR, onde as prioridades são atribuídas às SSs com melhores condição de canal de transmissão, influenciando diretamente no atraso médio.

6.3.7. Análise do comportamento do algoritmo proposto para a classe nrtPS

Este experimento de simulação tem por objetivo verificar o impacto do aumento da carga de tráfego do serviço rtPS sobre o desempenho do serviço nrtPS. Assim, é possível analisar se o algoritmo proposto é capaz de garantir a largura de banda mínima para o serviço nrtPS. A

rede de acesso simulada possui uma BS, 15 SSs com uma conexão nrtPS por SS, e 25 SSs com uma conexão rtPS por SS. O número de conexões rtPS ativas varia de 5 a 25. A fonte de tráfego de cada conexão nrtPS gera um tráfego FTP com a taxa de 300 Kbps, e o requisito mínimo de largura de banda para cada conexão é de 30 Kbps. Este experimento foi executado com o algoritmo proposto, e com os algoritmos: WRR e RR. A Figura 6.14 mostra a vazão média das conexões nrtPS em função da carga de tráfego nrtPS [75].

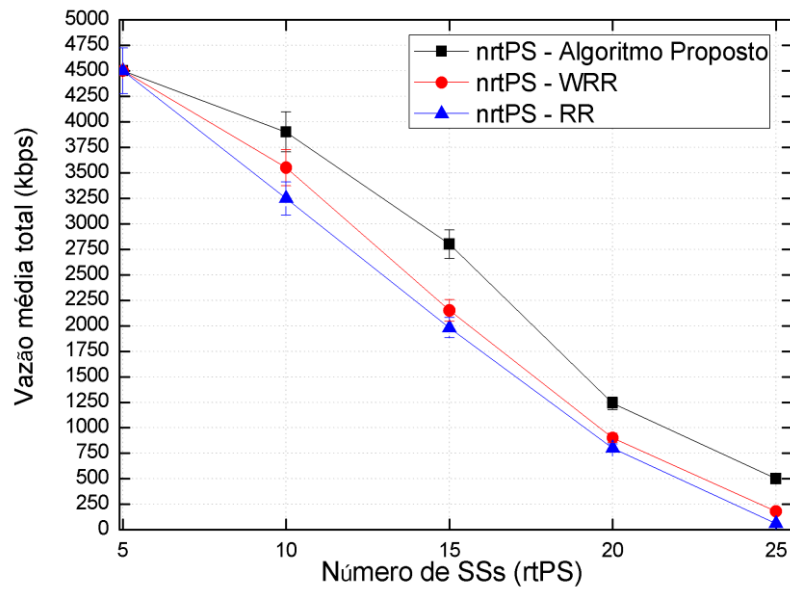


Figura 6.14: Vazão média total vs. carga de tráfego rtPS.

Como pode ser visto na Figura 6.14, a vazão das conexões nrtPS diminuiu de acordo com o aumento da carga de tráfego rtPS. Este comportamento já era esperado devido ao aumento da carga de tráfego de uma classe que possui maior prioridade. Entretanto, o algoritmo proposto mostrou melhor desempenho quando comparado com os algoritmos RR e WRR, uma vez que ele interage com o mecanismo de *polling unicast*, e ajusta o intervalo de *polling* de forma dinâmica. Desta forma, de acordo com os recursos disponíveis, as SSs recebem um número maior de *grants* para requisitar maior quantidade de largura de banda. Por outro lado, os algoritmos RR e WRR não interagem com o mecanismo de *polling unicast*,

e utilizam o intervalo fixo de *polling*, ofertando menor quantidade de largura de banda se comparado com o algoritmo proposto.

6.3.8. Análise do comportamento do algoritmo proposto para as classes nrtPS e BE

Este experimento de simulação tem como objetivo analisar como o algoritmo proposto distribui os recursos para as aplicações não tempo real, associadas às classes de serviço nrtPS e BE. O cenário de simulação é composto de uma BS e 20 SSs com uma conexão BE por SS. 30 SSs com uma conexão nrtPS por SS, onde o número de SSs ativas varia de 5 a 30. Foram utilizadas neste experimento 5 SSs com uma conexão UGS por SS e 5 SSs com uma conexão rtPS por SS como tráfego de *background*. A Figura 6.15 ilustra a vazão das classes de serviço nrtPS e BE [75].

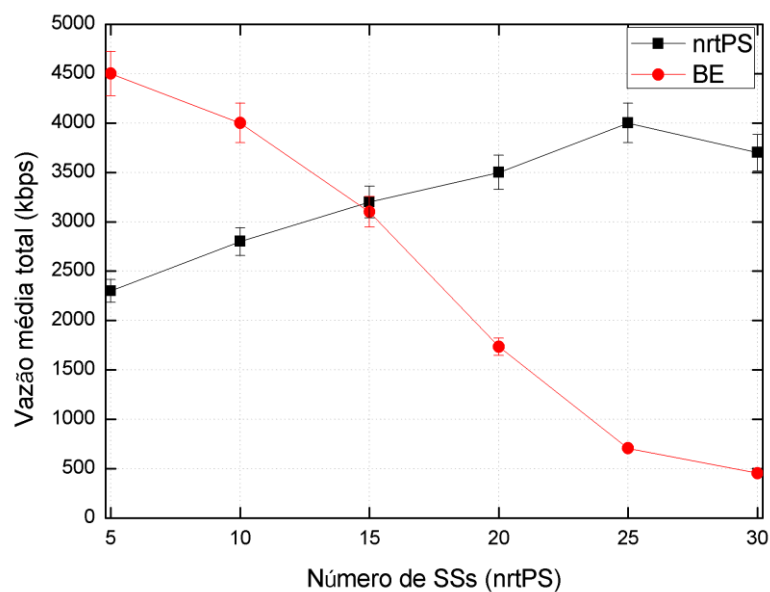


Figura 6.15: Vazão média das classes de serviço nrtPS e BE vs. a carga de tráfego nrtPS.

Como pode ser observado na Figura 6.15, inicialmente, o serviço BE possui uma maior vazão média em relação ao serviço nrtPS. Isso ocorreu porque o escalonador alocou os recursos (*slots*) não utilizados pelas SSs que possuem maior prioridade (por exemplo, UGS, rtPS e

nrtPS) para a classe de serviço BE. Quando o número de SSs com conexões nrtPS aumentou, o escalonador distribuiu os recursos existentes entre as conexões nrtPS, causando redução na vazão da classe de serviço BE, uma vez que esta classe é destinada as aplicações elásticas. A Figura 6.16 ilustra o atraso médio das classes de serviço rtPS e nrtPS em função da carga de tráfego nrtPS [75].

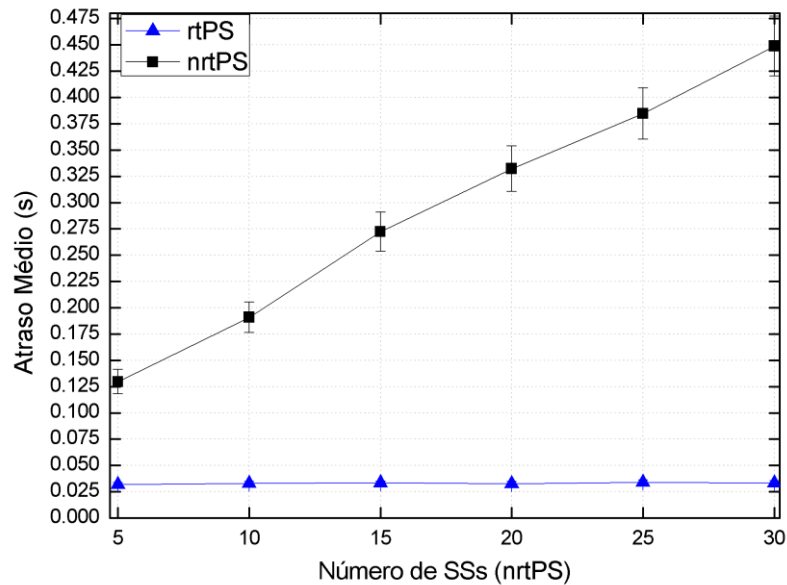


Figura 6.16: Atraso médio das conexões nrtPS e rtPS vs. a carga de tráfego nrtPS.

Como pode ser visto na Figura 6.16, o atraso médio da classe de serviço rtPS não foi afetado pelo aumento da carga de tráfego do serviço nrtPS. Esse comportamento está relacionado com o mecanismo de gerenciamento de *polling unicast*, que pode ajudar o escalonador a fazer o balanceamento entre as restrições temporais existentes na classe rtPS e os requisitos de vazão das aplicações da classe nrtPS.

6.3.9. Análise do comportamento da classe de serviço rtPS em função da utilização do mecanismo de CAC.

Este experimento de simulação tem por objetivo analisar o desempenho da classe de serviço rtPS em função do mecanismo de CAC utilizado. A rede simulada inclui uma BS e 20

SSs com uma conexão rtPS por SS. A taxa de transmissão varia entre 200 a 800 kbps por conexão, e o número de SSs na rede varia de 2 a 20. As características do tráfego são as mesmas utilizadas nos experimentos anteriores. O atraso máximo definido para a classe rtPS foi de 200 ms. A Figura 6.17 ilustra os resultados obtidos.

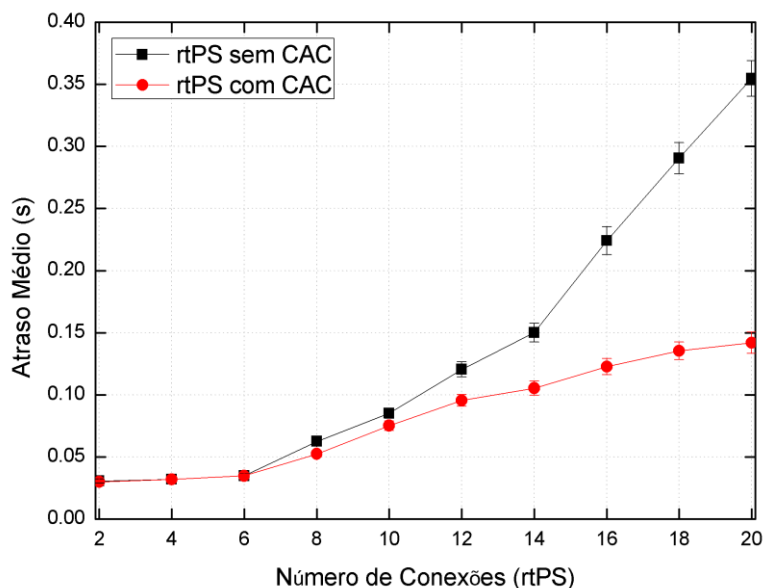


Figura 6.17: Atraso médio das conexões rtPS vs. o número de conexões rtPS.

Como pode ser observado na Figura 6.17, quando o algoritmo de CAC foi utilizado, o atraso médio das conexões rtPS ficou abaixo do atraso máximo definido. Isso aconteceu porque o algoritmo de CAC somente aceita conexões rtPS de acordo com os recursos disponíveis.

6.3.10. Análise do comportamento das classes de serviço UGS, ertPS e rtPS em função da utilização do mecanismo de CAC.

Este experimento de simulação tem por objetivo analisar o comportamento da classe de serviço rtPS em função do aumento do número de conexões UGS e ertPS. O experimento foi executado com e sem o mecanismo de CAC. A rede simulada inclui uma BS, 10 SSs com uma conexão UGS por SS, 10 SSs com uma conexão ertPS por SS, e 20 SSs com uma

conexão rtPS por SS. Nesse caso, definiu-se um atraso máximo para as conexões rtPS de 100 ms. A Figura 6.18 mostra o resultado do atraso médio das conexões.

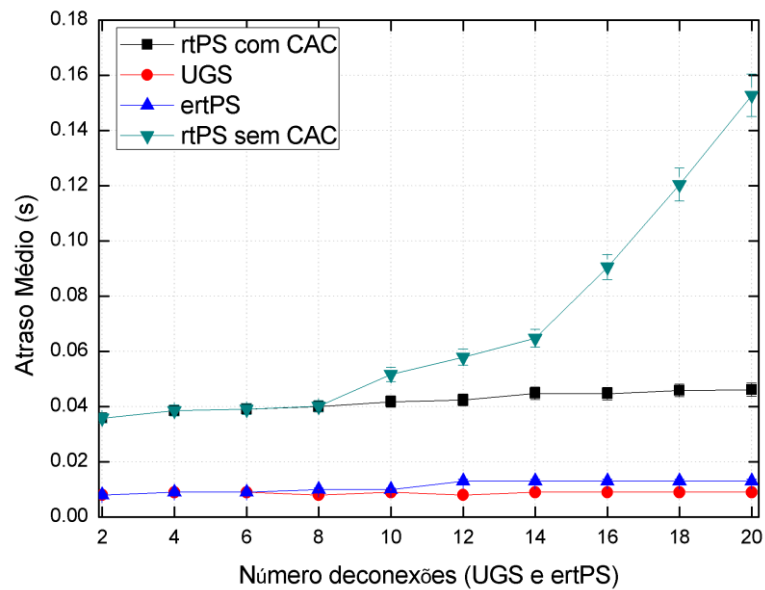


Figura 6.18: Atraso médio das conexões rtPS vs. o aumento do número de conexões UGS e ertPS.

Como pode ser observado na Figura 6.18, o atraso médio das conexões UGS e ertPS permaneceu praticamente constante. O atraso médio das conexões rtPS permaneceu abaixo do limiar definido quando utilizou-se o mecanismo de CAC, pois a alocação dos recursos existentes é feita de forma dinâmica. Assim sendo, o mecanismo de CAC utiliza, quando necessário, os recursos pré-definidos para as aplicações das classes de serviço sem restrição temporal. Quando se utiliza alocação de forma estática, os recursos reservados são atribuídos primeiramente para as aplicações das classes de serviço com maior prioridade, nesse caso as classes UGS e ertPS. Os recursos restantes são atribuídos para a classe de serviço rtPS. A Figura 6.19 ilustra o resultado da vazão média total das conexões UGS, ertPS e rtPS.

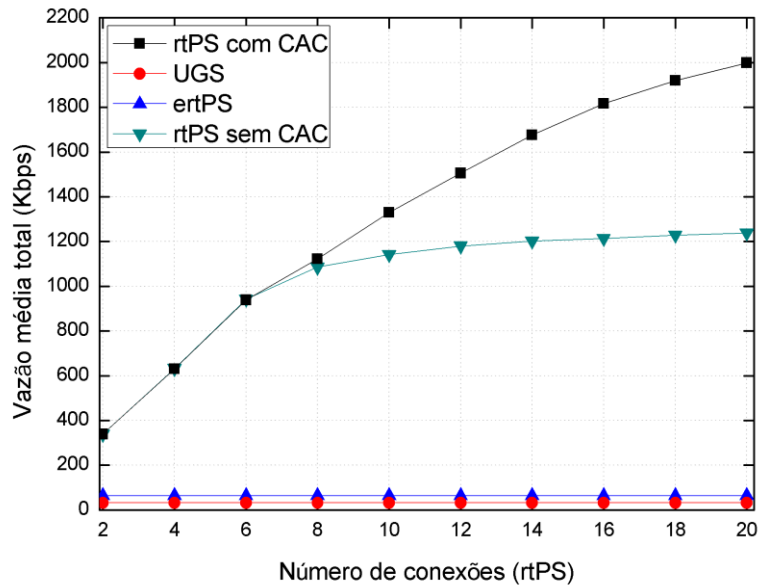


Figura 6.19: Vazão média total das conexões UGS, ertPS e rtPS vs. número de conexões rtPS.

Como pode ser observado na Figura 6.19, a vazão média dos serviços UGS e ertPS permaneceu praticamente constante. O aumento do número de conexões da classe de serviço rtPS não interferiu no desempenho das classes de serviço UGS e ertPS. É possível observar também que, quando utilizou-se o mecanismo de CAC, a vazão da classe de serviço rtPS aumentou, pois utilizou-se recursos pré-reservados para as aplicações sem restrição temporal. Tal controle não ocorre quando não se utilizou o mecanismo de CAC, Com isso, é possível observar que a vazão é menor

6.3.11. Análise do comportamento da classe de serviço nrtPS em função da utilização do mecanismo de CAC.

Este experimento tem por objetivo analisar o comportamento da classe de serviço nrtPS com e sem a utilização do mecanismo de CAC. A rede simulada inclui uma BS, 15 SSs com uma conexão rtPS por SS; 10 SSs com uma conexão nrtPS por SSs, onde a fonte de tráfego de cada conexão nrtPS gerou 200 kbps. A taxa mínima reservada para o serviço nrtPS foi de 55

kbps. Neste experimento, 10 SSs com uma conexão UGS foram utilizadas como tráfego de *background*. A Figura 6.20 ilustra a vazão média das conexões nrtPS.

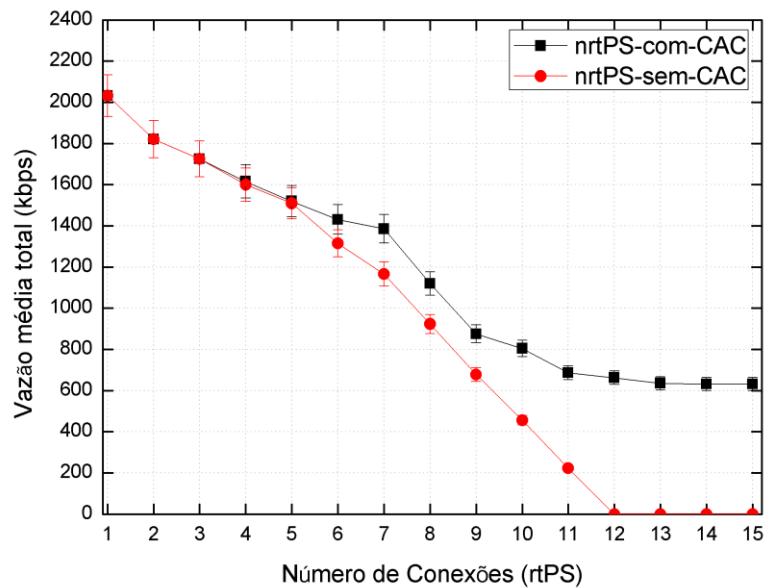


Figura 6.20: Vazão média total das conexões nrtPS vs. número de conexões rtPS.

Como pode ser observado na Figura 6.20, a taxa mínima reservada para a classe de serviço nrtPS foi garantida quando o mecanismo de CAC foi utilizado. O mecanismo de CAC reserva parte dos *slots* para as aplicações de tempo real, e também reserva a quantidade de *slots* necessária para garantir os recursos mínimos para as aplicações do serviço nrtPS. Além disso, existe a interação do mecanismo de *polling*, onde o intervalo de *polling unicast* é controlado de forma dinâmica. A Figura 6.21 ilustra o atraso médio da classe de serviço nrtPS em função do número de conexões rtPS.

Como pode ser observado na Figura 6.21, o atraso médio das conexões nrtPS é bem menor quando o mecanismo de CAC é empregado, se comparado com o atraso médio das conexões nrtPS quando não se utilizou o mecanismo de CAC. O mecanismo de CAC reserva o mínimo de recursos para as conexões nrtPS, diferentemente do que acontece quando não se utiliza o mecanismo de CAC.

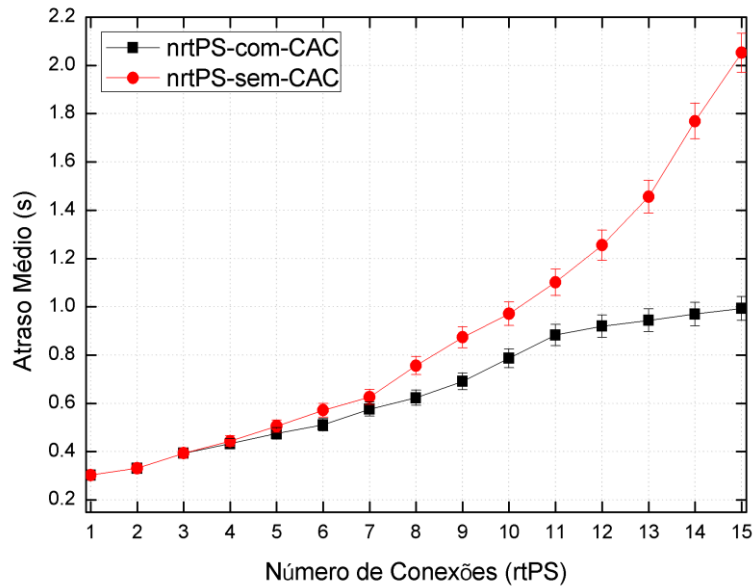


Figura 6.21: Atraso médio das conexões nrtPS vs. número de conexões rtPS.

6.4. Trabalhos relacionados

É proposto em [77] um algoritmo de escalonamento para o tráfego *uplink*. A arquitetura do escalonador é composta de três filas de prioridade, de forma que uma fila de baixa prioridade só é atendida quando as filas de maior prioridade estiverem vazias. A fila de menor prioridade armazena as requisições de banda do serviço BE. Na fila intermediária estão as requisições dos serviços rtPS e nrtPS enviadas pelas Ss e a fila de maior prioridade armazena os *grants* para o envio de dados do serviço UGS. As requisições das filas intermediárias podem migrar para a fila de maior prioridade no momento em que precisarem ser atendidas para garantir QoS. O escalonamento proposto em [77] foi estendido pelos autores em [78], onde utilizou-se uma abordagem *cross-layer*. Desta forma, as decisões de alocação de largura de banda passam a ser feitas levando em consideração as informações sobre a qualidade do canal de comunicação. O algoritmo proposto nesta tese armazena as mensagens de requisição de largura de banda das classes de serviço rtPS, nrtPS e BE em suas respectivas filas. As informações contidas nas mensagens de requisição de largura de banda, como também as informações sobre a qualidade do canal de comunicação são utilizadas para calcular o

deadline para as conexões da classe de serviço rtPS, o que determina a ordem de escalonamento dessas conexões, e também para calcular e verificar a quantidade de largura de banda alocada para as conexões de tempo real e não tempo real. O serviço UGS possui alocação de banda fixa, não sendo necessário o envio de requisição de largura de banda por parte das SSS, uma vez que a alocação de largura de banda para o serviço UGS é negociada na fase de “*setup*” da conexão UGS. Assim sendo, primeiramente é feito o escalonamento da classe UGS e, depois, o escalonamento das outras classes de serviço, conforme suas prioridades.

O algoritmo de escalonamento proposto em [79] tem como objetivo alocar recursos de largura de banda para ambas as direções, *downlink* e *uplink*. A solução proposta baseia-se em aplicar o mecanismo *token bucket* para determinar a transmissão de *grants* para o serviço CBR, sendo este serviço mapeado para a classe UGS. Neste caso, o mecanismo *token bucket* não é utilizado para policiar o tráfego, mas sim para medir e regular dinamicamente as transmissões de cada fluxo de serviço. Porém, esta solução contempla apenas as classes de serviço UGS e BE, deixando para trabalhos posteriores a implementação da solução para os serviços rtPS e nrtPS. A solução apresentada nesta tese controla dinamicamente as transmissões dos fluxos de serviço rtPS e nrtPS.

Os autores em [80] desenvolveram um algoritmo de escalonamento utilizando uma abordagem *cross-layer*. O algoritmo desenvolvido é executado em dois estágios. No primeiro estágio, valores de largura de banda são atribuídos dinamicamente para as classes de serviço. No segundo estágio, diferentes conexões pertencentes à mesma classe de serviço são escalonadas. Neste estágio, aplica-se uma função para determinar as prioridades para as diferentes conexões de acordo com os parâmetros de QoS, onde definiu-se um *Quantum* para cada classe de serviço. Entretanto, os autores não deixam claro como o *Quantum* é calculado. O algoritmo proposto neste trabalho faz o escalonamento das conexões de acordo com um

novo *deadline* que é definido para as aplicações de tempo real e também de acordo com os requisitos de largura de banda das aplicações de tempo real e não tempo real.

Os autores em [81] fizeram um levantamento sobre a existência de várias arquiteturas de escalonamento desenvolvidas para redes WiMAX, e questões como mecanismos de sinalização e técnicas de controle de admissão foram discutidas. Comparando-se as soluções propostas, várias questões foram levantadas, como por exemplo, o impacto no sistema decorrente da utilização do mecanismo de requisição e alocação de largura de banda em termos de vazão e atraso. Tais questões foram levadas em consideração no desenvolvimento do algoritmo proposto.

Os autores em [82] desenvolveram um algoritmo de escalonamento baseado na política de escalonamento EDF. O tamanho, em bytes, de cada pacote pertencente a uma determinada conexão é convertido em unidades de *slots* físicos, e o *deadline* atribuído para cada pacote é computado em unidade de *frames*. Para cada conexão, é criada uma fila FIFO, onde os pacotes são armazenados. As conexões existentes no sistema são representadas e gerenciadas através de uma matriz. Os autores em [82] não deixam claro qual MCS é utilizada para calcular o tamanho dos pacotes em número de *slots*. Além disso, a utilização de uma matriz para gerenciar as filas existentes exige uma maior complexidade na implementação do algoritmo. O algoritmo de escalonamento proposto nesta tese também utiliza o algoritmo de escalonamento EDF. Entretanto, o *deadline* é atribuído para cada mensagem BR, e não a cada pacote, pois a BS não possui informações do tamanho dos pacotes que estão armazenadas nas filas das SSs.

Em [83] foi proposto um algoritmo de escalonamento híbrido para o tráfego *uplink*. Tal algoritmo combina dois algoritmos de escalonamento: o EDF e o WFQ. A largura de banda é alocada para as classes de serviço baseando-se no número de SSs existente na rede e no parâmetro de largura de banda mínima, exigida por algumas classes de serviço. Segundo os

autores, o algoritmo de escalonamento *uplink* é executado de acordo com a chegada de cada pacote nas filas das SSs. Porém, não fica claro como isso é feito, uma vez que a BS não possui a informação da chegada de cada pacote nas SSs. O *deadline* definido no algoritmo proposto nesta tese utiliza as informações das mensagens de requisição de largura de banda enviada pelas SSs (*BR*). Essas mensagens refletem o tamanho das filas das SSs, no momento em que as mesmas foram geradas. Além disso, o escalonamento da classe de serviço rtPS é baseada no algoritmo EDF.

Foi proposto em [32] um algoritmo de escalonamento para o tráfego *uplink* que leva em consideração informações da camada física. Os recursos são alocados para as SSs de acordo com a qualidade do canal de comunicação. Quanto melhor for o canal de comunicação, maior será a prioridade de transmissão. Embora tal abordagem permita uma alta utilização do enlace, o escalonamento não é justo, pois as SSs com qualidade de transmissão inferior podem não ser atendidas pelo escalonador. Já em [84] foi proposta uma modificação do algoritmo descrito desenvolvido por [32]. Tal proposta consiste em servir as SSs que não possuem requisição de largura de banda no mesmo *frame*, possibilitando atender as requisições com qualidade de canal inferior. Todavia, tal solução foi avaliada apenas com a classe de serviço rtPS, e não foram consideradas as classes de serviço UGS, ertPS e nrtPS. Os resultados apresentados sobre a eficiência dos algoritmos de escalonamento desenvolvidos em [32] e [84] contemplam apenas a vazão. Entretanto, o parâmetro de atraso máximo é muito importante e deve ser levado em consideração. O algoritmo proposto nesta tese foi desenvolvido levando em consideração as classes de serviço UGS, ertPS, rtPS, nrtPS e BE. Além disso, para a classe de serviço rtPS, avaliou-se a vazão, o atraso médio e o *jitter* médio. Alguns desses parâmetros, como por exemplo, o atraso médio, foram avaliados em dois ambientes: um ambiente com apenas uma MCS, e um ambiente com várias MCSs.

6.5. Considerações finais

Este capítulo apresentou uma avaliação do algoritmo de escalonamento proposto e que foi descrito no Capítulo 5. Experimentos foram realizados considerando vários cenários de simulação. Considerou-se nestes cenários, a transmissão dos dados utilizando vários perfis de rajadas, e também cenários com apenas um perfil de rajada. Considerou-se também cenários específicos em que utilizou-se o mecanismo de CAC e também cenários nos quais não utilizou-se o mecanismo de CAC.

Os resultados apresentados mostram que o algoritmo proposto é capaz de prover QoS considerando os vários cenários de simulação, fornecendo uma solução compatível com o padrão IEEE 802.16, inclusive no que diz respeito a utilização de várias MCSs na rede de acesso.

O algoritmo de escalonamento proposto foi comparado com alguns algoritmos de escalonamento tradicionais (RR, WRR, EDF), e também foi comparado com alguns algoritmos de escalonamento que utilizam abordagem *cross-layer* (*mSIR* e *mmSIR*). Em todos os casos, os resultados mostraram uma melhor eficiência do algoritmo de escalonamento proposto.

Os resultados de vazão média total e atraso médio mostram que o algoritmo de escalonamento proposto, em conjunto com o mecanismo de CAC, é capaz de garantir para as classes de serviço os requisitos mínimos de QoS definidos pelo padrão. Além disso, com a utilização do mecanismo de *polling* adaptativo, é possível minimizar o atraso médio, além de fazer um melhor uso do mecanismo de *polling unicast*.

Capítulo 7

CONCLUSÕES GERAIS E DESENVOLVIMENTOS FUTUROS

7.1. Conclusões

O padrão IEEE 802.16 apresenta uma solução para as redes de acesso de banda larga sem fio. Uma das grandes vantagens do padrão IEEE 802.16 é a especificação, na camada MAC, de mecanismos para a provisão de QoS. Entretanto, algumas questões são deixadas em aberto pelo padrão, para que os fabricantes de dispositivos possam diferenciar seus produtos, o que tem despertado grande interesse na comunidade científica.

Assim sendo, esta tese concentrou-se na investigação e proposta de um algoritmo de escalonamento adaptativo para o tráfego *uplink* em redes IEEE 802.16. Os desafios de projeto de algoritmos de escalonamento em redes IEEE 802.16, que foram introduzidos no Capítulo 5, motivaram a solução proposta nesta tese. O algoritmo de escalonamento proposto foi desenvolvido utilizando uma abordagem *cross-layer*. Deste modo, o escalonador aloca os recursos para as aplicações levando em consideração a MCS utilizada, e faz o escalonamento levando em consideração tal informação. Os resultados mostraram que o algoritmo de escalonamento fornece garantias de atraso máximo limitado para as aplicações de tempo real

e largura de banda mínima para as aplicações de tempo real e não tempo real, sendo compatível com o padrão IEEE 802.16.

O algoritmo de escalonamento proposto foi desenvolvido considerando as cinco classes de serviço definidas pelo padrão. O escalonamento das conexões da classe UGS e ertPS é feito de acordo com critérios definidos pelo padrão IEEE 802.16. O escalonamento das conexões rtPS é feito baseado em *deadlines*, onde os mesmos são calculados a partir de informações obtidas da camada física.

A avaliação do algoritmo de escalonamento proposto foi realizada por meio de modelagem e simulação. Diversos experimentos foram realizados em cenários com transmissão de dados utilizando apenas uma MCS e também utilizando várias MCSs. Além disso, o algoritmo proposto foi comparado com algoritmos de escalonamento tradicionais, que são frequentemente descritos nos estudos comparativos e trabalhos relacionados.

Os resultados obtidos mostraram que o algoritmo de escalonamento proposto é capaz de garantir um atraso máximo limitado para as aplicações de tempo real nos cenários testados. Além disso, considerando o resultado dos algoritmos de escalonamento tradicionais, foi possível observar que a utilização de várias MCSs na rede de acesso interfere no atraso máximo percebido pelas aplicações. Entretanto, o algoritmo de escalonamento proposto foi capaz de minimizar tal atraso, sendo apropriado para tal cenário.

O algoritmo de escalonamento proposto interage com o mecanismo de *polling* da BS. Com isso, o intervalo de alocação consecutiva de *slots* para que as SSs possam enviar suas requisições de largura de banda é feito de forma dinâmica para as classes de serviço rtPS e nrtPS. Os resultados obtidos mostraram que o controle dinâmico do *polling* contribui com o algoritmo de escalonamento para minimizar o atraso de escalonamento decorrente da utilização do mecanismo de *polling*. O mecanismo de gerenciamento de *polling* faz um

balanceamento do *polling* para as classes de serviço rtPS e nrtPS, contribuindo para com a garantia de largura de banda mínima para as classes de serviço rtPS e nrtPS.

Tendo em vista que a provisão da QoS em redes baseadas no padrão IEEE 802.16 depende da ação conjunta do mecanismo de escalonamento e do mecanismo de CAC, desenvolveu-se nesta tese um novo mecanismo de CAC para atuar em conjunto com o algoritmo de escalonamento proposto. Tal mecanismo também foi desenvolvido utilizando uma abordagem *cross-layer*. A avaliação do desempenho do mecanismo de CAC foi feito em cenários onde utilizou-se várias MCSs para a transmissão dos dados. Os resultados mostraram que o mecanismo de CAC desenvolvido nesta tese é eficiente, pois foi possível garantir os requisitos de QoS em todos os cenários nos quais utilizou-se o mecanismo de CAC.

7.2. Desenvolvimentos futuros

Um dos objetivos do padrão IEEE 802.16 é prover acesso em banda larga sem fio a qualquer hora e em qualquer lugar. Tal objetivo envolve a capacidade de prover mobilidade para seus usuários, como especificado no padrão IEEE 802.16e. O mecanismo de CAC deve atender as novas solicitações, bem como atender as solicitações provenientes de células diferentes. Assim sendo, o mecanismo de CAC pode ser estendido para atender as conexões provenientes de outras células, dando suporte para o *handoff*.

O gerenciamento dinâmico do *polling* é pouco explorado na literatura, se comparado com o mecanismo de escalonamento e CAC. A maioria dos algoritmos de escalonamentos desenvolvidos utiliza intervalo fixo de *polling*. O mecanismo dinâmico de *polling* desenvolvido nesta tese pode ser estendido para outras soluções de algoritmos de escalonamento existentes na literatura.

O IEEE aprovou o padrão 802.16m, que atualiza o padrão WiMAX. Esse padrão define velocidades superiores às velocidades definidas nos padrões já existentes. Assim sendo, um

estudo interessante seria investigar a viabilidade de se utilizar o escalonador proposto nesse novo padrão, que visa o desenvolvimento de redes 4G.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] CHANG, C. A Mobile-IP Based Mobility System for Wireless Metropolitan Area Networks. International Conference Workshops on Parallel Processing (ICPP), pp. 429 – 435, June 2005.
- [2] IEEE 802.16 Working Group. IEEE Standard for Local and Metropolitan Area Networks – Part. 16: Air Interface for Fixed Broadband Wireless Access Systems. October, 2004.
- [3] AHSON, S.; ILYAS, M. WiMAX Technologies, Performance Analysis, and QoS, CRC Press, September, 2007.
- [4] LI, B. et al. A Survey on Mobile WiMAX. IEEE Communications Magazine, pp. 70-75, December, 2007.
- [5] MA, M; FU, C.P. Hierarchical Scheduling Framework for QoS Service in WiMAX Point-to-Multipoint Networks. IET Communications, vol. 4, no. 9, pp. 1073 – 1082, June, 2010.
- [6] CHUNK, D.; CHANG, J. M. Bandwidth Recycling in IEEE 802.16 Networks. IEEE Transactions on Mobile Computing, vol. 9, no. 10, pp. 1451 – 1464, October, 2010.
- [7] NS-2. The Network Simulator. Disponível em: <<http://www.isi.edu/nsnam/ns/>>. Acessado em: Junho, 2012.
- [8] GHOSH, A.; WOLTER, D. R.; ANDREWS, J. G.; CHEN, R. Broadband Wireless Access with WiMax/802.16: Current Performance Benchmarks and Future Potential. IEEE Communications Magazine, vol. 43, no. 2, pp. 129 – 136, February, 2005.

- [9] SENGUPTA S. et all. Exploiting MAC Flexibility in WiMAX for Media Streaming. Proceedings of 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks. Giardini Naxos, Italy, pp. 338–343, June, 2005.
- [10] NUAYMI, L. WiMAX: Technology for Broadband Wireless Access. John Wiley & Sons, Ltd, 2007.
- [11] ANDREWS, J. G. et all. Fundamentals of WiMAX Understanding Broadband Wireless Networking, Prentice Hall, February, 2007.
- [12] IEEE 802.16e-2005, IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Mobile Broadband Wireless Access Systems. IEEE Std., Rev. IEEE Std 802.16-2005, 2005.
- [13] CHUCK, D.; CHEN, K.; CHANG, J. M. Comprehensive Analysis of Bandwidth Request Mechanisms in IEEE 802.16 Networks. IEEE Transactions on Vehicular Technology, vol. 59, no. 4, pp. 2046 – 2056, May, 2010.
- [14] LAKKAKORPI, J.; SAYENKO, A. Uplink VoIP Delays in IEEE 802.16e Using Different ertPS Resumption Mechanisms. Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp. 157 – 162, December, 2009.
- [15] DHRONA, P.; ABU ALI, N.; HASSANEIN, H. S. A performance study of uplink scheduling algorithms in point-to-multipoint WiMAX networks. Computer Communications, vol. 32, no. 3, pp. 511– 521, February, 2009.
- [16] SO-IN, C.; JAIN, R.; TAMIMI, A. Scheduling in IEEE 802.16e Mobile WiMAX Networks: Key Issues and a Survey. IEEE Journal on Selected Areas in Communications (JSAC), vol. 27, no. 2, pp. 156 – 171 February, 2009.

- [17] MSADAA, I. C.; CAMARA, D.; FILALI, F. Scheduling and CAC in IEEE 802.16 Fixed BWNs: A Comprehensive Survey and Taxonomy. IEEE Communications Surveys & Tutorials, vol. 12, no. 4, pp. 459 – 487, May, 2010.
- [18] CHENG, S. T.; HSIEH, M. T.; CHEN, B. F. Fairness-based Scheduling Algorithm for Time Division Duplex Mode IEEE 802.16 Broadband Wireless Access Systems. IET Communications, vol. 9, p. 1065 – 1072, April, 2010.
- [19] SAYENKO, A. et al. Comparison and analysis of the revenue-based adaptive queuing models. Computer Networks, vol. 50, no. 8, pp. 1040 – 1058, June, 2006.
- [20] SHREEDHAR, M.; VARGHESE, G. Efficient fair queuing using deficit round robin. IEEE/ACM Transactions on Networking, vol. 4, no. 3, pp. 375 – 385, June, 1996.
- [21] ANDREWS, M. Probabilistic end-to-end delay bounds for earliest deadline first scheduling. In proceedings IEEE Computer Communication Conference, vol.2, pp. 603 – 612, Israel, February, 2000.
- [22] CICCONE, C. et al. Quality of Service Support in IEEE 802.16 Networks, IEEE Network, vol. 20, no. 2, pp. 50 – 55, April, 2006.
- [23] SAYENKO, A.; ALANEN, O.; HAMALAINEN, T. Scheduling Solution for the IEEE 802.16 Base Station. International Journal of Computer Networks, vol. 52, no. 1, pp. 96 – 115, January, 2008.
- [24] CICCONE, C.; ERTA, A.; LENZINI, L.; MINGOZZI, E. Performance Evaluation of the IEEE 802.16 MAC for QoS Support. IEEE Transactions on Mobile Computing, vol.6, no. 1, pp.26 – 38, January, 2007.
- [25] RUANGCHAIJATUPON, N.; WANG, L.; JI, Y. A Study on the Performance of Scheduling Schemes for Broadband Wireless Access Networks. In Proceedings of International Symposium on Communications and Information Technology, pp. 1008 – 1012, October, 2006.

- [26] WONGTHAVARAWAT, K.; GANZ, A. Packet Scheduling for QoS Support in IEEE 802.16 Broadband Wireless Access Systems. International Journal of Communications Systems, vol. 16, no. 1, pp. 81 – 96, February, 2003.
- [27] VINAY, K.; SREENIVASULU, N.; JAYARAM, D.; DAS, D. Performance Evaluation of End-to-end Delay by Hybrid Scheduling Algorithm for QoS in IEEE 802.16 Network. In Proceedings of the International Conference on Wireless and Optical Communication Networks, pp. 1 – 5, April, 2006.
- [28] CHEN, J.; JIAO, W.; WANG, H. A Service Flow Management Strategy for IEEE802.16 Broadband Wireless Access Systems in TDD Mode. In Proceedings of the IEEEICC, pp. 3422 – 3426, August, 2005.
- [29] ALSAHAG, A. M.; ALI, B. M.; NOORDIN, N. K.; MOHAMAD, H. Fair Bandwidth Assignment in Hierarchical Scheduling for Mobile WiMAX System. 17th Asia-Pacific Conference on Communications (APCC), pp. 743 – 747, February, 2011.
- [30] LIU, N.; LI, X.; PEI, P.; YANG, B. Delay Character of a Novel Architecture for IEEE 802.16 Systems. 6th International Conference on Parallel and Distributed Computing, Applications and Technologies, pp 293 – 296, December, 2005.
- [31] KURAN, M. S.; GUR, G.; TUGCU, T.; ALAGOZ, F. Applications of the Cross-Layer Paradigm for Improving the Performance of WiMax. IEEE Wireless Communications, vol. 17, no. 3, pp. 86 – 95, June 2010.
- [32] BALL, C. F.; TREML, F.; GAUBE, X.; KLEIN, A. Performance Analysis of Temporary Removal Scheduling applied to mobile WiMAX Scenarios in Tight Frequency Reuse. The 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications, PIMRC'2005, Berlin. vol. 2. pp. 888 – 898, September, 2005.

- [33] KHAN, M. A. S.; SATTAR, A. R.; MUSTAFA, T.; AHMAD, S. Performance Evaluation and Enhancement of Uplink Scheduling Algorithms in Point to Multipoint WiMAX Networks. European Journal of Scientific Research, vol. 42, no. 3, pp. 491 – 506, March, 2010.
- [34] LIU, Q.; WANG, X.; GIANNAKIS, G. B. A Cross-Layer Scheduling Algorithm with QoS Support in Wireless Networks. IEEE Transactions on Vehicular Technology, vol. 55, no. 3, pp. 839 – 847, June, 2006.
- [35] EL-FISHAWY, N. A.; ZAHRA, M.; EBRAHIM, M.; EL-GAMALA, M. M. Modified Cross-Layer Scheduling for Mobile WiMAX Networks. 28th National Radio Science Conference (NRSC), pp. 1 – 10, April, 2011.
- [36] RATH, K. H; ABHIJEET B; SHARMA, V. An Opportunistic Uplink Scheduling Scheme to Achieve Bandwidth Fairness and Delay for Multiclass Traffic in Wi-Max (IEEE802.16) Broadband Wireless Networks. Global Telecommunications Conference, IEEE Globecom, pp. 1 – 5, April, 2006.
- [37] SHAHWAN, I.; MUHAMMED, A.; OBAIDAT, M.; DORSINVILLE, R. WiMax:Cross Layer Bandwidth Allocation Strict Priority Based Adaptive modulation and Coding. International Conference on Communications and Information Technology (ICCIT), Aqaba, pp. 73 – 76, May, 2011.
- [38] GE, Y.; KUO, G. S. An efficient admission control scheme for adaptive multimedia services in IEEE 802.16e networks. In IEEE 64th Vehicular Technology Conference, pp. 1 – 5, September, 2006.
- [39] WANG, H.; LI, W.; AGRAWAL, D. P. Dynamic Admission Control and QoS for 802.16 Wireless MAN. In Wireless Telecommunication. Symposium, pp. 60 – 66, April, 2005.

- [40] JIANG, C. H.; TSAI, T. C. Token bucket based CAC and packet scheduling for IEEE 802.16 broadband wireless access networks. In 3rd IEEE Consumer Communication Networks Conference, vol. 1, pp. 183 – 187, January, 2006.
- [41] CHANDRA, S.; SAHOO, A. An efficient call admission control for IEEE 802.16 networks. In 15th IEEE LAN/MAN Workshop (LANMAN 2007), pp. 188 – 193, June, 2007.
- [42] NIYATO, D.; HOSSAIN, E. Radio resource management games in wireless networks: An approach to bandwidth allocation and admission control for polling service in IEEE 802.16. IEEE Wireless Communication, vol. 14, no. 1, pp. 27 – 35, February, 2007.
- [43] WANG, L.; LIU, F.; JI, Y.; RUANGCHAIJATUPON, N. Admission control for non-preprovisioned service flow in wireless metropolitan area networks. In Fourth European Conference on Universal Multiservice Networks (ECUMN '07), pp. 243 – 249, February, 2007.
- [44] WU, H.; KE, K.; YANG, C. The Design of QoS Provisioning Mechanisms for Mobile WiMAX Networks. International Conference on New Trends in Information and Service Science (NISS), pp. 775 – 780, Beijing, September, 2009.
- [45] YANG, S.; CHENG, C.; WU, R. Enhanced CAC with QoS Scheme for Improving the Efficiency of Resource Allocation on the IEEE 802.16 Network. IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA), pp. 715 – 720, Biopolis, March, 2011.
- [46] KWON, E.; LEE, J.; JUNG, K.; RYU, S. A performance model for admission control in IEEE 802.16. In 3rd International Conference. Wireless/Wired Internet Communication. (WWIC 2005), Xanthi, Greece, pp. 159 – 168, May, 2005.

- [47] EUNHYUN, J. L.; KWON, H. Y.; LEE, J.; JUNG, K. Markov model for admission control in the wireless AMC networks. IEICE Transaction Communications, vol. E89-B, no. 8, pp. 2230 – 2233, 2006.
- [48] JANG, H.; YANG, K. A QoS Aware Multi-modulation CAC for WiMax. International Symposium on Computer Science and Society (ISCCS), pp. 365 – 368, Kota Kinabalu, 2011.
- [49] THEODORIDIS, G.; PAVLIDOU, F. An SNR-based CAC algorithm for optimizing resource assignment in the downlink of M-WiMAX. IEEE Wireless Communications and Networking Conference (WCNC), pp. 1 – 6, Sydney, Australia, April, 2010.
- [50] YIN, F. Performance Comparison of Polling Mechanisms in IEEE 802.16 Networks. Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST '08), pp. 405 – 410, September, 2008.
- [51] SOHAIL, A. Performance Analysis of Bandwidth Request Mechanism of rtPS and nrtPS in WiMAX Uplink Traffic. International Conference on Computer Modelling and Simulation (UKSim), pp. 636 - 639, March, Vienna, Austria, 2012.
- [52] NIE, C.; VENKATACHALAM, M.; YANG, X. Adaptive Polling Service for Next-Generation IEEE 802.16 WiMAX Networks, IEEE Global Telecommunications Conference (GLOBECOM), pp. 4754 - 4758, November, 2007.
- [53] PARK, E.; KIM, H.; KIM, J.; KIM, H. Dynamic Bandwidth Request-Allocation Algorithm for Real-Time Services in IEEE 802.16 Broadband Wireless Access Networks. Conference on Computer Communications, pp. 852 - 860, April, 2008.
- [54] LAKKAKORPI, J.; SAYENKO, A.; MOILANEN, J. Comparison of Different Scheduling Algorithms for WiMAX Base Station. Deficit Round-Robin vs. Proportional Fair vs. Weighted Deficit Round-Robin. IEEE Wireless Communications and Networking Conference (WCNC'08), pp. 1991 – 1996, April 2008.

- [55] DONG, G.; DAI, J. Scheduling Service Algorithms for Multimedia in IEEE802.16e OFDM System. International Conference on Wireless Communications, Networking and Mobile Computing, (WiCom 2007), pp. 200 – 203, September 2007.
- [56] BELGHITH, A.; NUAYMI, L. WiMAX capacity estimations and simulation results. 67th IEEE Vehicular Technology Conference (VTC), pp. 1741 – 1745, Marina Bay, Singapore, May, 2008.
- [57] WU, C. F. Real-time Scheduling for Multimedia Services in IEEE 802.16 Wireless Metropolitan Area Network. Information Technology Journal, vol. 9, no. 6, pp. 1053-1067, June 2010.
- [58] GIDLUND, M.; WANG, G. Design of uplink schedulers for broadband wireless access networks. In 10th Annual Wireless and Microwave Technology Conference, pp. 1 – 2, August, 2009.
- [59] SAFA, H.; ARTAIL, H.; KARAM, M.; SOUDAH, R.; KHAYAT, S.; New Scheduling Architecture for IEEE 802.16 Wireless Metropolitan Area Network. In International Conference on Computer Systems and Applications, pp. 203 – 210, May, 2007.
- [60] CHEN, J.; JIAO, W.; WANG, H. A service flow management strategy for IEEE 802.16 broadband wireless access systems in TDD mode. IEEE International Conference on Communications, vol. 5, pp. 3422 – 3426, May 2005.
- [61] CICCONE, C.; ERTA, A.; LENZINI, L.; MINGOZZI, E. Performance Evaluation of the IEEE 802.16 MAC for QoS Support. IEEE Transactions on Mobile Computing, vol.6, no. 1, pp. 26 – 38, January, 2007.
- [62] KUBOTA, F. A.; BORIN, J.; FONSECA, L. S. Opportunistic Cross-layer Uplink Scheduler for the IEEE 802.16 Standard. Proceedings of the IEEE International Conference on Communications, pp. 1 – 5, June, 2011.

- [63] ZHANG, Y. WiMAX Network Planing and Optimization, CRC Press, September, 2008.
- [64] TEIXEIRA, M. A.; GUARDIEIRO, P. R. Scheduling Mechanims. Quality of Service and Resources Allocation in WiMAX, editado por Roberto Hincapie e Javier E. Sierra, Editora Intech, Fevereiro, 2012. ISBN 979-953-307-675-0.
- [65] TEIXEIRA, M. A.; GUARDIEIRO, P. R. A predictive scheduling algorithm for the uplink traffic in IEEE 802.16 networks. 12th International Conference On Advanced Communications Technology, ICACT 2010, Phoenix Park, Korea, 7 – 10 February 2010.
- [66] BELGHITH, A.; NUAYMI, L. Scheduling Techniques for WiMAX. In: Maode Ma Current Technology Developments of WiMAX Systems. Springer, pp 61 – 84, January, 2009.
- [67] BELGHITH, A.; NUAYMI, L. Design and Implementation of a QoS-included WiMAX Module for NS-2 Simulator. First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, SimuTools 2008, Marseille, France, pp. 3 – 7, March, 2008.
- [68] BELGHITH, A., NOROVJAV, O.; WANG, L. WiMAX spectrum efficiency: Considerations and simulation results. 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC), Rennes Cedex, France, pp. 503 – 510, May 2010.
- [69] RAMADAS, K.; JAIN, R.; WiMAX. System Evaluation Methodology, version 2.1 July 2007.
- [70] TARHINI, C.; CHAHED, T. On Capacity of OFDMA-based IEEE802.16 WiMAX Including Adaptive Modulation and Coding (AMC) and inter-cell Interference. 15th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN'2007), pp. 139 – 144, June, 2007.

- [71] STUBER, G. L. Principles of Mobile Communication, 2nd ed. Norwell, Springer, MA: Kluwer, 2001.
- [72] HOYMANN, C. Analysis and Performance Evaluation of the OFDM-based Metropolitan Area Network IEEE 802.16. Computer Networks, vol. 49, no. 3, pp. 341 – 363, June, 2005.
- [73] TSAI, F. C. et all. The Design and Implementation of WiMAX Module for ns-2 Simulator, Proc. ACM VALEUTOOLS 2006, Pisa, Italy, October, 2006.
- [74] TEIXEIRA, M. A.; GUARDIEIRO, P. R. Escalonamento de pacotes para o tráfego uplink em redes IEEE 802.16. 8th International Information and Telecommunication Technologies Symposium, I2TS 2009, Florianopolis, Santa Catarina, pp. 09 – 11 December 2009.
- [75] TEIXEIRA, M. A.; GUARDIEIRO, P. R. A new and efficient adaptive scheduling algorithm packets for the uplink traffic in WiMAX Networks. EURASIP Journal on Wireless Communications and Networking. September, 2011.
- [76] TEIXEIRA, M. A.; GUARDIEIRO, P. R. Adaptive Packet Scheduling for the Uplink Traffic in IEEE 802.16e Networks, International Journal of Communications Systems. January, 2012.
- [77] BORIN, J. F.; FONSECA, N. L. S. Scheduler for IEEE 802.16 Networks. IEEE Communications Letters, vol. 12, no. 4, pp. 274 – 276, 2008.
- [78] KUBOTA, F. A.; BORIN, J. F.; FONSECA, L. S. Cross-Layer Uplink Scheduler for the IEEE 802.16 Standard. IEEE Global Telecommunications Conference (GLOBECOM 2010), pp. 1 – 6, December, 2010.
- [79] BESTETTI, A.; GIAMBENE, G.; HADZIC, S.; KAKANUO, O. Performance Evaluation of a Two-Class Scheduler for WiMAX Networks. In IEEE GLOBECOM Workshops, pp. 1 – 5, November, 2008.

- [80] GAN, W.; XIAN, J.; XIE, X.; RAN, J. A cross-layer designed scheduling algorithm for WiMAX uplink. In 9th International Conference on Electronic Measurement & Instruments, pp. 122 – 127, October, 2009.
- [81] AHMET, S. Y.; IVANOVICH, M.; YEGIN, A. A survey of MAC based QoS implementations for WiMAX networks. Journal of Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 53, no. 14, pp. 2517 – 2536, September, 2009.
- [82] MANDELBRD, M. A deadline and minimal overhead scheduling algorithm for the WiMAX logical scheduler. IEEE European Wireless Conference, pp. 7 – 13, April, 2010.
- [83] GIDLUND, M.; WANG, G. Design of uplink schedulers for broadband wireless access networks. In 10th Annual Wireless and Microwave Technology Conference, pp. 1 – 2, August, 2009.
- [84] BELGHITH, A.; NUAYMI, L.; Comparison of WiMAX scheduling algorithms and proposals for the rtPS QoS class. In 14th European Wireless Conference, pp. 1 – 6, June, 2008.