

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Uma Proposta de Abordagem *Fuzzy* para Tratamento de QoS em Redes IEEE 802.16

Max Mauro de Oliveira Silva

Vitória

Novembro de 2008

Max Mauro de Oliveira Silva

Uma Proposta de Abordagem *Fuzzy* para Tratamento de QoS em Redes IEEE 802.16

Universidade Federal do Espírito Santo
Centro Tecnológico
Programa de Pós-Graduação em Engenharia Elétrica

Vitória

Novembro de 2008

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Silva, Max Mauro de Oliveira, 1982-
S586p Uma proposta de abordagem fuzzy para tratamento de QoS
em redes IEEE 802.16 / Max Mauro de Oliveira Silva. - 2008.
115 f. : il.

Orientador: Anilton Salles Garcia.

Co-Orientador: Edson de Paula Ferreira.

Dissertação (mestrado) - Universidade Federal do Espírito
Santo, Centro Tecnológico.

1. IEEE 802.16 (Normas). 2. Redes locais sem fio. 3.
Algoritmos difusos. 4. Lógica difusa. 5. Alocação de recursos. 6.
Telecomunicações - Tráfego. I. Garcia, Anilton Salles. II.
Ferreira, Edson de Paula. III. Universidade Federal do Espírito
Santo. Centro Tecnológico. IV. Título.

CDU: 621.3

Max Mauro de Oliveira Silva

Uma Proposta de Abordagem *Fuzzy* para Tratamento de QoS em Redes IEEE 802.16

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Engenharia Elétrica – Telecomunicações.

Aprovada em 04 de novembro de 2008.

Comissão Examinadora:

Prof. Dr. Anilton Salles Garcia
Universidade Federal do Espírito Santo, Orientador

Prof. Dr. Edson de Paula Ferreira
Universidade Federal do Espírito Santo, Co-Orientador

Prof. Dr. José Augusto Suruagy Monteiro
Universidade Salvador

Prof. Dr. Moisés Renato Nunes Ribeiro
Universidade Federal do Espírito Santo

Vitória, novembro de 2008.

Dedicatória

Não posso dedicar esse trabalho a outra pessoa senão a Mariana, presença fiel e carinhosa ao longo de toda essa caminhada. A você, MUITO OBRIGADO!

Agradecimentos

Acima de tudo, agradeço a Deus e aos meus anjos da guarda pelas bênçãos e pelo amparo.

Agradeço a minha família pelo apoio, incentivo e conselhos nas horas mais decisivas.

Agradeço aos meus amigos pela compreensão nas horas de ausência e pelos momentos de alegria.

Obrigado também aos meus professores, que me trouxeram lições importantes sobre o mundo e a vida. Especialmente, aos meus orientadores, Anilton e Edinho, agradeço pela confiança, apoio e fundamentais contribuições para a concretização deste trabalho.

Ao meu amor, Mariana, todo o carinho e afeto que posso dedicar, em retribuição a sua fidelidade, dedicação e ao seu companheirismo, enfim ao amor mais puro e verdadeiro que possa existir.

Max Mauro de Oliveira Silva

“O valor das coisas não está no tempo que elas duram, mas na intensidade com que acontecem. Por isso, existem momentos inesquecíveis, coisas inexplicáveis e pessoas incomparáveis...”

Fernando Pessoa

Lista de Tabelas

1	Algumas linhas da base de regras <i>fuzzy</i>	47
2	Relação entre aplicação, porta e classe IEEE 802.16.	50
3	Exemplo para ilustrar o mapeamento de conexões em objetos <i>ServiceFlows</i>	51
4	Valores de limiares configurados para o mecanismo de histerese.	53
5	Resultados do processo de inferência para algumas entradas específicas, apresentadas manualmente ao controlador.	54
6	Parâmetros de tráfego para o Cenário 2.	57
7	Parâmetros de tráfego para o Cenário 3.	60
8	Parâmetros de tráfego para o Cenário 4.	69

Lista de Figuras

1	Alocador <i>fuzzy</i> proposto em [1].	10
2	Alocador <i>fuzzy</i> proposto em [2].	10
3	Funções de pertinência associadas aos termos da variável lingüística “atraso”.	19
4	Operação de (a) união e (b) intersecção entre os conjuntos <i>fuzzy</i> A e B	19
5	Arquitetura de um Sistema Especialista <i>Fuzzy</i>	21
6	Topologia de redes IEEE 802.16 [3].	27
7	Pilha de protocolos definida no padrão IEEE 802.16 [4].	28
8	Modulação adaptativa prevista no padrão IEEE 802.16 [4].	30
9	Estrutura do quadro 802.16 no modo TDD [5].	31
10	Relacionamento entre as classes de objetos da camada MAC 802.16.	33
11	Arquitetura de QoS para a camada MAC do padrão IEEE 802.16 proposta por Malini [6].	36
12	Diagrama de classes simplificador do modelo de simulação desenvolvido em [6].	37
13	Esquema de distribuição de largura de banda entre as filas de serviço.	39
14	Mecanismo de histerese usado para identificar a ocorrência de um evento significativo que justifique uma redistribuição de recursos.	42
15	Algoritmo responsável por determinar a ocorrência de um evento.	43
16	Algoritmo responsável por acionar o alocador <i>fuzzy</i> caso seja detectado algum evento.	43
17	Partição <i>fuzzy</i> para as variáveis de entrada: (a) $\overline{W_{q_UGS}}$, (b) $\overline{W_{q_rtPS}}$	45
18	Partições <i>fuzzy</i> para as variáveis de entrada: (a) $\overline{N_{nrtPS}}$, (b) N_{BE}	46

19	Partição <i>fuzzy</i> para as variáveis de saída (a) BW_{UGS} , (b) BW_{rtPS} , (c) BW_{nrtPS} , (d) BW_{BE}	46
20	Filas no canal DL presentes na BS.	51
21	Rede para o cenário 2.	56
22	Tempo médio de espera nas filas UGS.	57
23	Alocações atribuídas à classe UGS.	58
24	Ocupação na fila BE.	58
25	Alocações atribuídas à classe BE.	59
26	Rede para o cenário 3.	60
27	Pesos associados às classes durante a simulação.	61
28	Ocupação média nas filas nrtPS.	62
29	Ocupação média na fila BE.	62
30	Tempo médio de espera nas filas UGS.	63
31	Pesos atribuídos à classe UGS.	64
32	Utilização de largura de banda pela classe UGS.	64
33	Pesos atribuídos à classe nrtPS.	66
34	Utilização de largura de banda pela classe nrtPS.	66
35	Pesos atribuídos à classe BE.	67
36	Utilização de largura de banda pela classe BE.	68
37	Rede para o cenário 4.	69
38	Pesos associados às classes durante a simulação.	70
39	Tempo médio de espera nas filas UGS.	71
40	Pesos atribuídos à classe UGS.	72
41	Utilização de largura de banda pela classe UGS.	72
42	Tempo médio de espera nas filas rtPS.	73
43	Pesos atribuídos à classe rtPS.	73

44	Utilização de largura de banda pela classe rtPS.	73
45	Ocupação média nas filas nrtPS.	74
46	Pesos atribuídos à classe nrtPS.	74
47	Utilização de largura de banda pela classe nrtPS.	75
48	Ocupação média nas filas BE.	75
49	Pesos atribuídos à classe BE.	76
50	Utilização de largura de banda pela classe BE.	76
51	Processo de inferência ilustrado no Apêndice A.	86
52	Resultado após a defuzzificação no processo de inferência ilustrado no Apêndice A.	87
53	Partição <i>fuzzy</i> para as variáveis de entrada: (a) $\overline{W_{q_UGS}}$, (b) $\overline{W_{q_rtPS}}$	91
54	Partições <i>fuzzy</i> para as variáveis de entrada: (a) $\overline{N_{nrtPS}}$, (b) N_{BE}	91
55	Prioridades: $(d)BW_{BE} > (c)BW_{nrtPS} > (b)BW_{rtPS} > (a)BW_{UGS}$	92
56	Prioridades: $(d)BW_{BE} = (c)BW_{nrtPS} = (b)BW_{rtPS} = (a)BW_{UGS}$	92
57	Prioridades: $(a)BW_{UGS} > (b)BW_{rtPS} > (c)BW_{nrtPS} > (d)BW_{BE}$	93

Lista de Acrônimos

ATM: *Asynchronous Transfer Mode*

BE: *Best Effort*

BPSK: *Binary Phase Shift Keying*

BS: *Base Station*

BWA: *BroadBand Wireless Access System*

CBR: *Constant Bit Rate*

CID: *Connection Identifier*

COG: *Center Of Gravity*

DAMA: *Demand Assigned Multiple Access*

DFPQ: *Deficit Fair Priority Queue*

DSL: *Digital Subscriber Line*

EDF: *Earliest Deadline First*

FDD: *Frequency Division Duplexing*

FIFO: *First In First Out*

FTP: *File Transfer Protocol*

GPC: *Grants per Connection*

GPSS: *Grants per Subscriber Station*

HTTP: *Hypertext Transfer Protocol*

HUMAN: *High Speed Unlicensed Metropolitan Area Network*

IEEE: *Institute of Electrical and Electronics Engineers*

IPv4: *Internet Protocol - versão 4*

LOS: *Line Of Sight*

MAC: *Medium Acces Control*

MIT: *Massachusetts Institute of Technology*

MMS: *Multimedia Messaging Service*

MPEG: *Moving Picture Experts Group*

NLOS: *Non Line Of Sight*

NRT: *non-real time*

nrtPS: *non-real time Polling Service*

NS-2: *Network Simulator - Versão 2*

OFDM: *Orthogonal Frequency Division Multiplexing*

OFDMA: *Orthogonal Frequency Multiple Access*

OSI: *Open Systems Inteerconnection*

PDU: *Protocol Data Unit*

PMP: *Point-to-Multipoint*

QoS: *Quality of Service*

QPSK: *Quadrature Amplitude Modulation*

QPSK: *Quadrature Phase Shift Keying*

RED: *Random Early Detection*

RMON: *Remote Network Monitoring*

RR: *Round Robin*

RT: *Real Time*

rtPS: *real time Polling Service*

SDU: *Service Data Unit*

SEF: *Sistema Especialista Fuzzy*

SFID: *Service Flow ID*

S-OFDMA: *Scalable-OFDM*

SS: *Subscriber Station*

TDD: *Time Division Duplexing*

TDM: *Time Division Multiplexing*

TDMA: *Time Division for Multiple Access*

ToS: *Type of Service*

UFES: *Universidade Federal do Espírito Santo*

UGS: *Unsolicited Grant Service*

WFQ: *Weighted Fair Queueing*

WiFi: *Wireless Fidelity*

WiMAX: *Worldwide Interoperability for Microwave Access*

WRR: *Weighted Round Robin*

Sumário

Lista de Acrônimos	xi
Resumo	xvii
Abstract	xviii
1 Introdução	1
1.1 Motivação	2
1.2 Justificativa da Dissertação	3
1.3 Objetivos da Dissertação	3
1.4 Metodologia	4
1.5 Principais Contribuições	5
1.6 Organização da Dissertação	5
2 Trabalhos Relacionados	7
2.1 Alocação de <i>slots</i> através da disciplina WRR.	8
2.2 Alocação de recursos para fontes NRT em redes <i>wireless</i> usando lógica <i>fuzzy</i>	9
2.3 Alocação dinâmica de recursos por meio de uma estrutura hierárquica.	10
2.4 Pré-alocação de recursos para classes com escalonamento circular das filas	11
3 Modelagem <i>Fuzzy</i>	14
3.1 Visão Geral	15
3.2 Teoria <i>Fuzzy</i>	16
3.3 Sistema Especialista <i>Fuzzy</i>	20

3.3.1	Fuzzificador	21
3.3.2	Base de Regras	22
3.3.3	Máquina de Inferência	22
3.3.4	Defuzzificador	23
4	O Padrão IEEE 802.16	25
4.1	Visão Geral	26
4.2	Camadas PHY e MAC	28
4.3	Mecanismos de QoS Previstos no Padrão	32
4.3.1	UGS	33
4.3.2	rtPS	33
4.3.3	nrtPS	34
4.3.4	BE	34
5	Estratégia <i>Fuzzy</i> para Tratamento de QoS	35
5.1	Arquitetura de QoS e Ambiente de Simulação proposto por Malini	36
5.2	Estratégia <i>Fuzzy</i> para Alteração Dinâmica de Pesos	38
5.2.1	Classificação dos Pacotes e Filas de Serviço	40
5.2.2	Gerenciamento de Filas	40
5.2.3	Alocador <i>Fuzzy</i>	44
5.2.4	Escalonadores	48
6	Resultados Numéricos	49
6.1	Descrição do Ambiente de Simulação	50
6.2	Cenário 1	53
6.3	Cenário 2	55
6.4	Cenário 3	59
6.5	Cenário 4	68

7 Conclusões e Trabalhos Futuros	78
7.1 Conclusões	79
7.2 Trabalhos Futuros	80
Referências	81
Apêndice A - Inferência <i>Fuzzy</i>	84
Apêndice B - Base de Regras Adotada para a Estratégia <i>Fuzzy</i>	88
Apêndice C - Funções de pertinência para o Cenário 2	91
Apêndice D - Composição detalhada dos tráfegos de cada cenário de simulação.	94

Resumo

O padrão IEEE 802.16 foi desenvolvido no intuito de padronizar as redes metropolitanas de acesso sem fio de banda larga. Ele introduz diversas vantagens, incluindo mecanismos de suporte a QoS na camada MAC. Para garantir qualidade de serviço a diferentes tipos de aplicações é preciso definir um algoritmo para distribuir de forma eficiente recursos entre as conexões. Considerando a natureza dinâmica e imprevisível das redes convergentes atuais, este trabalho propõe uma nova abordagem para distribuir recursos no canal downlink em redes 802.16. Como uma alternativa às soluções analíticas tradicionais, a proposta apresentada baseia-se em lógica *fuzzy*. Comparada à modelagem analítica, a abordagem *fuzzy* proporciona maior flexibilidade e generalidade às soluções devido à sua baixa especificidade e por permitir a inclusão de decisões *ad-hoc*.

Abstract

The IEEE 802.16 standard aims to define a metropolitan wireless broadband access network technology. It introduces several advantages, including the support for QoS at the MAC layer. To ensure a good quality of service to different kinds of applications, it should be defined an algorithm in order to efficiently distribute resources among the connections. Considering the dynamic and unpredictable nature of the current converging network, this work proposes a new approach, based on fuzzy logic, to distribute resources in the downlink (DL) channel as an alternative to traditional analytical solutions. Compared to the analytical modeling, the fuzzy approach provides greater flexibility and general solutions due to its low specificity and the possibility to include ad-hoc decisions.

1 *Introdução*

“Alguns homens vêem as coisas como são, e dizem ‘Por quê?’

Eu sonho com as coisas que nunca foram e digo ‘Por que não?’.”

George Bernard Shaw

Nos últimos anos, a demanda por acesso rápido à Internet, bem como por serviços multimídia, tem crescido rapidamente. Entre outros motivos, esse crescimento se sustenta sobretudo pela expansão da infra-estrutura de acesso às redes de comunicação. Nesse sentido, o padrão IEEE 802.16, conhecido comercialmente como WiMAX (*Worldwide Interoperability for Microwave Access*), apresenta-se como grande vertente tecnológica, por propiciar uma estratégia custo-efetiva ao oferecer serviços banda larga, com garantias de qualidade de serviço, em regiões além da fronteira estabelecida pelas redes baseadas em cabos.

A tecnologia WiMAX integra a família de sistemas BWA (*BroadBand Wireless Access System*), que foram concebidos exatamente para prover conectividade à chamada “última milha”, enlace entre o usuário final e a estação local. As principais vantagens do WiMAX, quando comparado a outras redes de acesso, tal como WiFi (*Wireless Fidelity*), são grande cobertura aliada a altas taxas de transmissão e suporte à mecanismos de QoS (*Quality of Service*) sofisticados. A sua camada MAC (Medium Access Control) foi desenvolvida nos moldes do processo de convergência, ou seja, para suportar a diversidade de tecnologias, protocolos e principalmente de aplicações cada vez mais sofisticadas e com diferentes requisitos de qualidade.

Embora o padrão IEEE 802.16 ofereça facilidades para o suporte à QoS, é importante destacar nesse cenário de convergência que o tráfego sobre a rede é cada vez mais imprevisível e dinâmico, resultado da sofisticação dos serviços oferecidos. Nesse contexto,

em geral, os mecanismos de garantia de qualidade de serviço não dispõem de informações exatas, e em tempo adequado, sobre o sistema de comunicação a ser controlado. Esses fatores contribuem para tornar a modelagem analítica, ferramenta amplamente adotada nas soluções dessa área, uma tarefa bastante complexa.

Em cenários como esse, não lineares e repletos de imprecisão, a modelagem *fuzzy* pode ser considerada uma alternativa mais interessante que a abordagem analítica convencional devido à sua baixa especificidade e por permitir a inclusão de decisões *ad-hoc*. Essas características propiciam soluções mais flexíveis e com maior generalidade. Em termos gerais, a lógica *fuzzy* permite o uso do conhecimento causal humano aliado ao raciocínio aproximado, sendo um meio de reproduzir a habilidade de um operador humano na tomada de decisões [7–9].

1.1 Motivação

Devido à potencialidade observada nas tecnologias BWA, incluindo o padrão IEEE 802.16, diversas pesquisas têm sido desenvolvidas em laboratórios de universidades e indústrias ao redor do mundo visando transformar as determinações técnicas, presentes nos padrões, em mecanismos concretos, adequados às operações em campo. Motivado pelas mesmas razões, em 2005, foi criado um grupo de estudos de WiMAX no Departamento de Informática da UFES (Universidade Federal do Espírito Santo), com o objetivo de discutir aspectos de Redes Metropolitanas sem Fio, principalmente em relação ao desenvolvimento de uma arquitetura de QoS adequada para transmissão em banda larga de aplicações heterogêneas em um ambiente de comunicação hostil, e propor soluções voltadas especialmente para o padrão IEEE 802.16.

Os estudos realizados dentro do grupo propiciaram o desenvolvimento de diversos trabalhos [6, 10, 11]. A dissertação [6] se destaca, entre outras razões, por contribuir para a consolidação do conhecimento acumulado pelo grupo ao propor uma arquitetura de QoS para redes WiMAX. Outra importante contribuição foi a implementação de uma primeira versão de um ambiente de simulação, aderente ao padrão IEEE 802.16, com as funcionalidades essenciais para o funcionamento e caracterização de redes baseadas nessa tecnologia. As discussões no grupo, somadas àquelas desenvolvidas no escopo de [6], indicaram muitos caminhos para pesquisa. Módulos como o de controle de admissão, policiamento de tráfego e escalonamento de serviços, tanto na BS (*Base Station*) quanto na SS (*Subscriber Station*), ainda precisam ser implementados. Além disso, melhorias

e novas funcionalidades devem ser incorporadas ao modelo de simulação. Todos esses fatores são razões motivadoras para o desenvolvimento deste trabalho.

1.2 Justificativa da Dissertação

O documento IEEE 802.16, em muitos aspectos, preocupa-se essencialmente em definir as diretrizes sobre “o que fazer”, mas deixa a cargo dos desenvolvedores decidirem “como fazer”. Funcionalidades como controle de admissão, policiamento de tráfego e escalonadores de recursos precisam ser especificados e implementados. Dentre todos os mecanismos, os escalonadores de recursos figuram entre os que mais influenciam o desempenho do sistema. Um algoritmo de escalonamento eficiente é essencial para garantir que os requisitos de QoS sejam atendidos.

Em linhas gerais, a justificativa para os esforços nesta pesquisa está relacionada com o crescimento do uso de aplicações multimídia, cujas exigências de nível de serviço somam-se aos desafios de implementação da comunicação em uma interface aérea. Desse modo, devido à existência no padrão de diversos pontos em aberto, inclusive sobre o escalonamento de recursos, há uma grande demanda por pesquisa e desenvolvimento para que a tecnologia possa, de fato, dar suporte às diversas aplicações multimídia com alta qualidade e bom desempenho.

1.3 Objetivos da Dissertação

Esta dissertação tem como objetivo geral contribuir para a arquitetura de recursos em redes IEEE 802.16. Especificamente, este trabalho pretende:

1. Investigar a aplicabilidade de lógica *fuzzy* no provisionamento de recursos em redes WiMAX.
2. Propor uma nova estratégia para alocação de largura de banda no canal de *down-link* em redes IEEE 802.16, baseada em lógica *fuzzy*, como alternativa às soluções analíticas.
3. Implementar a estratégia proposta de forma a permitir incorporá-la ao ambiente de simulação WiMAX desenvolvido em [6] e validá-la.
4. Apresentar possíveis desafios na tarefa de provisionamento de recursos em redes WiMAX para direcionar trabalhos futuros.

Com isso, é dado mais um passo para disponibilizar à comunidade científica, especialmente ao meio acadêmico, uma arquitetura de QoS completa para redes 802.16, bem como um módulo gratuito e de código aberto para simulação de redes WiMAX.

1.4 Metodologia

O primeiro passo no desenvolvimento deste trabalho foi o estudo de tecnologias de redes metropolitanas sem fio, especificamente sistemas BWA. Para tal, foi realizada uma busca na literatura por livros, artigos, dissertações e teses sobre o assunto (e.g. [12–14]). Nessa fase também foi feito um estudo detalhado do padrão IEEE 802.16, com atenção especial aos mecanismos de QoS [5].

Após discussões dentro do grupo WiMAX, citado anteriormente, e estudo em detalhes da dissertação [6], decidiu-se por trabalhar em um mecanismo inteligente de alocação de recursos. Com a linha de trabalho definida, os esforços se concentraram em buscar na literatura propostas de algoritmos e estratégias de escalonamento de serviços em redes metropolitanas sem fio. A grande maioria dos trabalhos, como em [1, 15, 16], trazem soluções baseadas em modelagem analítica e, em geral, não detalhavam com profundidade como a proposta poderia ser implementada em um cenário real. Essas razões, somadas às conveniências da lógica *fuzzy*, pesaram na decisão de desenvolver uma solução alternativa à abordagem tradicional, baseada em sistemas inteligentes.

O passo seguinte foi o estudo da teoria e modelagem *fuzzy*. Para isso, foi realizado um estudo dirigido sobre o tema com a produção de um documento como atividade final. A partir desse resumo foi elaborado o Capítulo 3 desta dissertação.

Após o domínio dos conceitos de redes BWA e modelagem *fuzzy*, o próximo passo consistiu em desenvolver um modelo esquemático para a proposta de escalonamento inteligente de recursos, representado na Figura 13. A partir do esquemático, teve início a fase de implementação. A proposta foi implementada usando a linguagem C++ para permitir sua incorporação ao ambiente de simulação WiMAX. Após a implementação, seguiu-se a fase de ajustes e validação do modelo. Essa etapa foi realizada incorporando a proposta deste trabalho ao modelo de simulação mencionado anteriormente. Para finalizar, todas as informações reunidas ao longo dos estudos foram consolidadas neste documento.

1.5 Principais Contribuições

Dentre os principais resultados alcançados com a elaboração deste trabalho, as seguintes contribuições podem ser relacionadas:

1. Uma nova proposta, baseada em lógica *fuzzy*, para o escalonamento de recursos no canal *downlink* em redes WiMAX;
2. Novos módulos para o modelo de simulação proposto em [6], voltados especificamente para o gerenciamento e atendimento das filas de serviço do canal *downlink*;
3. Validação da proposta em um ambiente de simulação que reproduz as principais características de redes WiMAX;
4. Análise de uso de modelagem *fuzzy* no tratamento de QoS em redes WiMAX.

1.6 Organização da Dissertação

De modo a expor toda a proposta e os trabalhos realizados, essa dissertação foi dividida em 7 capítulos e 4 apêndices. A seguir, cada capítulo é brevemente descrito.

Capítulo 1: Introdução

Inicia a discussão abordando a idéia defendida na dissertação. Apresenta a motivação e as justificativas para o trabalho. Traz uma revisão da metodologia e enumera os objetivos, resultados esperados e as principais contribuições percebidas no trabalho.

Capítulo 2: Trabalhos Relacionados

Faz uma revisão bibliográfica sobre o escalonamento de recursos em redes WiMAX ao apresentar algumas das principais pesquisas relacionadas ao tema. Na relação de trabalhos discutidos, encontra-se o algoritmo proposto em [6], principal referência para esta dissertação.

Capítulo 3: Modelagem *Fuzzy*

Introduz os conceitos envolvidos no processo de modelagem *fuzzy*. Inicialmente é feita uma discussão entre a modelagem tradicional e a *fuzzy*. Na seqüência, são enunciados os principais pontos na teoria, para, em seguida, abordar o funcionamento de um sistema de inferência *fuzzy*, sistema esse que constitui a base para a proposta desenvolvida neste trabalho.

Capítulo 4: O padrão IEEE 802.16

Traz uma apresentação dos pontos mais relevantes que envolvem o padrão IEEE 802.16, incluindo os aspectos técnicos das camadas PHY e MAC. Uma seção é dedicada exclusivamente à descrição dos mecanismos de QoS previstos no padrão, incluindo os serviços diferenciados de escalonamento.

Capítulo 5: Estratégia *Fuzzy*

Discute a estratégia *fuzzy* proposta nesta dissertação, sendo incorporada à arquitetura de QoS desenvolvida em [6]. Para isso, é apresentado não só a arquitetura de QoS do trabalho citado, como também o ambiente de simulação WiMAX que é utilizado para validar a estratégia.

Capítulo 6: Resultados Experimentais

Apresenta as simulações realizadas e os resultados computacionais obtidos. Neste capítulo, são explicados os parâmetros de simulação configurados para cada um dos quatro cenários concebidos para avaliar a estratégia *fuzzy* de escalonamento. São realizadas análises dos resultados, visando investigar a viabilidade do uso de lógica *fuzzy* na tarefa de provisionamento de recursos.

Capítulo 7: Conclusões e Trabalhos Futuros

Encerra o trabalho com as considerações finais e sugestões de trabalhos futuros.

Apêndices

Finalmente, o Apêndice A complementa as discussões realizadas no Capítulo 3 através de uma descrição passo a passo de processo de inferência *fuzzy*. Um exemplo simples, porém ilustrativo, encerra as explicações. O Apêndice B traz a base de regras usada no sistema *fuzzy* proposto. Finalmente, o Apêndice C apresenta as funções de pertinência adotadas no Cenário 2 e o Apêndice D detalha a composição do tráfego empregado em cada um dos cenários de teste do Capítulo 6.

2 *Trabalhos Relacionados*

“O segredo de progredir é começar. O segredo de começar é dividir as tarefas árduas e complicadas em tarefas pequenas e fáceis de executar, e depois começar pela primeira.”

Mark Twain

ESTE trabalho concentra esforços em soluções para o provisionamento de recursos em redes WiMAX. Neste capítulo, alguns trabalhos direcionados para esse propósito são apresentados e discutidos no intuito de traçar um paralelo com a proposta desta dissertação. Em particular, o trabalho de Malini ganha destaque por ser um dos pilares da estratégia defendida nos capítulos seguintes.

2.1 Alocação de *slots* através da disciplina WRR.

No trabalho *Scheduling solution for the IEEE 802.16 base station* [17], os autores propõem uma solução para alocação de *slots* de tempo em uma BS 802.16; baseada nas classes de serviço IEEE 802.16 e requisição de banda, no caso de canal *uplink* ou estado de fila (ocupação), para o canal *downlink*. Para o caso específico do canal *downlink*, a BS destina uma fila separada para cada conexão. Além disso, cada fila é associada a uma classe de serviço. O estado de uma fila é atualizado sempre que um novo pacote chega a essa fila ou quando a deixa. Outro ponto importante diz que o mesmo escalonador é usado tanto no canal *downlink* quanto *uplink*, uma vez que, segundo os autores, cálculos similares podem ser empregados para a alocação de *slots* em ambos os canais.

O escalonador proposto baseia-se na disciplina WRR (*Weighted Round Robin*), em que os pesos representam o número de *slots* alocados para cada conexão. A alocação, no caso do canal *downlink*, é determinada pelo tamanho de fila e parâmetros de QoS associados a cada conexão. A estratégia apresentada consiste em três estágios. No primeiro estágio é alocado um valor mínimo de recursos para garantir os requisitos básicos de QoS de cada conexão. Conforme a classe de serviço associada à fila de atendimento, é adotada uma fórmula diferente para alocação. Os *slots* não alocados no primeiro estágio, caso haja algum, são distribuídos entre as conexões em um segundo estágio. Nesse caso, as conexões contempladas com *slots* excedentes podem ser escolhidas aleatoriamente ou podem ser definidas de acordo com a ordem de prioridade da classe de serviço. No terceiro estágio, a BS pode dividir os *slots* alocados entre vários quadros de forma a cumprir requisitos de tempo, tal como *jitter*. Segundo os autores, apenas a implementação do primeiro estágio é estritamente necessária para o funcionamento do escalonador, mas eles frisam a importância do segundo estágio para a eficiência na utilização da largura de banda disponível.

Com essas explicações, pode-se estabelecer um paralelo entre as propostas do artigo [17] e desta dissertação. Em primeiro lugar, ambos os trabalhos adotam as características das classes de serviço e estado das filas de atendimento como critérios para a decisão de alocação. No entanto, o artigo citado adota parâmetros quantificados numericamente para compor as fórmulas, como por exemplo, o valor de *Maximum Sustained Traffic Rate* no caso de UGS. Esse é um exemplo clássico de solução baseada em modelagem analítica para o provisionamento de recursos. Por outro lado, na abordagem *fuzzy* deste trabalho, as propriedades de cada classe são quantificadas subjetivamente, de acordo com o conhecimento de um especialista, usando termos ou instâncias num certo

número de conjuntos *fuzzy*, conforme é mais bem explicado em capítulos posteriores.

2.2 Alocação de recursos para fontes NRT em redes *wireless* usando lógica *fuzzy*.

O trabalho *Resource Allocator for Non Real-Time Traffic in Wireless Networks Using Fuzzy Logic* [1] baseia-se nas condições do canal para decidir sobre a alocação de recursos. O ambiente de redes indicado é formado tanto por tráfego NRT (*non-real time*) quanto RT (*real time*), sendo o tráfego RT de maior prioridade. Os fluxos compartilham os recursos via mecanismo TDMA (*Time Division for Multiple Access*). Nessa técnica o canal aéreo é segmentado em quadros que, por sua vez, é dividido em *slots* de tempo, para, então serem distribuídos entre as fontes de dados.

Inicialmente, são alocados recursos para tráfego RT, só então os *slots* excedentes podem ser compartilhados entre as fontes NRT, as quais obedecem ao algoritmo de balde furado. A distribuição é decidida por um controlador *fuzzy*, cujas entradas são a porcentagem de recursos alocados para tráfego RT, %RT, e um indicador de qualidade do canal para cada fonte NRT. A saída do controlador é um vetor de tamanho igual ao número de fontes NRT, contendo os parâmetros m e r de controle da política de balde furado de cada fonte NRT. Para garantir um nível mínimo de QoS, um valor mínimo de *slots* de tempo é alocado para cada fonte NRT. Alocação acima desse valor somente ocorre se houver *slots* restantes. Nesse sentido, o parâmetro m determina a alocação de cada fonte NRT. Adicionalmente, as fontes NRT transmitem usando códigos de repetição (*repetition coding*), com cada pacote da l -ésima fonte sendo repetido rep_l vezes. O número de repetições, rep_l , é inversamente proporcional ao parâmetro r_l .

O alocador *fuzzy* ilustrado na Figura 1 é implementado por meio de dois sistemas de inferência *fuzzy* distintos, sem interconexão: sistema de controle de taxa de repetição e um sistema de alocação de *slots*. São duas as variáveis lingüísticas de entrada: $RATE_{eff}$ com espaço de entrada formado por seis conjuntos *fuzzy* (*very low, low, moderate, moderate high, high e very high*) e %RT com espaço de entrada formado por cinco conjuntos *fuzzy* (*very low, low, moderate, high e very high*). Observe que somente o indicador de qualidade de canal $RATE_{eff}$ é dado como entrada para o sistema de controle de taxa de repetição, enquanto para a alocação, são passados o indicador e a porcentagem %RT.

Esse trabalho ilustra bem o uso de lógica *fuzzy* na atividade de alocação de recursos. Entretanto, o escopo definido pelo trabalho (generalização de redes *wireless* e apenas uma

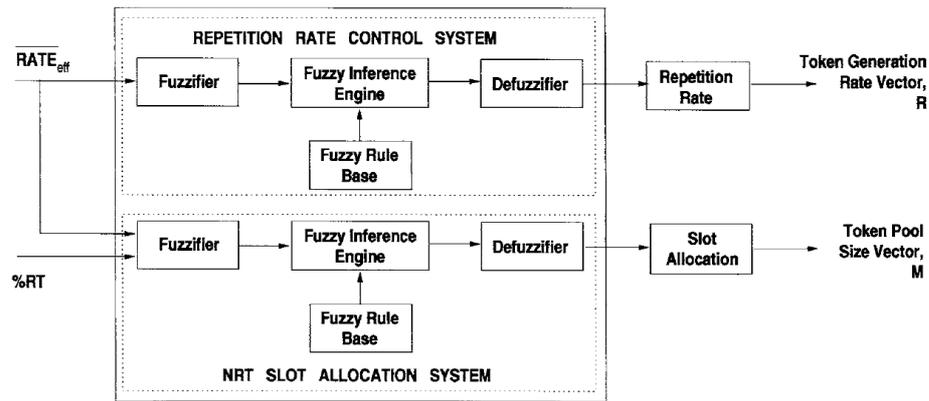


Figura 1: Alocador *fuzzy* proposto em [1].

classe de tráfego NRT), assim como os critérios adotados nas decisões (recursos disponíveis e qualidade do canal) indicam simplificações do mundo real de operação. Além disso, os critérios para definição das funções de pertinência, bem como da base de regras não estão tão claros.

2.3 Alocação dinâmica de recursos por meio de uma estrutura hierárquica.

No trabalho *A Service Flow Management Strategy for IEEE 802.16 Broadband Wireless Access Systems in TDD Mode* [2], os autores propõem uma estrutura de escalonamento hierárquico como meio de suportar todos os tipos de fluxos de serviço. O escalonador opera no modo TDD (*Time Division Duplexing*) e utiliza uma combinação de diversas disciplinas de escalonamento. A Figura 2 ilustra o escalonador proposto nesse trabalho.

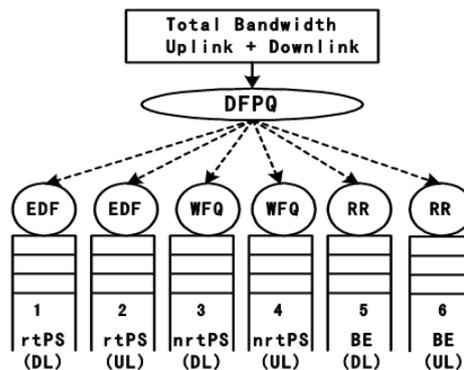


Figura 2: Alocador *fuzzy* proposto em [2].

São definidas seis filas de acordo com seu sentido de comunicação (*downlink* - *DL*

ou *uplink* - *UL*) e classe de serviço: DL_{rtPS} e UL_{rtPS} , DL_{nrtPS} e UL_{nrtPS} , DL_{BE} e UL_{BE} . Cabe ressaltar que os recursos para classe UGS são alocados antes do processo de escalonamento propriamente dito. Para os demais serviços, a proposta consiste em uma estrutura hierárquica de distribuição de recursos em dois níveis.

O primeiro nível se baseia na disciplina DFPQ (*Deficit Fair Priority Queue*) para escalonar recursos entre as filas de atendimento. A ordem de prioridade seguida é: $DL_{rtPS} > UL_{rtPS} > DL_{nrtPS} > UL_{nrtPS} > DL_{BE} > UL_{BE}$. Pode-se observar que o sentido DL tem prioridade maior que o sentido UL. A BS mantém uma lista de filas ativas, ou seja, filas não vazias, de modo que o DFPQ obedeça à prioridade estabelecida, concedendo banda somente às filas ativas. A quantidade de banda concedida é função do parâmetro *maximum sustained traffic rate* associado às conexões.

O segundo nível consiste em disciplinas de escalonamento para atendimento interno às filas. Assim, filas *rtPS* são atendidas com EDF (*Earliest Deadline First*), tal que pacotes com menor limite de tempo são escalonados primeiro. As filas *nrtPS* são atendidas por meio da disciplina WFQ (*Weighted Fair Queueing*), na qual o peso atribuído a cada conexão é obtido como uma razão entre o parâmetro *minimum reserved traffic rate* associado à conexão e à soma dos parâmetros *minimum reserved traffic rate* de todas as conexões. Finalmente, o restante da largura de banda é dividido entre as conexões BE por meio da política RR (*Round Robin*).

Um aspecto interessante nesse trabalho é o uso da estrutura hierárquica para a distribuição dos recursos. Essa idéia também é adotada nesta dissertação, como é visto posteriormente. Uma das vantagens dessa estratégia é permitir isolar a prioridade entre classes de serviço da prioridade entre as filas de atendimento dentro de uma mesma classe de serviço. Assim, os algoritmos de escalonamento podem ser customizados para cada tipo de serviço, enquanto requisitos comuns a filas de uma mesma classe são tratados em um nível superior.

2.4 Pré-alocação de recursos para classes com escalonamento circular das filas

Em seu trabalho, Malini propôs um algoritmo de escalonamento de recursos no canal *downlink*. Seu pseudocódigo é ilustrado em seguida.

```

bandwidth_UGS    = WEIGHT_UGS    * burstsize;
bandwidth_rtPS   = WEIGHT_rtPS   * burstsize;
bandwidth_nrtPS  = WEIGHT_nrtPS  * burstsize;
bandwidth_BE     = WEIGHT_BE     * burstsize;

enquanto (há banda disponível && há dados a serem transmitidos)
faça para cada classe de serviço k
{
    enquanto (há banda bandwidth_k da classe k && há dados a serem
    transmitidos da classe k) faça para cada ServiceFlow da classe k
    {
        retire o primeiro pacote da fila e adicione à PDU de transmissão;
    }
    adicione a largura de banda restante à banda disponível para a
    próxima classe de serviço a ser atendida;
}

```

Inicialmente, a largura de banda *burstsize* disponível para o sub-quadro *downlink* sendo montado é dividida entre as classes de serviço UGS, rtPS, nrtPS e BE, por meio dos pesos *WEIGHT_UGS*, *WEIGHT_rtPS*, *WEIGHT_nrtPS* e *WEIGHT_BE*. De forma indireta, os pesos atribuídos podem ser interpretados como uma alocação estática de banda para cada classe. O algoritmo é executado em dois laços aninhados. No laço externo, enquanto houver banda de *downlink* total e dados a serem transmitidos, o algoritmo atua circularmente sobre as quatro classes de serviço definidas, obedecendo à ordem de prioridade estabelecida no padrão, ou seja, primeiro UGS, seguida de rtPS, nrtPS e por fim BE. No laço interno, também de forma circular, os *ServiceFlows* que compõem a classe de serviço em questão são atendidos. Enquanto houver banda alocada àquela classe e dados daquela classe de serviço a serem transmitidos, o primeiro pacote da fila do *ServiceFlow* é retirado e adicionado à PDU (*Protocol Data Unit*) de transmissão. Caso os recursos alocados para uma determinada classe não sejam totalmente utilizados, o excedente é repassado para a classe de prioridade imediatamente inferior (no caso da classe BE, caso haja banda excedente, a transferência é feita para a classe UGS). Com isso, o sub-quadro *downlink* somente é transmitido com espaço vazio caso não exista mais nenhum pacote de nenhuma classe de serviço aguardando transmissão.

É importante mencionar que o autor destaca em seu trabalho a flexibilidade oferecida pelo algoritmo proposto para diferentes cenários de operação. Conforme a demanda

de tráfego de cada classe de serviço, as constantes $WEIGHT_UGS$, $WEIGHT_rtPS$, $WEIGHT_nrtPS$ e $WEIGHT_BE$ podem ser alteradas para utilizar a largura de banda da melhor maneira possível. No entanto, essa alteração referida pelo autor é realizada no início da operação do sistema e, se não houver intervenção humana, os valores permanecem indefinidamente iguais, mesmo que o perfil da rede mude ao longo do tempo. Ocorre que, no cenário atual de redes convergentes, a demanda de tráfego de cada classe de serviço varia dinamicamente, de forma imprevisível. Nesse sentido, é inviável a configuração estática de pesos, tal como proposto no algoritmo.

Outro aspecto a ser levantado é que o autor considerou em seus testes cargas de tráfego que demandam no máximo 90% da capacidade de transmissão. Essa abordagem determinística para controle de QoS, baseada em garantias fortes, limita o proveito de recursos e são adequadas em casos restritos, como de aplicação de missão crítica. É mais comum e eficiente a abordagem estatística em que são admitidas conexões cuja soma das demandas ultrapasse 100% da largura de banda total. Isso pode ser feito, pois espera-se que, na média, a utilização efetiva seja inferior à capacidade total de transmissão. Essa situação reforça a importância de alocação dinâmica, porque nos instantes de tempo em que a demanda supera o limite da capacidade, o sistema deve intervir para garantir recursos às aplicações de maior prioridade.

Também é importante ressaltar que o algoritmo foi implementado de tal modo que filas podem sofrer com o problema de *starvation*. Essa situação ocorre, pois a ordem de atendimento das filas dentro de uma mesma classe é sempre igual. Dessa maneira, se a quantidade de dados a serem transmitidos supera os recursos disponíveis em um número seguido de quadros DL, as filas serão atendidas sempre na mesma ordem e o recurso irá se esgotar sempre antes de atender as últimas filas dentro de uma classe.

3 *Modelagem Fuzzy*

“Na medida em que as proposições da matemática se referem à realidade, elas não são certas; na medida em que são certas, elas não se referem à realidade.”

Albert Einstein

A lógica *fuzzy* permite modelar não só o conhecimento objetivo como também o subjetivo. Em geral, isso permite construir soluções mais gerais e flexíveis. Este capítulo traz uma breve descrição dos conceitos da teoria *fuzzy* e sua aplicação em sistemas especialistas. Vale destacar que a estratégia proposta nesta dissertação tem como elemento central um sistema especialista *fuzzy*.

3.1 Visão Geral

Tradicionalmente, problemas de engenharia são modelados através de ferramentas matemáticas clássicas, como as equações diferenciais. Essa abordagem faz uso do conhecimento objetivo humano, materializado nos modelos matemáticos convencionais, para representar o sistema em estudo com a maior fidelidade possível. O modelo é tanto mais fiel à realidade quanto maior for o número de informações disponíveis sobre o domínio do problema e a principal vantagem dessa abordagem reside na precisão de modelo alcançada. O problema dessa abordagem é que, em muitas situações, processos dinâmicos tornam-se demasiadamente complexos e freqüentemente não é possível reunir informações ou identificar parâmetros que permitam caracterizar precisamente o sistema. Outro aspecto a ser considerado é que um modelo muito complexo pode levar a formulações intratáveis. Finalmente, deve-se observar que, muitas vezes, a precisão não é requisito fundamental para a solução desejada. Essa discussão é traduzida pelo princípio da incompatibilidade apresentado em [18]:

“Conforme a complexidade de um sistema aumenta, nossa habilidade de fazer declarações precisas e significativas sobre o seu comportamento diminui, até alcançar um limite além do qual precisão e relevância tornam-se características mutuamente exclusivas.”

Contrapondo o conhecimento objetivo humano, a natureza de diversos problemas é inerentemente subjetiva. O mundo real, incluindo grande parte das aplicações de interesse na área de engenharia, apresenta propriedades que são definidas de forma vaga e imprecisa. É comum, por exemplo, em uma conversa entre dois especialistas em redes, o uso de termos como “prioridade alta”, “tempo médio de espera baixo” ou “custo alto”. Apesar do caráter vago e impreciso, a mensagem é perfeitamente compreendida lingüisticamente pelos interlocutores. Isso se deve à capacidade humana de recuperar informações, conectar pensamentos e idéias de maneira não-linear e tomar decisões [19]. Tais características permitem, por exemplo, a um operador humano aplicar seu raciocínio aproximado no intuito de intervir em um sistema e ajustar seus parâmetros de funcionamento. Aliás, em muitas situações, a modelagem do problema em um determinado domínio passa pela representação do conhecimento acumulado por um especialista nesse domínio, ou seja, essa abordagem faz uso não só do conhecimento humano objetivo como também do subjetivo ou difuso. Para o conhecimento difuso, que neste contexto também pode ser referenciado como conhecimento nebuloso ou *fuzzy*, a questão central é a imprecisão implícita na

linguagem natural, conforme observado em [7]. Sentenças como “maior prioridade” é de difícil tratamento com as ferramentas matemáticas usuais.

Motivado por esses desafios, em 1965, Lotfi Zadeh, professor no Departamento de Engenharia Elétrica e Ciências da Computação da Universidade da Califórnia, em Berkeley, publicou seu trabalho que introduziu a teoria dos conjuntos *fuzzy* ou nebulosos com o objetivo de traduzir em termos matemáticos as informações de caráter impreciso ou vago, expressas por um conjunto de regras lingüísticas [7]. Essa teoria proporcionou um meio de combinar de forma efetiva tanto o conhecimento humano objetivo quanto o subjetivo, tornando-se uma ferramenta com habilidades suficiente para lidar com problemas não-lineares e complexos cuja formulação é imprecisa, vaga e até mesmo ambígua. Nesse sentido, a teoria dos conjuntos nebulosos aplicada em um contexto lógico, como o de sistemas baseado em conhecimento, passa a ser referenciada como lógica nebulosa, lógica difusa ou lógica *fuzzy*. A lógica *fuzzy*, devido sua baixa especificidade e por permitir a inclusão de regras *ad-hoc*, produz soluções mais flexíveis e com maior generalidade se comparadas às equivalentes analíticas, também reconhecidas pela descrição em termos quantitativos convencionais. Em termos gerais, os sistemas que usam a lógica nebulosa buscam modelar o modo de raciocínio aproximado, tentando imitar a habilidade humana de tomar decisões racionais em um ambiente de incerteza e imprecisão [20].

O papel central da lógica *fuzzy* é fornecer um meio para representar o conhecimento *fuzzy* ou difuso. Conforme enunciado em [19], a forma humana de resolver um problema parte da estruturação do seu conhecimento sobre esse problema em conceitos gerais para então observar as relações essenciais entre esses conceitos. Esse processo de modelagem “*top-down*” propicia uma forma de converter relações essencialmente gerais e imprecisas, obtidas em um primeiro momento, em algoritmos operacionais mais detalhados, em uma segunda fase. De fato, uma vez obtido um modelo *fuzzy*, este deve constituir um aproximador numérico que quantifica adequadamente a subjetividade e as imprecisões do problema. A eficácia do modelo pode ser verificada através da relação entre entrada e saída do sistema, uma vez que o modelo é construído baseado na resposta que se deseja.

3.2 Teoria Fuzzy

A teoria clássica estabelece conjuntos através de uma relação de pertinência dicotômica entre os elementos de um universo de discurso U . Dito de outro modo, para um conjunto qualquer A , cada elemento $x \in U$ deve assumir apenas um de dois estados

possíveis: $x \in A$ ou $x \notin A$. Essa dicotomia pode ser quantificada ou representada de várias maneiras, em particular por 1 e 0, respectivamente. Numa visualização, pode-se perceber que os conjuntos clássicos possuem fronteira abrupta ou rígida com respeito aos seus elementos. A teoria dos conjuntos nebulosos, por sua vez, foi concebida para permitir uma transição gradual entre a noção de pertencer completamente ou não pertencer completamente a um dado conjunto. Para isso foi introduzido o conceito de graus de pertinência. Enquanto na teoria clássica a imagem de um elemento pela função de pertinência assume valores em $\{0, 1\}$, na teoria *fuzzy* essa imagem pode assumir qualquer valor dentro do intervalo $[0, 1]$. Formalmente, um conjunto *fuzzy* \tilde{A} no universo de discurso U pode ser entendido como um mapeamento dos elementos de U em um espaço de pertinência M através da função de pertinência $\mu_{\tilde{A}}(x)$, isto é, $\mu_{\tilde{A}} : U \rightarrow M$. Assim, \tilde{A} é definido como um conjunto de pares ordenados da seguinte forma:

$$\tilde{A} = \left\{ \left(x, \mu_{\tilde{A}}(x) \mid x \in U \right) \right\} \quad (3.1)$$

Geralmente, M é representado pelo intervalo unitário $[0, 1]$. A função de pertinência indica o grau de compatibilidade entre x e o conceito expresso por \tilde{A} , ou seja:

- $\mu_{\tilde{A}}(x) = 1$ indica que x é completamente compatível com \tilde{A} ;
- $\mu_{\tilde{A}}(x) = 0$ indica que x é completamente incompatível com \tilde{A} ;
- $0 < \mu_{\tilde{A}}(x) < 1$ indica que x é parcialmente compatível com \tilde{A} , com grau $\mu_{\tilde{A}}(x)$.

Segundo essa definição, um conjunto A na teoria clássica pode ser entendido como um conjunto *fuzzy* \tilde{A} específico, denominado usualmente de *crisp*, para o qual $\mu_{\tilde{A}} : U \rightarrow (0, 1)$, ou seja, o espaço de pertinência é bivalente, “sim” ou “não”. Portanto, a teoria *fuzzy* pode ser vista como uma generalização da teoria clássica de conjuntos. Da mesma forma, a lógica *fuzzy* pode ser considerada uma extensão da lógica clássica bivalente. É definida como uma lógica que suporta os modos de raciocínio aproximados, em vez de exatos, tal como são enunciados na lógica clássica.

Na abordagem *fuzzy*, as variáveis do problema não assumem valores numéricos num domínio infinito, mas assumem termos ou instâncias num certo número de conjuntos nebulosos, definindo uma classificação ou agregação em categorias gerais ou conjuntos de possíveis soluções. Dessa maneira, valores escalares são mapeados em conjuntos nebulosos parcialmente sobrepostos, que particionam um determinado universo de discurso. O

número de partições que formam o universo de discurso de uma variável determina o grau de precisão que a solução exige para o problema.

A classificação das variáveis de um problema com suas imprecisões, em termos quantitativos, é implementada na forma de variáveis lingüísticas. Uma variável lingüística é um elemento simbólico utilizado para descrever o conhecimento que assume valores num conjunto de palavras ou sentenças em linguagem natural ou artificial. Por exemplo, “prioridade” é uma variável lingüística que pode assumir os valores: “muito baixa”, “baixa”, “média”, “alta”, “muito alta”, entre outros. Os termos lingüísticos, portanto, representam conjuntos nebulosos que particionam o universo de discurso e o número de partições determina a precisão da solução. Este conceito foi introduzido por Zadeh para prover um meio de caracterizar de forma aproximada um fenômeno complexo ou que não seja tão bem definido a ponto de permitir descrevê-lo em termos quantitativos convencionais.

A variável lingüística é caracterizada formalmente por uma quintupla $(x, T(x), U, G, M)$, onde x é o nome da variável; $T(x)$ é o conjunto de termos de x , ou seja, o conjunto com os nomes dos valores que a variável x pode assumir; U é o universo de discurso; G é uma regra sintática para gerar os nomes de valores para x ; M é uma regra semântica para associar cada valor de x com seu significado, isto é, uma função que associa a cada termo em $T(x)$ uma função de pertinência. Como exemplo, seja a variável lingüística “atraso”, medida em “ms” e definida sobre $U = [0 - 200]$ com $T(x) = \text{“baixo”, “moderado”, “alto”}$ e G baseado no senso comum. Assim,

- $M(\text{“baixo”}) =$ Conjunto nebuloso que representa “atrasos aproximadamente menores que 50 ms” com pertinência $\mu_{baixo}(x)$.
- $M(\text{“moderado”}) =$ Conjunto nebuloso que representa “atrasos aproximadamente iguais a 100 ms” com pertinência $\mu_{moderado}(x)$.
- $M(\text{“alto”}) =$ Conjunto nebuloso que representa “atrasos aproximadamente acima de 150 ms” com pertinência $\mu_{alto}(x)$.

Do mesmo modo que na teoria clássica, uma operação na teoria *fuzzy* representa uma relação entre conjuntos que produz outro conjunto. São cinco as operações básicas sobre conjuntos *fuzzy*, também chamadas operações básicas de Zadeh: complemento, interseção, união, igualdade, subconjunto. No contexto deste trabalho, apenas as operações de interseção e união, ilustradas na Figura 4 são empregadas. Dessa maneira, sejam os conjuntos *fuzzy* A e B definidos no universo de discurso U :

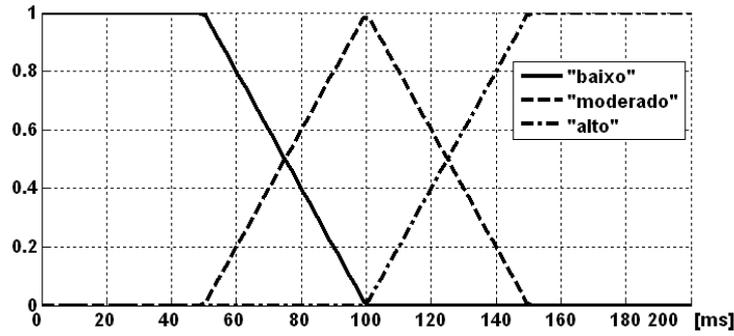


Figura 3: Funções de pertinência associadas aos termos da variável linguística “atraso”.

- 1. União:** A união entre os conjuntos *fuzzy* A e B , denotada como $A \cup B$, é definida pela Equação 3.2:

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \equiv \mu_A(x) \vee \mu_B(x) \quad \forall x \in U. \quad (3.2)$$

- 2. Intersecção:** A intersecção de conjuntos *fuzzy* A e B , denotada como $A \cap B$, é definida pela Equação 3.3:

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \equiv \mu_A(x) \wedge \mu_B(x) \quad \forall x \in U. \quad (3.3)$$

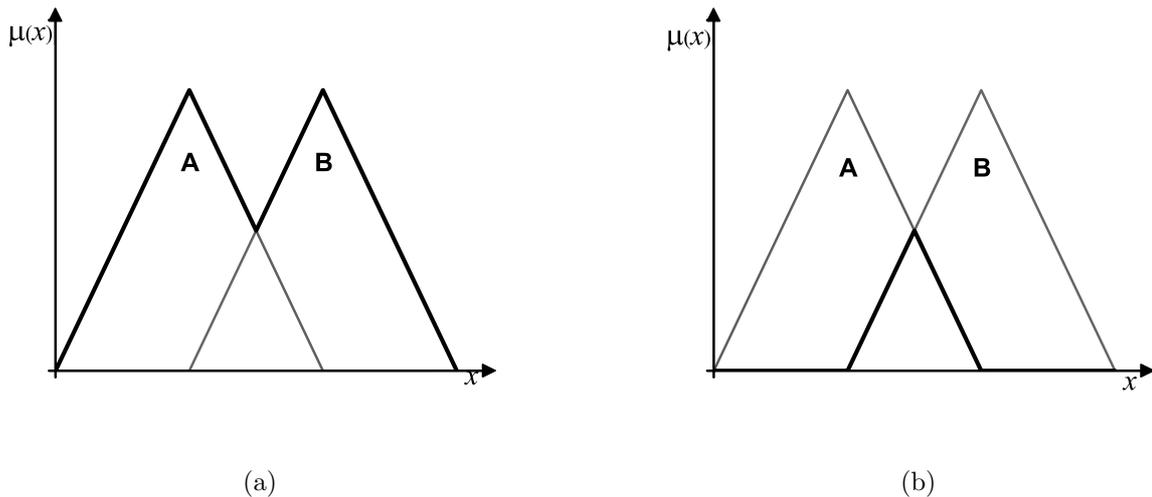


Figura 4: Operação de (a) união e (b) intersecção entre os conjuntos *fuzzy* A e B .

Além das operações básicas de Zadeh, a teoria *fuzzy* oferece uma série de operações paramétricas e não paramétricas que podem ser adotadas em cada situação, de acordo

com as especificidades do sistema a ser modelado. Essas operações são generalizações das operações básicas e conferem à teoria *fuzzy* ferramentas para tratar de uma extensa gama de problemas. Dentre todas as alternativas, merecem destaque as generalizações das operações de união e interseção, referenciadas como normas triangulares (t-normas) e conormas triangulares (t-conormas), respectivamente [9].

3.3 Sistema Especialista *Fuzzy*

Um sistema especialista visa solucionar problemas que são resolvidos habitualmente por um especialista humano. Por isso, requerem conhecimento sobre a habilidade, a experiência e as heurísticas usadas pelo especialista no domínio do problema [9, 19]. Além disso, a lógica *fuzzy* propicia um meio eficiente para representar o conhecimento especializado em um ambiente de incertezas e imprecisão. Por essas razões, um SEF (Sistema Especialista *Fuzzy*) apresenta-se como uma alternativa bastante adequada para o problema de distribuição justa e eficiente de recursos entre serviços heterogêneos e dinâmicos, tal como ocorre em redes convergentes ou multimídia.

O espaço de entrada de um SEF é formado pelas variáveis características do problema ou as variáveis de interesse, que são monitoradas periodicamente. As variáveis que se deseja controlar ou que são objeto de decisão formam o espaço de saída. Seguindo a definição de variáveis lingüísticas, o vetor de entrada X , composto pelas variáveis lingüísticas de interesse x_i , e o vetor de saída Y , composto pelas variáveis lingüísticas sob controle y_i , podem ser definidos respectivamente por:

$$X = \{(x_i, U_i, \{T_{x_i}^1, T_{x_i}^2, \dots, T_{x_i}^{k_i}\}, \{\mu_{x_i}^1, \mu_{x_i}^2, \dots, \mu_{x_i}^{k_i}\}) |_{i=1..n}\} \quad (3.4)$$

$$Y = \{(y_i, V_i, \{T_{y_i}^1, T_{y_i}^2, \dots, T_{y_i}^{k_i}\}, \{\mu_{y_i}^1, \mu_{y_i}^2, \dots, \mu_{y_i}^{k_i}\}) |_{i=1..m}\} \quad (3.5)$$

onde as variáveis lingüísticas de entrada x_i formam o espaço de entrada fuzzy $U = U_1 \times U_2 \times \dots \times U_n$ e as variáveis lingüísticas de saída y_i formam o espaço de saída fuzzy $V = V_1 \times V_2 \times \dots \times V_n$. O tamanho ou cardinalidade do conjunto de termos $|T(x_i)| = k_i$ é chamada de partição fuzzy de x_i . A partição *fuzzy* determina a granularidade da decisão obtida em um SEF. Quanto maior for o número de termos, mais refinada será a inferência realizada pelo SEF, e em contrapartida maior será a complexidade do sistema.

A literatura apresenta diversos modelos de inferência *fuzzy* [9, 21] que determinam de

que modo é extraído e processado o conhecimento armazenado na forma de declarações condicionais *fuzzy*. Basicamente, os modelos são classificados em dois grupos: os clássicos, compreendendo o modelo de Mamdani e o de Larsen, e os de interpolação, representado pelo modelo de Takagi-Sugeno e o de Tsukamoto. Os modelos diferem quanto à forma de representação dos termos na premissa das regras de decisão, quanto à representação das ações de decisão e quanto aos operadores utilizados para implementação do sistema especialista.

Para este trabalho foi adotado o modelo de Mamdani. A arquitetura do SEF para esse modelo é composta, basicamente, por quatro partes: Fuzzificador, Base de Regras *Fuzzy*, Máquina de Inferência *Fuzzy* e Defuzzificador. A Figura 5 ilustra o diagrama típico de um sistema especialista *fuzzy* que utiliza o modelo de inferência de Mamdani. Cada um desses módulos é mais bem explicado em seguida. Adicionalmente, o Apêndice A complementa as explicações dadas neste capítulo com uma descrição passo a passo do processo de inferência *fuzzy*, sendo encerrado com um exemplo ilustrativo.

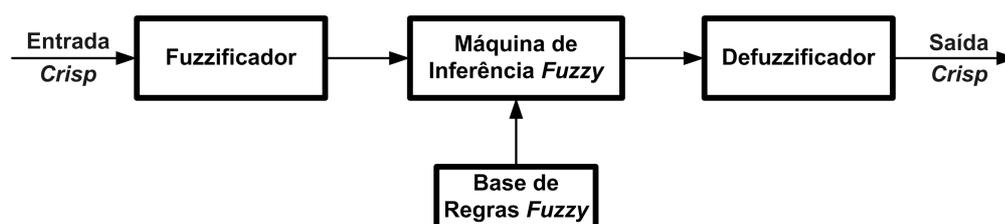


Figura 5: Arquitetura de um Sistema Especialista *Fuzzy*.

3.3.1 Fuzzificador

O bloco Fuzzificador implementa uma interface entre as medidas observadas, as quais caracterizam o estado do sistema, e o sistema especialista *fuzzy*. Sua função é realizar o mapeamento entre os valores de entrada *crisp* em conjuntos nebulosos para que possam se tornar instâncias de variáveis lingüísticas.

Cada termo lingüístico pode ser interpretado como um conjunto *fuzzy*. A cada conjunto *fuzzy*, é associada uma função de pertinência que indica com que grau o valor medido pertence ao conjunto. Assim, na instanciação, para um determinado valor *crisp*, a função de pertinência associada a um termo lingüístico retorna um número real do intervalo $[0, 1]$. Quanto mais próximo de 1, mais a medida é compatível com o valor lingüístico. Cabe ressaltar que um mesmo valor pode pertencer com certo grau, a um conjunto e, em outro grau, a outro conjunto.

3.3.2 Base de Regras

A base de regras caracteriza a relação entre os conjuntos *fuzzy* do espaço de entrada e os do espaço de saída. É na base de regras que se materializa o conhecimento especialista sobre um determinado domínio. É formada por uma coleção de regras com a seguinte estrutura:

$$Se < antecedente >, Então < conseqüente > \quad (3.6)$$

O antecedente é composto por um conjunto de premissas que, quando satisfeitas (mesmo parcialmente), determinam o processamento do conseqüente da regra por um mecanismo de inferência *fuzzy*. Esse é o processo de disparo da regra. Por sua vez, o conseqüente é composto por um conjunto de ações ou diagnósticos que são gerados com o disparo da regra. As partes que compõem um antecedente ou conseqüente de uma regra são relacionados pelo conectivo lógico “e” e o encadeamento entre as regras é feito pelo conectivo “ou”. Os conseqüentes das regras disparadas são processados em conjunto para gerar uma resposta determinística para cada variável de saída do sistema [19].

É importante destacar que devem existir tantas regras quantas forem necessárias para associar completamente as combinações dos termos das variáveis lingüísticas do espaço de entrada. Portanto, quanto maior a granularidade de cada variável lingüística, maior o número de entradas na base de regras e, portanto, maior a complexidade computacional na inferência de respostas. Desse modo, pelo princípio multiplicativo da matemática, o número de regras nr de um sistema *fuzzy* é determinado pela Equação 3.7:

$$nr = |T(x_1)| \times |T(x_2)| \times \dots \times |T(x_i)|, \quad (3.7)$$

onde $x_1 \dots x_i$ são as variáveis lingüísticas de entrada.

3.3.3 Máquina de Inferência

A máquina de inferência, por sua vez, simula o processo de tomada de decisão de um especialista humano. Dito de outro modo, a máquina de inferência recebe valores *fuzzy* provenientes do módulo de interface de entrada, processa as regras existentes na Base de Regras e gera um conjunto *fuzzy* de saída a partir da composição de todas as regras disparadas.

A correlação entre o antecedente e o conseqüente de uma regra se faz por meio de uma operação de implicação entre conjuntos *fuzzy*. São várias as definições de implicação que podem ser adotadas. Entretanto, neste trabalho é empregada a regra de implicação Mamdani, tal como definida na Equação 3.8.

$$A \rightarrow B = \min[\mu_A(u), \mu_B(v)] \equiv \mu_A(u) \wedge \mu_B(v) \quad (3.8)$$

A regra semântica geralmente utilizada para o processamento de inferências com o modelo de Mamdani é a chamada inferência *max-min*. Ela utiliza as operações de união e interseção entre conjuntos da mesma forma que Zadeh, por meio dos operadores de máximo e mínimo respectivamente [19].

Após as devidas explicações, considere o seguinte exemplo de inferência em que é empregado o silogismo condicional *modus ponens* generalizado.

$$\begin{array}{l} \text{Premissa 1 : } \textit{se } x \textit{ é } A, \textit{ Então } y \textit{ é } B \\ \text{Premissa 2 : } x \textit{ é } A' \\ \hline \text{Conclusão : } y \textit{ é } B' \end{array} \quad (3.9)$$

O *modus ponens* generalizado se caracteriza por permitir a inferência de um resultado mesmo se os antecedentes das premissas não são compatíveis, como é o caso: $A \neq A'$. Nesse sentido, a operação que representa as ações executadas no exemplo anterior é definida na Equação 3.10.

$$\mu_{B'}(x) = \max_{\forall x \in X} \min[\mu_{A'}(x), \min[\mu_A(x), \mu_B(x)]] \quad (3.10)$$

3.3.4 Defuzzificador

O defuzzificador desempenha o papel oposto do fuzzificador, mapeando o universo *fuzzy* em universo *crisp*. Dito de outro modo, o defuzzificador converte informações qualitativas em uma informação quantitativa, produzindo uma única ação de controle precisa. O procedimento compreende a identificação do domínio das variáveis de saída num correspondente universo de discurso e com a ação de controle *fuzzy* produzida executa-se uma ação de controle não *fuzzy*.

Os métodos de defuzzificação mais utilizados são:

- COG (Center of Gravity): Calcula-se o centro de gravidade para um dado conjunto *fuzzy* de saída, a abscissa do ponto de centro de gravidade correspondente, e a utiliza como valor escalar de saída.
- MOM (Mean of Maximum): Utiliza a média dos valores máximos de pertinência do conjunto a ser defuzzificado.
- SOM (Smallest of Maximum): Utiliza o menor valor de pertinência dentre os máximos de pertinência do conjunto *fuzzy* a ser defuzzificado.
- LOM (Largest of Maximum): Utiliza o maior valor de pertinência dentre os máximos de pertinência do conjunto *fuzzy* a ser defuzzificado.

No projeto desta dissertação foi adotado o método COG para o módulo de defuzzificação do sistema especialista.

4 *O Padrão IEEE 802.16*

“Se conheces o inimigo e conheces a ti mesmo, não deves temer pelo resultado de cem batalhas; se te conheces, mas não conheces o inimigo, por cada batalha ganha sofrerás também uma derrota; se não conheces o inimigo nem a ti mesmo, sucumbirás em todas as batalhas.”

Sun Tzu

O objetivo do padrão é especificar a interface aérea, incluindo a camada de acesso ao meio (MAC) e a camada física (PHY), para redes metropolitanas sem fio (WMAN), com diferenciação de serviços. Como será apresentado, o 802.16 fornece apenas o suporte para a implementação dessa arquitetura, possibilitando que cada fabricante desenvolva a sua própria solução. Este capítulo apresenta uma visão geral do IEEE 802.16 e uma breve descrição da camada física e MAC, bem como, da arquitetura de QoS definida pelo padrão.

4.1 Visão Geral

Embora as redes banda larga estejam disponíveis há algum tempo, o acesso ainda é limitado para grande parte das pessoas. Em geral, residências, comércios e pequenas e médias empresas localizados fora do alcance das provedoras de serviços DSL (*Digital Subscriber Line*) não são atendidos por serviços de acesso rápido. Além disso, expandir a infra-estrutura baseada em tecnologias tradicionais, tais como par trançado e cabo coaxial, para alcançar milhões de residências e escritórios em áreas de subúrbio ou rurais é inviável economicamente. Por esses motivos, em julho de 1999, começaram os trabalhos do grupo 802.16 criado pelo IEEE (*Institute of Electrical and Electronics Engineers*), cuja missão foi desenvolver um padrão para sistemas BWA (*Broadband Wireless Access*). Este padrão visa atender a demanda crescente por conectividade em banda larga na região conhecida como “última milha”, enlace entre o usuário final e a estação local mais próxima.

A primeira versão do documento, IEEE Std 802.16TM-2001, foi lançada em abril de 2002 especificando a operação na faixa de frequência de 10-66 GHz, portanto, restrito aos enlaces de comunicações com linha de visada ou LOS (*Line Of Sight*). A versão IEEE Std 802.16TM-2004 [5], correspondente à letra “d”, foi lançada em outubro de 2004 para operar nas frequências entre 2 e 11 GHz, permitindo a comunicação entre estações sem linha de visada ou NLOS (*Non Line Of Sight*). Essas versões são destinadas apenas a usuários fixos ou nômades. Mas, em dezembro de 2005, foi aprovado pelo IEEE o padrão IEEE 802.16e, adicionando mobilidade ao IEEE 802.16TM-2004.

Por razões práticas, o escopo dos padrões necessita ser reduzido a um número mínimo de funcionalidades essenciais à operação da tecnologia. Além disso, apenas desenvolver um novo padrão não garante que a tecnologia terá espaço no mercado alvo. Por essas razões, em junho de 2001, um grupo de fabricantes de componentes, desenvolvedores de equipamentos, provedores de serviço e indústrias interessadas fundou o WiMAX Forum[®], uma associação sem fins lucrativos composta atualmente por mais de 522 membros. Esse fórum tem por objetivo promover e certificar a compatibilidade dos equipamentos de rede com os padrões de banda larga sem fio e a interoperabilidade entre equipamentos de fabricantes distintos. Adicionalmente, pretende divulgar e inserir a tecnologia em mercados espalhados por todo o mundo. Para tanto, o WiMAX Forum desenvolveu perfis de aplicação dos padrões e definiu testes de conformidade a serem realizados em laboratórios certificados. Assim surgiu o termo WiMAX, sinônimo comercial do padrão IEEE 802.16 [22].

Como foi afirmado em parágrafos anteriores, inicialmente, o padrão foi concebido para solucionar o problema de conectividade na “última milha”, oferecendo uma alternativa às redes de acesso tradicionais. Entretanto, as vantagens apresentadas por essa tecnologia, tais como grande cobertura aliada a altas taxas de transmissão, mecanismos sofisticados para suporte à QoS, baixo custo de manutenção e implantação e alta escalabilidade, permitem o seu emprego em outros cenários como *backhaul* de redes *Mesh* ou celular, conexão entre escritórios e cobertura banda larga com portabilidade e mobilidade.

Em relação à sua arquitetura, as redes 802.16 baseiam-se no conceito de células e reuso de frequências, assim como em telefonia celular. Basicamente, são formadas por uma estação BS (*Base Station*) e uma ou mais SSs (*Subscriber Stations*), tal como apresentado na Figura 6. Para a versão IEEE Std 802.16TM-2004, o raio típico de uma célula é de 5 a 8 Km em aplicações NLOS, podendo chegar a um valor máximo teórico de 50 Km em aplicações LOS. Para a versão móvel do padrão, o raio típico é de 1,5 a 5 Km. O padrão define dois modos básicos de operação: PMP (*Point-to-Multipoint*) e *Mesh*. Enquanto no modo *Mesh* uma SS pode se comunicar com outras SSs e com a BS, no modo PMP é permitido à SS somente se comunicar com a BS.



Figura 6: Topologia de redes IEEE 802.16 [3].

O documento especifica as camadas física (PHY) e MAC do modelo de referência OSI (*Open Systems Interconnection*). O esquema de camadas definido no padrão é ilustrado na Figura 7.

A camada física é constituída por duas subcamadas, sendo a mais inferior responsável por lidar diretamente com o canal de transmissão. A subcamada de convergência e transmissão permite à camada de enlace de dados não se preocupar com o esquema de mo-

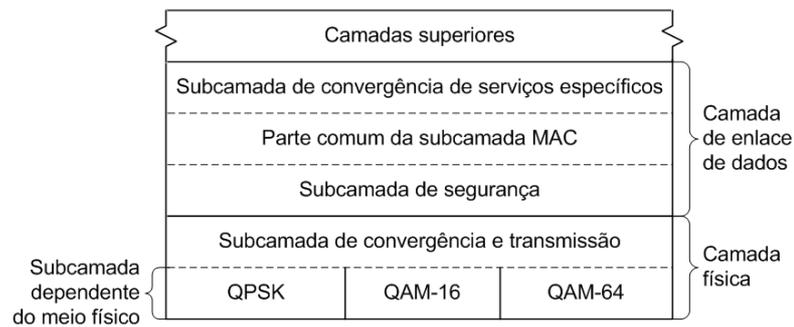


Figura 7: Pilha de protocolos definida no padrão IEEE 802.16 [4].

dulação e codificação empregados durante a transmissão. O subnível de segurança responde por funcionalidades como autenticação, intercâmbio seguro de chaves e encriptação.

A parte comum da subcamada MAC é responsável pelo gerenciamento dos canais de comunicação, provendo serviços essenciais como acesso ao sistema, alocação de banda, estabelecimento de conexão e manutenção de conexão. Destaca-se o fato dessa camada ser orientada à conexão, aspecto diferenciado em relação aos demais padrões que contribui para que o IEEE 802.16 ofereça suporte nativo à garantia de qualidade de serviço. É capaz de lidar tanto com tráfego contínuo, como, por exemplo, CBR (*Constant Bit Rate*), como em rajadas, por exemplo, FTP (*File Transfer Protocol*), dando suporte a uma extensa gama de serviços.

Finalmente, a subcamada de convergência de serviços específicos tem a finalidade de mapear o fluxo de dados oriundo da camada de rede externa em SDUs (*Service Data Units*). Esse mapeamento inclui classificar as SDUs em um fluxo de serviço e associá-las a um identificador de conexão, o CID (*Connection Identifier*). Desse modo, a camada de enlace de dados, orientada à conexão, pode operar independente da tecnologia utilizada na camada de rede. Esses mecanismos são mais bem explicados em capítulos posteriores. Atualmente, o padrão IEEE 802.16 especifica CSs para redes ATM (*Asynchronous Transfer Mode*) e para redes baseadas em pacotes.

4.2 Camadas PHY e MAC

As redes 802.16 podem operar em faixas de frequência licenciada e não licenciada, abaixo ou acima de 11 GHz, embora as aplicações mais atraentes, sob o ponto de vista comercial, sejam implementadas em frequências abaixo de 11 GHz e em banda licenciada. A frequência específica de operação depende de aspectos regulatórios, que podem variar de

país para país. Aqui, no Brasil, a versão do padrão com suporte à mobilidade está prevista para operar em banda licenciada de 2,5 GHz, enquanto a versão para usuários fixos ou nômades pode trabalhar nas faixas licenciadas de 3,5 e 10,5 GHz e na não licenciada de 5,8 GHz [23]. Em redes 802.16d, os canais podem ter tamanhos múltiplos de 1,25, 1,5 ou 1,75 MHz, até o limite de 20 MHz [24]. O benefício imediato dessa característica é o melhor aproveitamento do espectro disponível, uma vez que o valor de multiplicidade pode ser escolhido de tal forma que a soma das larguras de cada canal se aproxime ou iguale o valor total de espectro destinado ao sistema como um todo.

O padrão IEEE 802.16 prevê algumas opções de camada PHY. O documento IEEE Std 802.16TM-2004 especifica cinco alternativas [5]:

- WirelessMAN-SCTM : Interface aérea que usa a modulação com portadora simples. Opera na faixa de 10-66GHz, portanto destinada a aplicações com linha de visada;
- WirelessMAN-SCa TM : Interface aérea que usa a modulação com portadora simples. Opera em frequências licenciadas abaixo de 11 GHz, portanto destinada a aplicações sem linha de visada;
- WirelessMAN-OFDM TM : Interface aérea que adota um esquema com 256 portadoras em OFDM (*Orthogonal Frequency Division Multiplexing*). Opera em frequências licenciadas abaixo de 11 GHz, portanto destinada a aplicações sem linha de visada. Múltiplos acessos de diferentes SSs são coordenados pelo protocolo TDMA (*Time Division for Multiple Access*). Esta versão suporta a topologia *Mesh* e subcanalização no canal reverso (*uplink*);
- WirelessMAN-OFDMA: Adota um esquema para interface aérea com 2048 portadoras em OFDM. Opera em frequências licenciadas abaixo de 11 GHz, portanto destinada a aplicações sem linha de visada. Nesse caso, os múltiplos acessos são coordenados pelo esquema OFDMA (*Orthogonal Frequency Multiple Access*). Suporta subcanalização nos enlaces reverso (*uplink*) e direto (*dowlink*).
- WirelessHUMAN: esta opção não é, de fato, uma nova especificação de camada, mas sim uma definição de funcionalidades complementares às especificadas para as opções SCa, OFDM, OFDMA, visando adequá-las à operação em banda não licenciada na faixa 5-6 GHz, sendo por esse motivo denominada HUMAN (*High Speed Unlicensed Metropolitan Area Network*).

O documento IEEE 802.16e define mais uma camada, específica para o cenário de mobilidade e incompatível com as anteriores devido à variante de protocolo OFDMA adotada:

- **WirelessMAN-SOFDMA:** Assim como as interfaces anteriores, se baseia em OFDM, mas adota uma variante do protocolo OFDMA chamada S-OFDMA (*Scalable-OFDM*).

As distâncias envolvidas em uma célula 802.16 podem ser de vários quilômetros e as condições do canal podem variar de uma região para outra dentro de uma mesma célula de cobertura. Devido a isso, é natural que a potência percebida na BS varie de SS para SS, afetando a relação sinal-ruído percebida nas estações. Por esse motivo, o padrão IEEE 802.16 permite que parâmetros de transmissão, incluindo esquemas de modulação e codificação, sejam ajustados individualmente para cada SS [12]. Esse ajuste associado a alguns parâmetros como altura e ganho de antenas, potência de sinal, densidade de usuário em cada célula, condições do canal aéreo influencia na taxa de transmissão obtida. Teoricamente, segundo o padrão, pode-se alcançar até 120 Mbps para a camada WirelessMAN-SCTM, em um canal de 28 MHz, e até 75 Mbps para WirelessMAN-SCaTM, em um canal de 20 MHz. Um valor mais próximo do real, para um canal TDM (*Time Division Multiplexing*) de 10 MHz, com uma razão DL/UL de 3:1, é 25 Mbps e 6,7 Mbps para os sentidos DL e UL, respectivamente [24]. A Figura 8 ilustra a estratégia de modulação adaptativa em que são estabelecidos três perfis diferentes de transmissão.

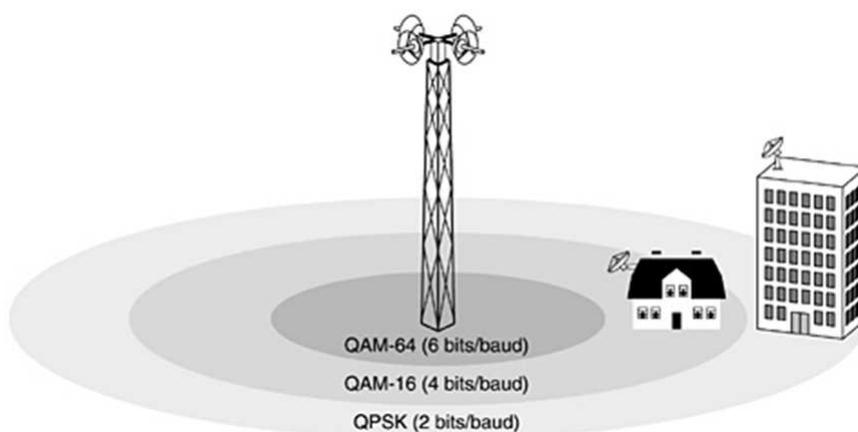


Figura 8: Modulação adaptativa prevista no padrão IEEE 802.16 [4].

Todas as camadas PHY especificadas operam no formato de quadros, cada um dividido em dois canais lógicos: sub-quadro *Downlink* (DL) e sub-quadro *Uplink* (UL). O padrão admite dois esquemas de duplexação dos sentidos de comunicação: em FDD (*Frequency*

Division Duplexing), os canais DL e UL usam faixas de frequência diferentes e, em TDD (*Time Division Duplexing*), os dois canais compartilham a mesma frequência e os dados são transmitidos em tempos diferentes. O esquema TDD tem sido mais empregado pela indústria de equipamentos, pois permite ajuste dinâmico na proporção de banda alocada para cada canal, permitindo um suporte eficiente de tráfego assimétrico. Além disso, os transceptores TDD são menos complexos o que representa custos menores.

Especificamente, no modo TDD, o canal aéreo é segmentado no tempo em quadros de tamanho fixo. Cada quadro é dividido em dois sub-quadros: um para o sentido DL e outro para UL. A duração desses sub-quadros é controlada dinamicamente pela BS de acordo com a demanda por banda em cada sentido [5]. O canal DL opera em *broadcast* por meio de um sinal TDM e somente a BS transmite nesse sub-quadro. O canal UL, por sua vez, é compartilhado por todas as SSs em um esquema TDMA. A Figura 9 ilustra o esquema TDD.

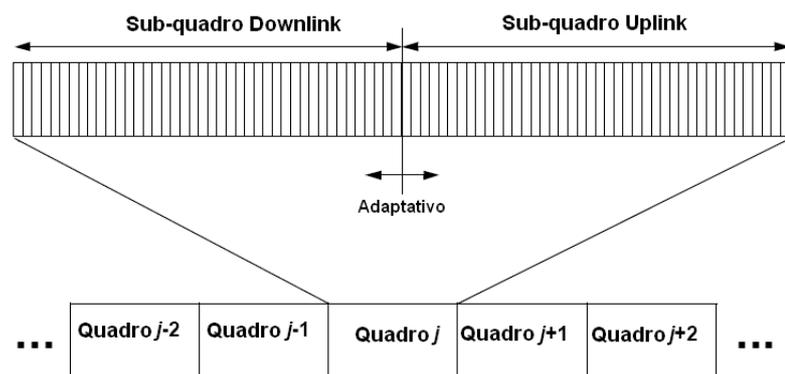


Figura 9: Estrutura do quadro 802.16 no modo TDD [5].

A camada MAC também incorpora a função de escalonamento de recursos, tarefa originalmente realizada em camadas superiores. A BS é responsável por essa tarefa tanto para o canal DL quanto para o UL. Especificamente, no canal DL, a BS decide sobre a distribuição de recursos conforme a demanda instantânea percebida nas filas de pacotes oriundos do núcleo da rede. Já para o canal UL, cada SS expressa sua intenção de transmitir enviando requisições de banda (*BW-Request*) para a BS. Nessas requisições, a SS informa qual a banda solicitada e quais os parâmetros de QoS exigidos pela aplicação. É com base nessas requisições que a BS, antes de qualquer coisa, verifica se possui recursos suficientes para aceitar essa nova demanda e também distribui recursos para cada SS. Tal esquema de alocação de recursos é chamado DAMA - TDMA (*Demand Assigned Multiple Access - Time Division Multiple Access*).

O mapeamento de alocação no canal UL é difundido pela BS através de uma mensagem chamada UL-MAP no começo de cada quadro. Após receber o UL-MAP, cada SS transmite os seus dados nos segmentos de tempo pré-definidos. De maneira análoga, as informações sobre as definições de acesso ao canal DL são difundidas pela mensagem DL-MAP. Esta mensagem especifica a duração do sub-quadro, número identificador do sub-quadro, número identificador do canal de DL e informações sobre sincronizações dentro de cada sub-quadro DL.

Outro ponto de destaque no padrão é o suporte à requisição de oportunidades de transmissão baseada em conexão (*Grants per Connection - GPC*) ou por estação (*Grants per Subscriber Station - GPSS*). No modo GPC, a BS trata cada conexão separadamente e a banda é explicitamente alocada para a conexão. Esse mecanismo é adequado para casos em que não existam muitas conexões por SS. Por outro lado, no modo GPSS, a BS trata todas as conexões de uma SS como uma entidade única e aloca recursos para a SS. Nesse caso, é preciso um escalonador adicional na SS para distribuir os recursos alocados entre as conexões.

4.3 Mecanismos de QoS Previstos no Padrão

O padrão IEEE 802.16 provê mecanismos de suporte a QoS em ambos os sentidos de comunicação. Para tanto, a camada MAC é orientada à conexão. Toda comunicação estabelecida é feita através de uma conexão unidirecional, negociada entre a BS e cada SS. Cada conexão recebe um identificador de 16 bits, chamado CID (*Connection ID*), permitindo 64k conexões em cada canal, UL e DL. O CID pode ser considerado um identificador mesmo para tráfegos não-orientados à conexão como IP, pois ele agrega em si informações sobre o destino e o contexto do fluxo trafegado [6].

O propósito das conexões é transportar as PDUs mapeadas em objetos denominados pelo padrão como fluxo de serviço. Um fluxo de serviço pode ser interpretado como um serviço de transporte da camada MAC que provê transporte unidirecional de pacotes tanto no canal DL quanto UL. Caracteriza-se por um conjunto de parâmetros de QoS como, atraso, *jitter* e taxa de transmissão, cujos valores são negociados antecipadamente entre a SS e a BS. Cada fluxo possui uma identificação única, o SFID (*Service Flow ID*), que é associado ao CID correspondente. Pode-se afirmar que esse é o principal mecanismo do padrão relacionado ao provisionamento de QoS.

O padrão também especifica serviços diferenciados de escalonamento de dados em que

cada tipo é definido por um conjunto de parâmetros de QoS que quantificam aspectos de seu comportamento. Cada conexão é associada a um único tipo de serviço. A versão IEEE Std 802.16TM-2004 especifica quatro opções de escalonamento, os quais definem diferentes níveis de QoS: *Unsolicited Grant Service* (UGS), *real-time Polling Service* (rtPS), *non-real-time Polling Service* (nrtPS) e *Best Effort* (BE). Cada uma dessas opções é mais bem explicada em seguida. A Figura 10 ilustra o relacionamento entre as classes de objetos usadas na arquitetura de QoS prevista no padrão.

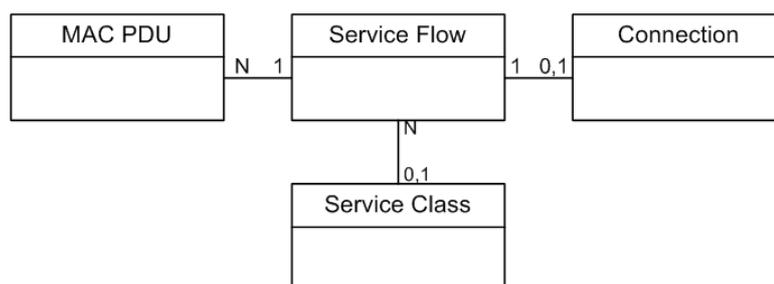


Figura 10: Relacionamento entre as classes de objetos da camada MAC 802.16.

4.3.1 UGS

A classe de serviço UGS *Unsolicited Grant Service* especifica transmissões em tempo real com características de tráfego CBR. Devido à necessidade de banda constante, os parâmetros de serviço dos fluxos dessa classe no canal UL são negociados uma única vez no momento em que a conexão é estabelecida. A partir de então, a banda concedida é garantida até o fim da conexão. Alguns exemplos de aplicações para essa classe são VoIP (*Voice over IP*) sem supressão de silêncio e tráfego E1/T1. Os parâmetros obrigatórios de QoS que definem esse tipo de tráfego são: *Maximum Sustained Traffic Rate*, *Maximum Latency*, *Tolerated Jitter* e *Request/Transmission Policy*.

4.3.2 rtPS

O tráfego rtPS *real-time Polling Service* é adequado a fluxos de dados de tempo real com pacotes de tamanho variável, tais como vídeo MPEG (*Moving Picture Experts Group*) e jogos interativos. Nessa classe, a SS recebe, em intervalos periódicos, a oportunidade de solicitar largura de banda no canal UL através de um mecanismo de *unicast polling* realizado pela BS. Os Os parâmetros obrigatórios são: *Minimum Reserved Traffic Rate*, *Maximum Sustained Traffic Rate*, *Maximum Latency* e *Request/Transmission Policy*.

4.3.3 nrtPS

Essa classe, *non-real-time Polling Service* (nrtPS), representa aplicações do tipo rajadas, que necessitam de um nível de serviço maior que o “melhor esforço”. Esperam-se rajadas de grande tamanho, refletindo um fluxo de dados mais intenso se comparado à classe BE. Também recebem oportunidades de solicitar banda através de *unicast polling*, mas em uma frequência menor que o serviço rtPS. Alguns exemplos são transferência de arquivo e MMS (*Multimedia Messaging Service*). Os parâmetros mandatórios de QoS associados são: *Minimum Reserved Traffic Rate*, *Maximum Sustained Traffic Rate*, *Traffic Priority*, *Request/Transmission Policy* e *Packet Loss*.

4.3.4 BE

A classe BE *Best Effort* é dirigida ao tráfego de “melhor esforço”, para o qual não existe garantia de QoS. Para essa classe, as SSs que desejam transmitir tráfego BE disputam entre si, em um segmento de tempo reservado, a oportunidade de enviar suas requisições de largura de banda. Exemplos tradicionais são aplicações HTTP (*Hypertext Transfer Protocol*) e terminal remoto. Os parâmetros de QoS nesse caso são: *Maximum Sustained Traffic Rate*, *Traffic Priority*, *Request/Transmission Policy* e *Packet Loss*.

5 *Estratégia Fuzzy de Distribuição de Recursos no Canal Downlink em Redes IEEE 802.16*

“...Descobri como é bom chegar quando se tem paciência. E para se chegar, onde quer que seja, aprendi que não é preciso dominar a força, mas a razão. É preciso antes de mais nada querer.”

Amyr Klynk

ESTE capítulo apresenta a estratégia *fuzzy* de distribuição de recursos no canal *downlink* em redes IEEE 802.16. A estratégia é composta por módulos, cada um apresentado em detalhes ao longo das seções. Entre os módulos, figura em destaque o sistema especialista *fuzzy*, elemento central da estratégia. A seção correspondente traz as decisões de modelagem das funções de pertinência, base de regras e demais componentes do sistema.

5.1 Arquitetura de QoS e Ambiente de Simulação proposto por Malini

Em sua dissertação, Malini [6] propôs uma arquitetura para a camada MAC do padrão IEEE 802.16, apresentada na Figura 11, de modo a suportar serviços multimídia distribuídos. Seu trabalho baseou-se em propostas anteriores de arquiteturas [15, 16, 25] e teve o mérito de complementá-las e detalhá-las. Nesse trabalho, são apresentadas as funcionalidades essenciais que a camada MAC deve oferecer para suportar os serviços multimídia planejados, bem como os blocos que implementam tais funcionalidades.

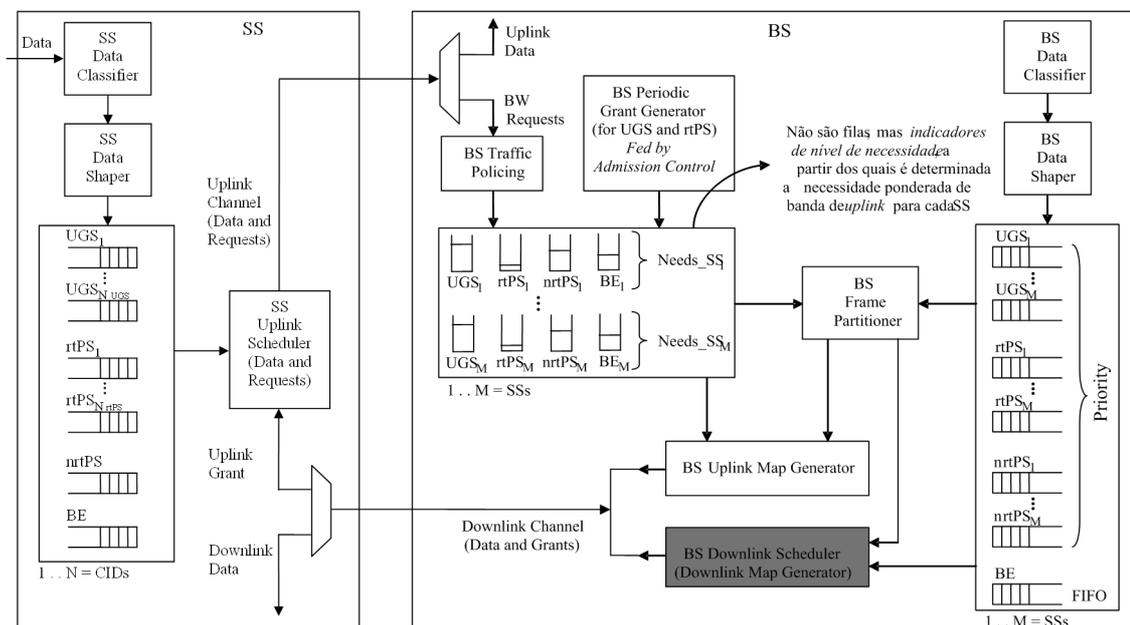


Figura 11: Arquitetura de QoS para a camada MAC do padrão IEEE 802.16 proposta por Malini [6].

O autor também apresentou um algoritmo de atendimento das classes de serviço no canal DL, visto na seção 2.4. Esse algoritmo é executado como parte da funcionalidade do módulo *BS Downlink Scheduler*, parte em cinza na Figura 11. Um ponto importante nessa arquitetura, para o escopo desta dissertação, diz respeito ao esquema de filas adotado na BS. Tal esquema é formado por no máximo M filas UGS, M filas $rtPS$ e M filas $nrtPS$, onde M representa o número de SSs cobertas na célula. O valor é dito ser máximo, pois a fila somente existe na BS se o serviço associado estiver ativo na SS correspondente. Além disso, uma única fila recebe todos os fluxos classificados como BE. Desse modo, fluxos de serviço associados a uma mesma classe e destinado a uma determinada SS recebem o mesmo encaminhamento. Não há prioridade entre as filas de uma mesma classe, somente entre os serviços. Por fim, deve-se ressaltar que as filas nesse modelo possuem capacidade

infinita.

Para validar a arquitetura de QoS, especialmente o algoritmo de escalonamento, também foi desenvolvido um modelo de simulação com as principais funcionalidades de redes WiMAX, referenciado como “Modelo Zero”. Esse *framework* foi concebido com o objetivo de prover um módulo WiMAX para o ambiente de simulação NS-2 (*Network Simulator - Versão 2*), simulador de eventos discretos direcionado para pesquisas em redes de computadores. O NS-2 é escrito em C++ e OTcl (linguagem de *script* TCL com extensão orientada a objeto desenvolvida no MIT - *Massachusetts Institute of Technology*), e por ser gratuito e ter seu código fonte disponível na Internet, é largamente utilizado no meio acadêmico e científico. O diagrama de classes simplificado para o modelo de simulação pode ser visto na Figura 12.

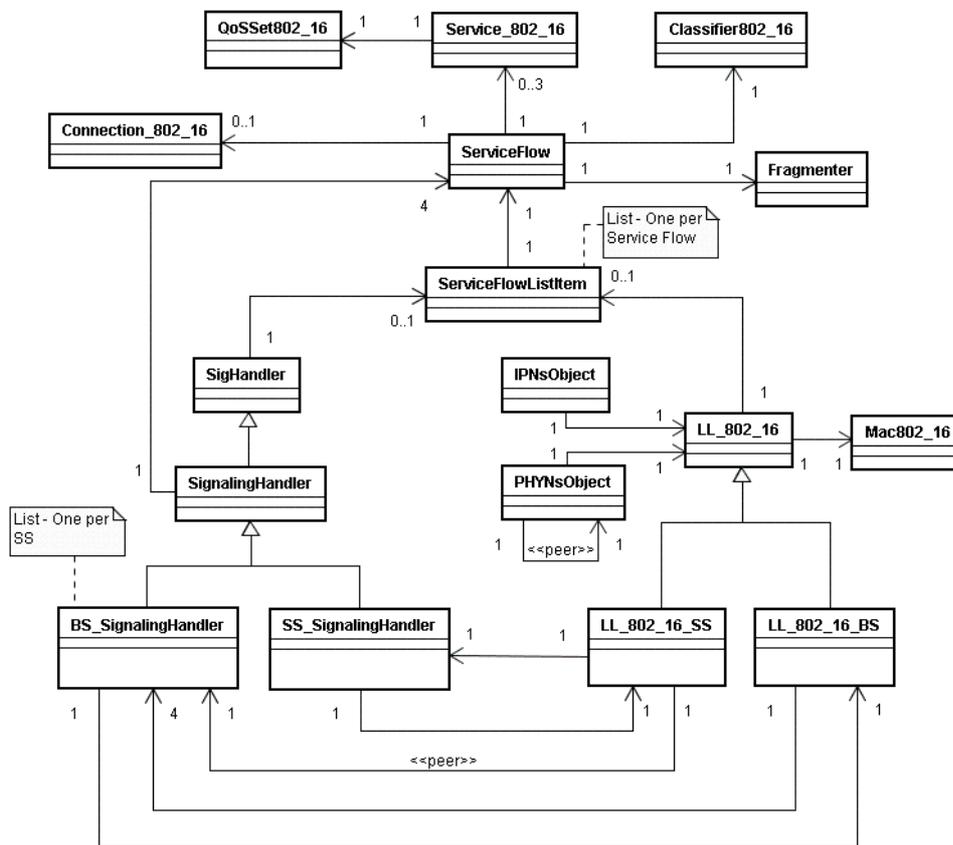


Figura 12: Diagrama de classes simplificador do modelo de simulação desenvolvido em [6].

Como defendido na seção 2.4, a configuração estática dos pesos e a ordem de atendimento das filas de serviço representam as principais fragilidades do algoritmo proposto por Malini [6]. Dessa maneira, considerando as discussões realizadas nos capítulos anteri-

ores, este trabalho propõe um novo mecanismo *fuzzy* para alterar dinamicamente os pesos atribuídos a cada classe conforme a mudança de perfil do tráfego na rede. Com isso, uma nova estratégia de escalonamento de recursos substitui o algoritmo que opera no módulo *BS Downlink Scheduler*. Esta proposta também é implementada em C++ no intuito de permitir sua inserção no modelo de simulação WiMAX. Assim, pode-se validar o trabalho e avaliar seu desempenho com respeito à satisfação das restrições de QoS associados aos fluxos de dados.

5.2 Estratégia Fuzzy para Alteração Dinâmica de Pesos

Observando as descrições das classes de serviço propostas pelo padrão IEEE 802.16, pode-se dividi-las em dois grandes grupos: no primeiro, UGS e rtPS agrupam aplicações em tempo real e no segundo, nrtPS e BE as aplicações em não tempo real. O primeiro grupo é reconhecido pela sensibilidade a atrasos e tolerância a perdas de pacotes, ao contrário do segundo que se caracteriza por ser tolerante a atrasos e sensível a perda de pacotes. Os mecanismos de QoS atendem essas e outras especificidades. Entretanto, para lograr êxito, o controle de QoS deve ser fim-a-fim, ou seja, deve ser aplicado em cada enlace ao longo do percurso entre as estações finais.

Nesse contexto, o escopo deste trabalho restringe-se ao canal DL entre a BS e as SSSs cobertas em uma célula IEEE 802.16. Baseado nas características dos serviços apresentados anteriormente, o objetivo é contribuir para o papel do nó WiMAX na tarefa de controle de QoS fim-a-fim, mitigando o atraso nas filas dos serviços UGS e rtPS e a perda de pacotes nas filas dos serviços nrtPS e BE. Para tanto, a estratégia proposta se baseia em lógica *fuzzy* para decidir dinamicamente qual a melhor configuração para os pesos $WEIGHT_{UGS}$, $WEIGHT_{rtPS}$, $WEIGHT_{nrtPS}$ e $WEIGHT_{BE}$, mencionados anteriormente, haja vista a quantidade de recursos disponíveis, a prioridade entre as classes de serviço e a situação das filas de atendimento. A Figura 13 representa a estratégia de atendimento das filas de serviço.

Ao contrário da alocação estática de [6], na proposta apresentada nesta dissertação a distribuição dos recursos entre os fluxos de dados é dinâmica e é realizada em dois níveis: no primeiro nível, representado pelo módulo Alocador *Fuzzy*, faz-se a alocação de recursos entre as classes de serviços e, no segundo nível, os módulos Escalonadores distribuem os recursos alocados entre as respectivas filas de atendimento. Os Escalonadores desempe-

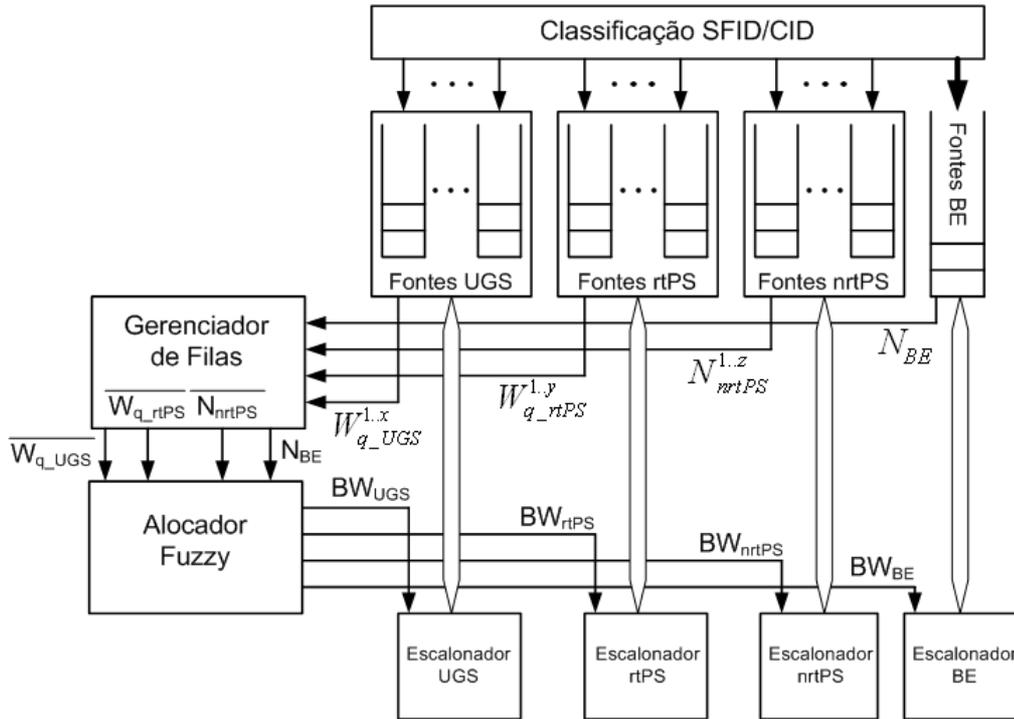


Figura 13: Esquema de distribuição de largura de banda entre as filas de serviço.

nham função similar ao algoritmo de Malini, ou seja, escalonamento de recursos. Portanto, a principal diferença entre as duas propostas reside na figura do alocador, tarefa realizada uma única vez no algoritmo de [6], no início da operação do sistema. Deve-se ressaltar que os escalonadores foram reescritos no intuito de contornar a situação de *starvation* verificada na seção 2.4. Além disso, tanto no algoritmo de Malini quanto neste trabalho, os pesos atribuídos são interpretados como percentuais de recursos alocados. Por fim, pode-se destacar, como mais um diferencial, o módulo dedicado ao gerenciamento das filas, referenciado como Gerenciador de Filas. Por meio de verificação periódica de parâmetros das filas, esse módulo determina quando o alocador deve ser acionado. Isso evita o processo de redistribuição de recursos a cada novo quadro MAC. Dito de outro modo, o alocador será acionado apenas quando um novo evento na rede for detectado. A identificação de um novo evento fica a cargo do mecanismo de histerese.

De uma forma simplificada, a estratégia trabalha da seguinte forma: os pacotes que chegam à BS, com destino às SSs, são classificados e inseridos nas respectivas filas. O gerenciador de filas verifica periodicamente os valores de ocupação e tempo médio de espera para decidir se é necessário uma nova realocação de recursos entre as classes. A decisão baseia-se em um mecanismo de histerese, para o qual são fixados limiares de parâmetros para cada classe. Os recursos alocados pelo sistema especialista *fuzzy* para

cada classe são distribuídos entre respectivas filas, segundo a disciplina estabelecida nos módulos escalonadores.

5.2.1 Classificação dos Pacotes e Filas de Serviço

Os pacotes que atravessam a camada MAC na BS são mapeados para a conexão correspondente segundo os parâmetros de classificação adotados. Esses parâmetros podem ser endereços IP de origem e destino, portas de origem e destino, campo *Type of Service* (ToS) do IP, *flags* do cabeçalho TCP entre outros. Por fim, o pacote já associado a um valor de CID e SFID pode então ser enviado à fila de DL apropriada. Nesse momento, cada pacote é marcado com um *timestamp* de chegada à fila.

O esquema de filas de atendimento do canal DL na BS segue o que foi proposto em [6] e descrito na seção 5.1. No entanto, neste trabalho, as filas possuem capacidade finita, medida em *bytes* e ajustada segundo o cenário de simulação desejado. Além disso, as filas são agrupadas segundo a classe de serviço, formando três grupos: UGS sources, rtPS sources, nrtPS sources. Como já explicado, os fluxos classificados como BE são todos enviados para uma única fila.

5.2.2 Gerenciamento de Filas

Para alcançar os objetivos estabelecidos, o tempo médio de espera nas filas UGS e rtPS e a ocupação nas filas nrtPS e BE são monitorados pelo módulo Gerenciador de Filas. Essas medidas são representadas respectivamente pelas variáveis W_{q-UGS}^i , W_{q-rtPS}^i , N_{nrtPS}^i , N_{BE} onde i indica a i -ésima fila em cada classe de serviço. A medição é feita em intervalos múltiplos do número de quadros MAC. O tamanho desse intervalo pode ser fixado segundo a dinâmica da rede, ou seja, em redes onde o perfil de tráfego varia constantemente é conveniente fixar um intervalo menor, caso contrário o intervalo entre medidas pode ser mais esparso.

O atraso de cada pacote na fila é computado no momento de seu envio por meio da subtração entre o tempo atual, momento em que está sendo enviado, e o seu *timestamp*, momento em que chegou à fila. Obtém-se o atraso médio dividindo-se a soma dos atrasos acumulados no intervalo entre medidas pelo número de pacotes enviados durante o intervalo. Além disso, para cada pacote inserido na fila incrementa-se um contador com o tamanho do pacote em bytes e na saída de um pacote esse contador é decrementado com o mesmo valor. Para obter a ocupação basta acessar o valor atual do contador de cada fila.

Finalmente, calcula-se para cada grupo de filas a média do parâmetro associado. Desta maneira, quatro novos parâmetros são obtidos:

- $\overline{W_{q_UGS}}$: média entre os tempos médios de espera nas filas do serviço UGS;
- $\overline{W_{q_rtPS}}$: média entre os tempos médios de espera nas filas do serviço rtPS;
- $\overline{N_{nrtPS}}$: média entre as ocupações nas filas do serviço nrtPS;
- N_{BE} : ocupação na fila do serviço BE.

O módulo adota um mecanismo de histerese para o monitoramento, semelhante ao implementado pelo algoritmo RMON (*Remote Network Monitoring*) em gerenciamento de redes TCP/IP [26]. Para a histerese, são configurados limiares de parâmetros que, quando ultrapassados, acionam procedimentos específicos para tratar o evento. O trabalho [27] também adota essa idéia em sua solução para mudar dinamicamente a quantidade de recursos alocados a cada conexão, limitando o número de transições entre alocações. Neste trabalho, podem ser definidos um ou dois limites para cada um dos quatro parâmetros: limiar superior ou limiar superior e inferior.

A partir dos limites é possível definir eventos que avisem sobre possíveis situações anormais. Por exemplo, a ultrapassagem do limiar superior pode indicar sub-alocação de recursos, enquanto a ultrapassagem do limiar inferior pode indicar sobre-alocação. Também é caracterizado um evento quando se verifica que ocorreu mudança de zero para diferente de zero, e vice e versa, em medidas consecutivas de um mesmo parâmetro. A medida de um parâmetro igual a zero indica que as filas da respectiva classe estão inativas, ou seja, não possuem pacotes aguardando atendimento, pelo menos não para o próximo quadro. Esse último caso é especialmente importante para melhorar a eficiência no uso dos recursos, haja vista que uma classe com filas inativas, não deve ser contemplada na distribuição dos recursos. Dadas essas explicações, podem ser enumerados quatro tipos de eventos:

1. Evento 1: Ultrapassagem de limiar superior.
2. Evento 2: Ultrapassagem de limiar inferior.
3. Evento 3: Classe ativa.
4. Evento 4: Classe inativa.

A Figura 14 apresenta graficamente os quatro tipos de eventos. Deve-se observar que o Evento 1 ocorre se qualquer um dos parâmetros ultrapassar seu respectivo limiar superior, enquanto para o Evento tipo 2, todos os parâmetros devem ser analisados conjuntamente, como será explicado em seguida.

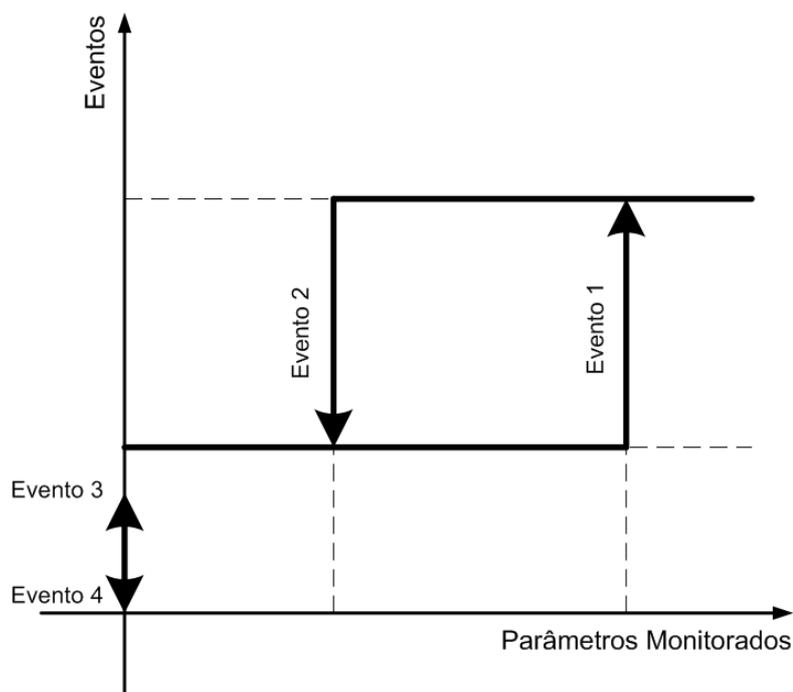


Figura 14: Mecanismo de histerese usado para identificar a ocorrência de um evento significativo que justifique uma redistribuição de recursos.

Um evento caracterizado por uma ultrapassagem de limiar inferior somente se verifica quando algumas medidas se encontram entre os seus limites inferior e superior e outras abaixo dos seus limites inferiores, indicando sobre-alocação. Se todas as medidas estiverem abaixo do limiar inferior, então há largura de banda excedente para todos. Por outro lado, ter todas as medidas dentro do intervalo indica que, pelo menos, a alocação atual é adequada aos requisitos de QoS para todos os serviços. Nesse caso, a operação *xor* é usada, pois sua tabela verdade traduz a situação descrita.

Caso ocorra algum evento, o módulo Alocador *Fuzzy* é acionado para redistribuir a largura de banda entre as classes de serviço. Esse mecanismo, associado ao intervalo entre medidas, evita que a operação de alocação *fuzzy* seja repetida a cada quadro, diminuindo a carga de operações. Assim, se o perfil de tráfego na rede não mudar ao longo do tempo, o único processamento realizado é o escalonamento FIFO (*First In First Out*) no nível de filas. Os módulos escalonadores são discutidos em seguida. As Figuras 15 e 16 traduzem num algoritmo o funcionamento da idéia apresentada para o caso de dois limites.

```

ALGORITMO gerenciaFilas

double supUGS = s1;           double supBE = s4;           bool nrtPSto0;
double infUGS = i1;           double infBE = i4;           bool nrtPSfrom0;
double supRTPS = s2;          bool UGSto0;                 bool BEto0;
double infRTPS = i2;          bool UGSfrom0;               bool BEfrom0;
double supNRTPS = s3;         bool rtPSto0;                bool evento = false;
double infNRTPS = i3;         bool rtPSfrom0;

//Evento caracterizado por uma ultrapassagem de limiar superior.
if (( $\overline{W}_{q\_UGS} > s1$ ) or ( $\overline{W}_{q\_nPS} > s2$ ) or ( $\overline{N}_{nPS} > s3$ ) or ( $N_{BE} > s4$ ))
    return (evento = true);

//Evento caracterizado por uma ultrapassagem de limiar inferior.
if (( $\overline{W}_{q\_UGS} < i1$ ) xor ( $\overline{W}_{q\_nPS} < i2$ ) xor ( $\overline{N}_{nPS} < i3$ ) xor ( $N_{BE} < i4$ ))
    return (evento = true);

//Evento de classe inativa.
if ((UGSto0) or (rtPSto0) or (nrtPSto0) or (BEto0))
    return (evento = true);

//Evento de classe ativa.
if ((UGSfrom0) or (rtPSfrom0) or (nrtPSfrom0) or (BEfrom0))
    return (evento = true);

return evento;

```

Figura 15: Algoritmo responsável por determinar a ocorrência de um evento.

```

ALGORITMO Alocação

unsigned im = intervaloMedida;
bool evento = false;

//accQuadros representa o número total de quadros MAC transmitidos até o atual momento.
if ( (accQuadros mod im) == 0 ) {
    calculaAtraso;
    calculaOcupação;

    evento = gerenciaFilas ( $\overline{W}_{q\_UGS}$ ,  $\overline{W}_{q\_nPS}$ ,  $\overline{N}_{nPS}$ ,  $N_{BE}$ );

    if (evento == true)
        alocadorFuzzy ( $\overline{W}_{q\_UGS}$ ,  $\overline{W}_{q\_nPS}$ ,  $\overline{N}_{nPS}$ ,  $N_{BE}$ );
}

```

Figura 16: Algoritmo responsável por acionar o alocador *fuzzy* caso seja detectado algum evento.

5.2.3 Alocador *Fuzzy*

O módulo Alocador *Fuzzy* é um tipo de sistema especialista *fuzzy* como definido em [20]. Seus espaços de entrada e saída são formados, cada um, por quatro variáveis lingüísticas: \overline{W}_{q_UGS} , \overline{W}_{q_rtPS} , \overline{N}_{nrtPS} , N_{BE} para o espaço de entrada e BW_{UGS} , BW_{rtPS} , BW_{nrtPS} , BW_{BE} para o espaço de saída. As variáveis de entrada, descritas na seção anterior, são passadas ao módulo por ocasião do acionamento do alocador. As variáveis de saída representam o percentual da largura de banda total disponível reservada para cada classe de serviço, sendo, portanto, os objetos de decisão da estratégia de distribuição de recursos. Elas correspondem aos pesos atribuídos estaticamente a cada classe no algoritmo apresentado em [6].

A partição *fuzzy* de cada uma das variáveis mencionadas e as configurações das respectivas funções de pertinência são definidas de modo a adequar o sistema *fuzzy* às características da rede onde o sistema será implantado. Neste trabalho, foram utilizadas funções triangulares e trapezoidais que têm a vantagem de serem implementadas de forma simples e eficiente, em geral sem prejuízo à qualidade dos resultados que justifique o uso de funções mais complexas [28,29]. O conjunto de termos lingüísticos das variáveis é formado pelos valores “low”, “medium” e “high”. Esse é o número mínimo de termos necessários para modelar o problema e conduz a um número mínimo de regras para a decisão. Foi seguida a estratégia usual de modelagem *fuzzy*, ou seja, começando pela estrutura mais simples. Modelos mais elaborados, com maior número de termos e funções mais complexas, podem ser necessários, caso se verifique a necessidade de refinar o controle sobre o sistema.

É importante ressaltar que as configurações específicas de cada variável de entrada, refletidas nas formas das respectivas funções de pertinência, visam traduzir as exigências em termos de atraso e ocupação em fila que são peculiares a cada grupo de aplicações. Por outro lado, o princípio norteador para a configuração das variáveis de saída é, exclusivamente, a prioridade entre as classes de serviço. Os critérios para configuração de cada variável adotados neste trabalho são apresentados durante esta seção.

Para a modelagem das funções de pertinência relativas às variáveis \overline{W}_{q_UGS} e \overline{W}_{q_rtPS} teve-se como base os requisitos de atraso apresentados em [21] e a recomendação ITU-T G.114 [30]. Os dois trabalhos definem faixas semelhantes de atraso para caracterizar níveis de serviço. Entretanto, os valores apresentados nos documentos se referem ao atraso total, ou seja, a soma de todos os tipos de atrasos ao longo do trajeto entre as estações finais. Por isso, foi adotado o uso de frações desses intervalos, para caracterizar apenas o

atraso nas filas da BS. A exemplo das partições *fuzzy*, esses intervalos são parametrizados, sendo ajustados segundo a definição dos níveis de serviço admitidos na rede. As funções de pertinência associadas aos termos correspondentes são apresentadas na Figura 17.

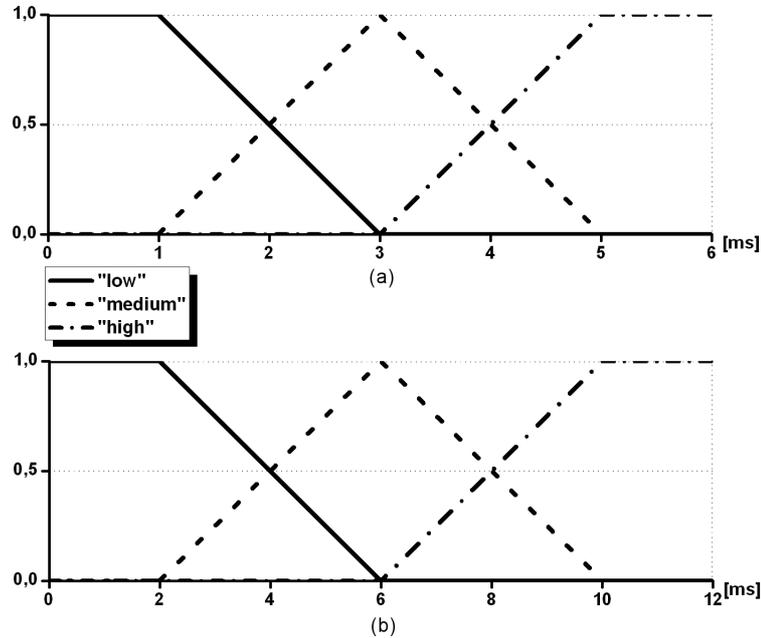


Figura 17: Partição *fuzzy* para as variáveis de entrada: (a) \overline{W}_{q-UGS} , (b) \overline{W}_{q-rtPS}

As funções de pertinência das variáveis lingüísticas \overline{N}_{nrtPS} e N_{BE} são apresentadas na Figura 18. Os intervalos relativos a essas variáveis foram definidos com base no algoritmo RED (*Random Early Detection*) [4, 21], para o qual são definidos níveis de ocupação de fila que determinam a probabilidade de descarte de pacotes. Nesta solução, os intervalos indicam o grau de necessidade de recursos para evitar o descarte de pacotes.

As variáveis que formam o espaço de saída, BW_{UGS} , BW_{rtPS} , BW_{nrtPS} e BW_{BE} , representam o percentual da largura de banda total disponível para DL alocada para cada classe de serviço. As funções de pertinência do espaço de saída são ilustradas na Figura 19. É importante frisar que podem ser estabelecidos outros critérios para configuração das funções de pertinência, seja do espaço de entrada seja do espaço de saída. A configuração deve ser tal que o sistema *fuzzy* seja capaz de refletir o conhecimento especialista sobre o ambiente específico onde se deseja operar.

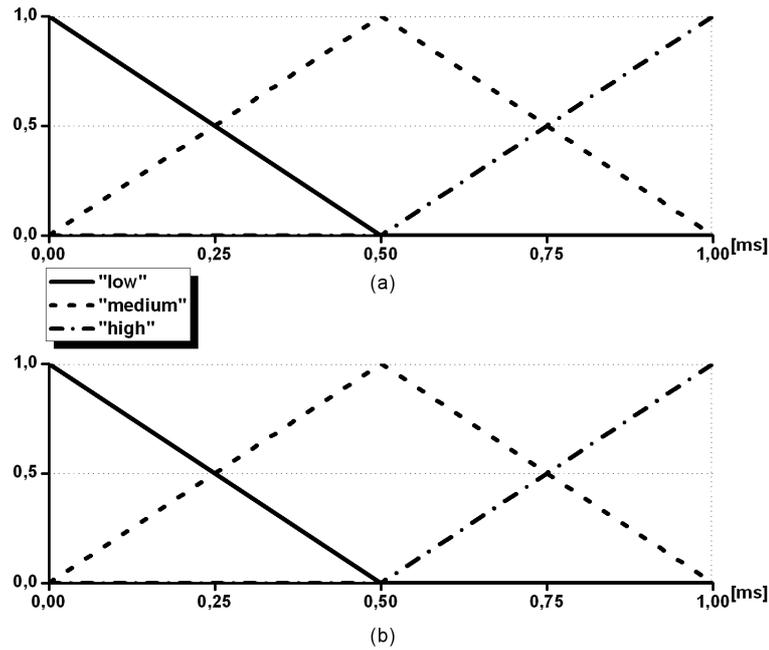


Figura 18: Partições *fuzzy* para as variáveis de entrada: (a) $\overline{N_{nrtPS}}$, (b) N_{BE}

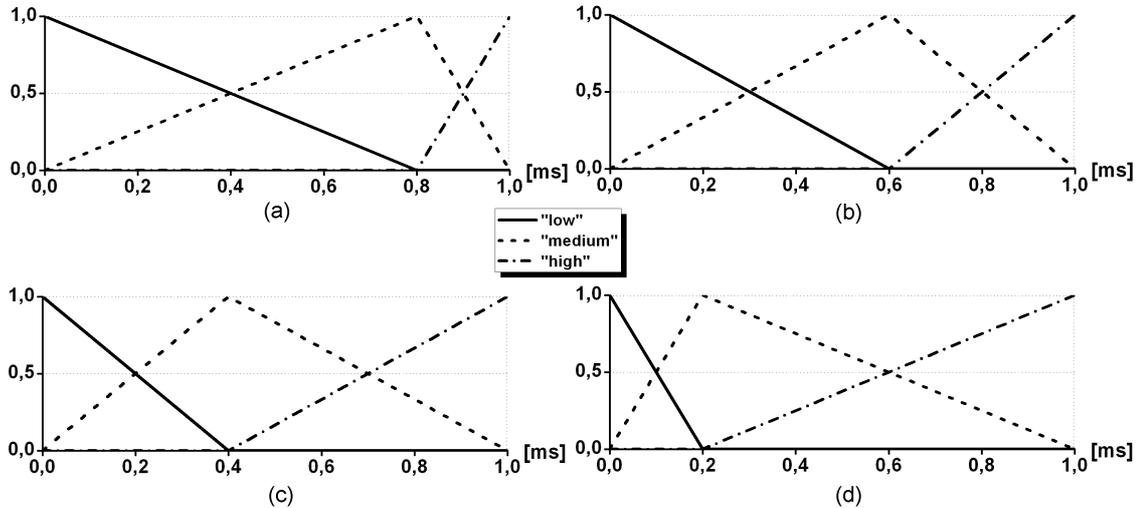


Figura 19: Partição *fuzzy* para as variáveis de saída (a) BW_{UGS} , (b) BW_{rtPS} , (c) BW_{nrtPS} , (d) BW_{BE} .

Para mapear todas as possíveis combinações na entrada, a base de regras possui 81 entradas, de acordo com a Equação 3.7. Em princípio, as regras foram construídas de forma empírica e utilizou-se como princípios norteadores a prioridade entre as classes, a situação das filas de atendimento e quantidade de recursos disponíveis. Com base nesses princípios foi elaborada uma primeira versão da base de regras. A partir dessa primeira versão, foram feitos ajustes subsequentes, buscando uma sintonização mais fina das regras.

Esse ajuste foi feito apresentando medidas conhecidas ao sistema especialista e observando as alterações nos valores de saída. Tendo sempre em mente o valor desejado para saída, ou seja, o valor de alocação para cada classe de acordo com o estado das variáveis de entrada, as regras foram sendo alteradas de forma a atender o valor esperado. Esse processo foi repetido diversas vezes, até se chegar à versão final da base de regras, apresentada no Apêndice B. A Tabela 1 traz algumas regras definidas para este trabalho.

Tabela 1: Algumas linhas da base de regras *fuzzy*.

$\overline{W_{q_UGS}}$	$\overline{W_{q_rtPS}}$	$\overline{N_{nrtPS}}$	N_{BE}	BW_{UGS}	BW_{rtPS}	BW_{nrtPS}	BW_{BE}
low	low	low	low	low	low	low	low
low	low	low	medium	low	low	low	medium
...
high	high	high	medium	high	high	medium	low
high	high	high	high	high	high	medium	low

O esquema de inferência adotada pela arquitetura é o Mamdani, tal como apresentado na seção 3.3. Sendo assim, a primeira regra, por exemplo, pode ser interpretada da seguinte forma: se ($\overline{W_{q_UGS}}$ é “low”) e ($\overline{W_{q_rtPS}}$ é “low”) e ($\overline{N_{nrtPS}}$ é “low”) e (BW_{BE} é “low”) Então (BW_{UGS} é “low”) e (BW_{rtPS} é “low”) e (BW_{nrtPS} é “low”) e (BW_{BE} é “low”).

Deve-se observar que os serviços UGS e rtPS recebem maior largura de banda mesmo com situação de tráfego semelhante aos serviços nrtPS e BE. Isso ocorre nesses exemplos de regras e em outras com situação semelhante, pois é levada em consideração a prioridade entre as classes de serviço também na base de regras. O encadeamento entre as regras é feito pela operação “or”. Finalmente, para a defuzzificação é usado o método COG (*Center Of Gravity*), cujo valor de saída é o centro de gravidade da função obtida após o processo de inferência, conforme explicado na seção na seção 3.3.4.

Após a defuzzificação, a soma dos valores percentuais de cada alocação não é necessariamente igual a 1. Além disso, pode ocorrer, em algumas situações específicas, que banda seja alocada para serviços em que não há fluxo de dados, ou seja, não há pacotes na fila aguardando atendimento, configurando desperdício de recursos. Esses casos ocorrem, pois o processo de defuzzificação adotado, e outros testados para o trabalho, nunca alcançam os extremos de decisão, ou seja, não possibilitam que valores de 0% ou 100% de alocação sejam apresentados como respostas. No caso da técnica COG, a explicação é

intuitiva, uma vez que o centro de gravidade de um objeto plano, como são os conjuntos *fuzzy* obtidos após o processo de inferência, nunca se situa em uma de suas extremidades. Desse modo, o valor escalar calculado, abscissa do centro de gravidade, não alcança nem 0 nem 1.

Para solucionar essas questões, o sistema especialista fuzzy foi complementado com um bloco de pós-processamento. Nesse bloco é feita a normalização dos valores de alocação de modo que a soma dos recursos seja exatamente 100% da largura de banda disponível. Adicionalmente, os recursos alocados para serviços, cujas medidas associadas sejam 0, são redistribuídos igualmente entre os serviços com valores diferente de 0, ou seja, entre aqueles que realmente necessitam.

5.2.4 Escalonadores

Uma vez alocada banda para as classes de serviço, os escalonadores se encarregam de distribuí-la entre as respectivas filas. Em cada classe, o algoritmo atua circularmente e a primeira fila a ser atendida no quadro atual é a última que foi atendida no quadro anterior. Isso evita o problema de *starvation* verificado no algoritmo discutido na seção 2.4. Formalmente, qualquer disciplina de escalonamento pode ser empregada. Entretanto, o problema se concentra em como distribuir recursos para aplicações com diferentes requisitos de qualidade e essa questão é resolvida com a alocação *fuzzy*, no nível de classes de serviço. Por esse motivo, não há razão tão forte que justifique uma disciplina de escalonamento mais complexa que a FIFO, no nível das filas. Adicionalmente, um possível critério que pode ser adotado na priorização dentro de uma mesma classe é o endereço IP de destino com o uso de políticas como WRR e WFQ.

6 *Resultados Numéricos*

“A única possibilidade de descobrir os limites do possível é aventurar-se um pouco mais além deles, até o impossível.”

Arthur C. Clarke

A PÓS a implementação da estratégia e sua inserção no Modelo Zero, foram desenvolvidos alguns cenários de operação visando validá-la e avaliar seu desempenho com respeito ao atendimento dos requisitos de QoS inerentes a cada classe de serviço. São quatro cenários que buscam aprofundar o entendimento sobre a estratégia apresentada, comprovar sua efetividade na tarefa de distribuir recursos dinamicamente, medir desempenho e apontar desafios e trabalhos futuro. O detalhamento do ambiente de simulação, feito em seguida, somente não se aplica ao primeiro cenário, haja vista que, nesse caso, foram realizados testes com o módulo *Fuzzy Allocator* isoladamente.

6.1 Descrição do Ambiente de Simulação

Este trabalho se concentra no estudo de redes WiMAX PMP em que toda comunicação é realizada por meio da BS. A duplexação dos canais DL e UL baseia-se na técnica TDD. Portanto, o canal aéreo é segmentado no tempo em quadros de tamanho fixo, sendo cada quadro dividido em dois sub-quadros: um para o sentido DL e outro para o UL. Dado que a estratégia proposta é destinada ao escalonamento de recursos em DL, apenas o tráfego nesse sentido foi considerado.

Seguindo o que foi definido no Modelo Zero, toda SS recebe um endereço IPv4 (*Internet Protocol - versão 4*) fixo e cada aplicação simulada é identificada por um número de porta específico. Os cenários de testes são construídos a partir de um número de conexões entre BS e SSs, cada uma criada para transportar dados gerados por uma dada fonte. As opções de aplicações para serem associadas às conexões são as mesmas adotadas nos testes em [6], uma vez que elas são suficientes para representar plenamente as classes IEEE 802.16. As associações entre aplicação/protocolo, porta e classe de serviço são resumidas na Tabela 2.

Tabela 2: Relação entre aplicação, porta e classe IEEE 802.16.

<i>Aplicações adotadas nas simulações</i>			
Aplicação Simulada	Protocolo	Porta	Classe de Serviço IEEE 802.16
MPEG Streaming Audio/Video	RTSP*	554	rtPS
Navegação WEB	HTTP	80	BE
VoIP	SIP*	5060	UGS
Transferência de Arquivo	FTP	21	nrtPS
Terminal Remoto	Telnet	23	BE

*Os protocolos *Real Time Streaming Protocol* (RTSP) e *Session Initiation Protocol* (SIP) realizam o estabelecimento e controle do fluxo de dados e operam em conjunto com o *Real-time Transport Protocol* (RTP), que transporta os dados.

Como apresentado anteriormente, o esquema de filas especifica, para cada SS, uma fila com prioridade na BS para cada classe de serviço, com exceção do serviço BE. Nesse caso, todos os tráfegos BE, destinados a todas as SSs, são encaminhados para uma única fila FIFO. Para reproduzir o conceito de fluxo de serviço, previsto no padrão IEEE 802.16 e descrito na seção 4.3, o Modelo Zero define a classe denominada *ServiceFlow*. A BS possui uma lista de objetos *ServiceFlow*, cada um deles representando um fluxo de comunicação entre a BS e a SS. Toda conexão criada está associada a um único objeto *ServiceFlow*, que, por sua vez, está associado a uma classe de serviço e a uma única fila de atendimento.

Para exemplificar, o mapeamento das conexões de aplicação descritas na Tabela 3 para *ServiceFlows* do modelo de simulação dá origem, na BS, ao esquema de filas mostrado na Figura 20.

Tabela 3: Exemplo para ilustrar o mapeamento de conexões em objetos *ServiceFlows*.

Conexões a serem mapeadas em <i>ServiceFlows</i>				
Origem	Destino	Aplicação	IPv4 do destino	Classe de Serviço
BS	SS1	VoIP	10.0.0.11	UGS
BS	SS2	Transferência de Arquivo	10.0.0.12	nrtPS
BS	SS3	Navegação WEB	10.0.0.13	BE
BS	SS4	Navegação WEB	10.0.0.14	BE

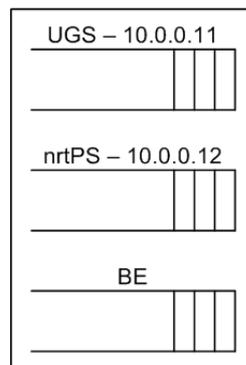


Figura 20: Filas no canal DL presentes na BS.

Um dos pontos de destaque presente no Modelo Zero é que, desde o princípio, sua elaboração buscou refletir as características de equipamentos reais, desenvolvidos especificamente para o padrão IEEE 802.16. Com isso, espera-se obter resultados bastante próximos da realidade. Para satisfazer esse objetivo, foram atribuídos aos parâmetros do modelo valores reais de operação, compatíveis com aqueles suportados pela primeira geração de equipamentos IEEE 802.16 lançada no mercado [6]. Por essa mesma razão, os valores desses parâmetros, apresentados em seguida, são repetidos nos testes realizados neste trabalho.

Cada quadro MAC tem duração $frameDuration = 10\text{ ms}$, que é um dos valores permitidos no padrão IEEE 802.16 [5] para a camada física baseada em OFDM. O número de quadros por segundo pode ser obtido facilmente pela expressão $(1/frameDuration)$. Assim, $FRAMES/SECOND = 100$. Um símbolo, unidade de transmissão de dados da camada física, tem duração $symbolDuration = 43.31\ \mu\text{s}$, sendo que o modelo suporta

sete configurações para *symbolSize*: 12, 36, 48, 72, 96 ou 108 *bytes*/símbolo. Cada configuração corresponde a um tipo de modulação suportado pelo padrão (BPSK, QPSK, QAM 16 ou QAM 64) combinado com um canal de tamanho 1,75 MHz ou 3,5 MHz. Nas simulações realizadas neste trabalho foram adotados *symbolSize* = 48 em um canal de 3,5 MHz. Todos os valores adotados são suportados pelo padrão IEEE 802.16.

A partir desses parâmetros, pode-se desenvolver a formulação para o cálculo da largura de banda total (DL+UL) disponível para simulação.

$$\frac{frameDuration}{symbolDuration} = SYMBOLS/FRAME \quad (6.1)$$

$$SYMBOLS/FRAME \times FRAMES/SECOND = SYMBOLS/SECOND \quad (6.2)$$

$$SYMBOLS/SECOND \times symbolSize = BYTES/SECOND \quad (6.3)$$

$$BYTES/SECOND \times 8 = bits/s \quad (6.4)$$

Seguindo a seqüência de equações e usando os valores de parâmetros adotados, obtém-se uma largura de banda de 8,46 Mbps. É importante ressaltar que, na prática, essa é a banda total disponível, tanto para o canal de DL quanto UL, seja para tráfego de dados (carga efetiva) e informações de controle, seja para uso em intervalos de contenção e de *ranging*. Neste trabalho, entretanto, o valor obtido é considerado integralmente como banda disponível para tráfego efetivo, uma vez que a versão atual do modelo de simulação ainda não reproduz todas as funcionalidades mencionadas que envolvem, de alguma forma, consumo de recursos. Outra funcionalidade ausente no ambiente de simulação e que está fora do escopo de implementação deste trabalho é o mecanismo de controle dinâmico da duração dos sub-quadros DL e UL, previsto no padrão IEEE 802.16 e explicado na seção 4.2. Nas simulações realizadas, portanto, a largura de banda total é dividida igualmente entre os dois sentidos de comunicação, conferindo ao canal DL uma banda de 4,23 Mbps.

O Modelo Zero contempla vários perfis de tráfego, permitindo emular diversos tipos de fontes de dados. Especificamente, nos cenários em que se aplica, são gerados tráfegos UGS e rtPS periódicos, tráfego nrtPS com periodicidade variável e tráfego BE pseudo-aleatório.

O tamanho dos pacotes é variável, exceto para tráfego UGS. Em cada cenário é configurada uma composição de tráfego diferente, no intuito de avaliar o comportamento da estratégia de alocação em diversas situações de demanda. O detalhamento da composição de tráfego para cada cenário é apresentada no Apêndice D.

Em todos os cenários, o intervalo de simulação é de 60 s e, considerando $frameDuration = 10\text{ ms}$, em cada simulação são gerados 6000 quadros MAC. O intervalo entre medidas usado no monitoramento das filas de serviço, exposto na seção 5.2.2, é de 10 quadros, o que resulta em 600 medidas por simulação. Em relação ao mecanismo de histerese, a Tabela 4 resume os valores fixados para os testes. Os resultados das simulações são apresentados graficamente. A abscissa indica o instante de tempo da simulação e a ordenada, o valor da respectiva medida naquele instante.

Tabela 4: Valores de limiares configurados para o mecanismo de histerese.

Limiares de histerese	
Limiar	Valor
upper_UGS	3,00 ms
lower_UGS	1,00 ms
upper_rtPS	6,00 ms
lower_rtPS	3,00 ms
upper_nrtPS	40%
lower_nrtPS	20%
upper_BE	60%
lower_BE	30%

Finalmente, para reduzir a complexidade do modelo de simulação, foram assumidas as seguintes suposições em sua implementação: os atrasos de propagação são ignorados; o canal é livre de erros (cada pacote transmitido é recebido corretamente em seu destino); não há ocorrência de colisões.

6.2 Cenário 1

O Cenário 1 tem por finalidade desenvolver sensibilidade sobre o funcionamento do módulo *Fuzzy Allocator* ao apresentar, para determinadas entradas, os valores dos pesos atribuídos a cada classe, como resultado do processo de inferência. Nesse caso, os testes são realizados sem incorporar o alocador ao modelo de simulação. Entradas específicas são passadas manualmente para o sistema especialista e as respostas obtidas são relacionadas às respectivas entradas, conforme mostrado na Tabela 5. Esses testes são úteis,

pois permitem antecipar como o alocador *fuzzy* se comporta em operação contínua, onde parâmetros são medidos e apresentados sucessivamente para tomada de decisão. Cabe observar que as configurações do sistema especialista (partições *fuzzy*, funções de pertinência e base de regras) seguem àquelas especificadas na seção 5.2.3.

Tabela 5: Resultados do processo de inferência para algumas entradas específicas, apresentadas manualmente ao controlador.

	\overline{W}_{q_UGS}	\overline{W}_{q_rtPS}	\overline{N}_{nrtPS}	N_{BE}	BW_{UGS}	BW_{rtPS}	BW_{nrtPS}	BW_{BE}
1	1,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00
2	0,00	0,00	0,00	0,10	0,00	0,00	0,00	1,00
3	5,00	0,00	0,00	1,00	0,56	0,00	0,00	0,44
4	5,00	0,00	1,00	0,00	0,54	0,00	0,46	0,00
5	5,00	10,00	0,00	0,00	0,52	0,48	0,00	0,00
6	0,00	0,00	1,00	1,00	0,00	0,00	0,52	0,48
7	0,00	10,00	1,00	0,00	0,00	0,52	0,48	0,00
8	3,00	3,00	0,00	0,00	0,77	0,23	0,00	0,00
9	0,00	0,00	0,50	0,50	0,00	0,00	0,54	0,46
10	5,00	10,00	1,00	1,00	0,40	0,37	0,20	0,03

As linhas 1 e 2 exemplificam a situação em que apenas uma das quatro classes de serviço tem seu parâmetro associado diferente de zero (na linha 1, filas UGS apresentam 1 ms de atraso e na linha 2, a fila BE possui 10 % de seu espaço ocupado), indicando que, entre as filas associadas à respectiva classe, existe pelo menos uma com dados a serem transmitidos. Nesses casos, os recursos são alocados integralmente apenas à classe que possui filas ativas, independente de sua necessidade. De forma geral, são alocados recursos apenas para serviços com filas ativas, ou seja, filas contendo pacotes aguardando atendimento.

As linhas 3-7 apresentam casos ilustrativos da influência da prioridade entre as classes na tarefa de alocação. Todos os valores de parâmetros foram escolhidos para serem mapeados para o termo *fuzzy* “*high*”, com grau de pertinência 1. Com a mesma interpretação para todas as variáveis lingüísticas, a diferença nas alocações se deve exclusivamente à prioridade estabelecida nas funções de pertinência de saída. Com isso, pode-se observar na tabela que, quanto maior a prioridade relativa entre duas classes (caso extremo na linha 3), maior é a diferença no valor de alocação. Esse valor cai gradualmente conforme a prioridade relativa diminui.

As linhas 8 e 9 mostram a situação inversa àquela ilustrada anteriormente, ou seja, fixam-se os mesmo valores para os parâmetros no intuito de observar como o mapeamento de um valor escalar nos respectivos conjuntos *fuzzy* interfere no processo de decisão. Sendo

assim, apesar de os valores em cada linha serem iguais, a tradução de cada parâmetro em seus termos lingüísticos é diferente. Na linha 8, por exemplo, 3 ms para \overline{W}_{q_UGS} é totalmente compatível com o conjunto “*medium*”, enquanto o mesmo valor para \overline{W}_{q_rtPS} é interpretado como “*low*”, com grau 0,75, e “*medium*”, com grau 0,25. Essa diferença de interpretação reflete as exigências de atraso inerentes a cada grupo de aplicações, conforme descrito na seção 5.2.3. Conseqüentemente, essa diferença de interpretação resulta em números de alocação diferentes, conforme pode ser visto na Tabela 5.

Finalmente, a linha 10 ilustra o caso em que os parâmetros de todas as classes são interpretados como “*high*”, com grau de pertinência 1. Naturalmente, as classes com maior prioridade recebem parte maior dos recursos e os valores de alocação decaem conforme a prioridade diminui. Entretanto, é interessante observar que, mesmo com classes mais prioritárias exigindo grande soma de recursos, as menos prioritárias não ficam sem um mínimo de recursos.

De acordo com o que foi exposto, o cenário 1 mostra que o módulo *Fuzzy Allocator* obedece às prioridades estabelecidas (linhas 3-7) e reproduz as características de cada classe de serviço no que se refere aos seus requisitos de QoS (linhas 8 e 9). Também é possível perceber como o ajuste das partições *fuzzy*, bem como das funções de pertinência envolvidas, influenciam na interpretação dos valores escalares e por conseqüência no processo de inferência.

6.3 Cenário 2

No Cenário 2 e nos subseqüentes, o módulo *Fuzzy Allocator*, bem como as demais funções que compõem a proposta apresentada, são incorporados ao Modelo Zero de simulação. O objetivo deste cenário, em específico, é permitir uma melhor compreensão de como se procede a alocação de banda quando duas ou mais classes, com prioridades diferentes, demandam recursos durante a simulação. Qualquer conjunto com duas ou mais classes poderia ser escolhido para esse exemplo. Entretanto, devido às prioridades de UGS e BE serem opostas, elas permitem uma melhor visualização do processo de decisão.

Para alcançar o objetivo definido, as funções de pertinência associadas às variáveis lingüísticas do espaço de saída são configuradas especialmente para esse cenário, de forma distinta àquela que foi apresentada na seção 5.2.3, como pode ser visto no Apêndice C. As funções de pertinência do espaço de saída são ajustadas de maneira a representar diferentes interpretações para a expressão “prioridade”. Para verificar como a configuração

dessas funções influencia na definição de prioridade, as funções relacionadas às variáveis de entrada são mantidas iguais àqueles mostradas na seção 5.2.3, em todos os testes. Assim como nos demais cenários, a base de regras adotada é a mesma mostrada no Apêndice B. Além disso, o mecanismo de histerese atua observando apenas o limiar superior.

São realizadas três simulações: na primeira, a prioridade de BE é maior que a de UGS, na segunda, as prioridades são iguais e na terceira, a prioridade de UGS é maior. Em todos os casos, a simulação começa com a largura de banda disponível dividida igualmente entre as classes de serviço. Espera-se com isso, em primeiro plano, mostrar a mudança no valor dos pesos atribuídos a cada classe à medida que se altera o perfil de tráfego na rede. Em paralelo, pretende-se verificar o comportamento do sistema de acordo com a ordem de prioridade estabelecida. Os demais parâmetros de simulação permanecem inalterados durante os três testes.

A rede construída, mostrada na Figura 21, é composta por duas SS e uma BS. Pacotes que partem de duas fontes de dados UGS na estação remota têm como destino SS_2 e SS_4 e pacotes de uma fonte BE chegam à SS_4 .

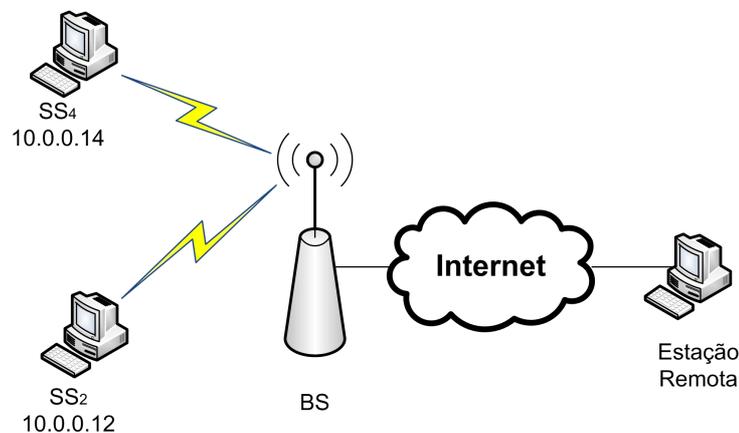


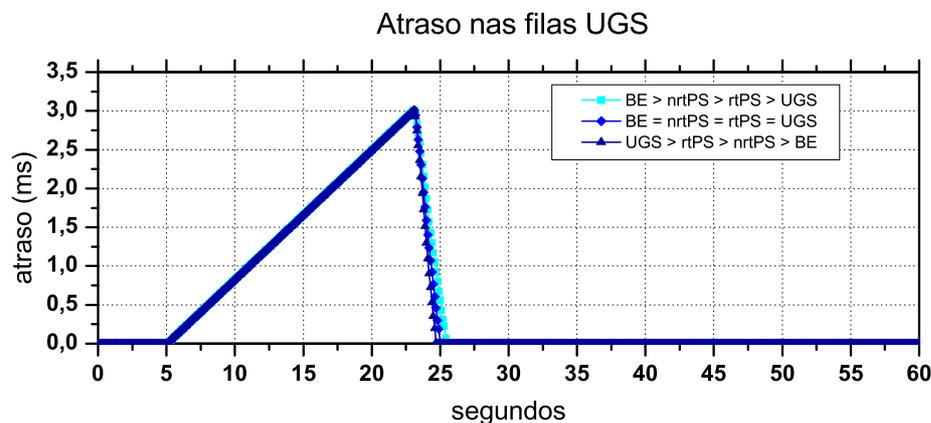
Figura 21: Rede para o cenário 2.

Os parâmetros de tráfego para este cenário são apresentados na Tabela 6. A última coluna apresenta a porcentagem da largura de banda disponível que cada classe demanda durante a simulação. Além disso, o intervalo de transmissão, mostrado na penúltima coluna, indica o início e fim, dentro dos 60 segundos de simulação, que cada classe demanda recursos. De acordo com a Tabela 6, as fontes transmitem dados simultaneamente por pelo menos 10 s. Esse período pode ser maior caso haja pacotes nas filas aguardando atendimento após o tráfego UGS cessar.

Tabela 6: Parâmetros de tráfego para o Cenário 2.

<i>Parâmetros de Tráfego</i>				
Classe	Aplicação	Destino	Início-Fim (s)	Demanda
UGS	VoIP	SS_2	5 – 30	10%
	VoIP	SS_2	5 – 30	20%
BE	Terminal Remoto	SS_4	20 – 40	60%

A Figura 22 mostra, para todas as simulações, que o tempo médio de espera na fila UGS cresce linearmente até o valor de 3,0 ms; comportamento esperado, já que é exatamente esse o limiar superior de histerese para a classe UGS. Esse valor é alcançado, precisamente, aos 23,2 segundos de simulação, instante em que é gerado um evento que aciona o alocador *fuzzy*, como pode ser visto na Figura 23. Após a redistribuição da largura de banda, o atraso na fila UGS cai até 0 ms. Observando as Figuras 22 e 23 conjuntamente, na simulação em que UGS possui maior prioridade (linha azul escura), a taxa com que o atraso decai é mais acentuada, concordando com a maior alocação (85% da largura de banda DL). À medida que a prioridade UGS diminui, diminuem também a porção de recursos alocados (linha azul indica 76% e linha azul clara 67%) e, conseqüentemente, a taxa com que o atraso cai.

**Figura 22:** Tempo médio de espera nas filas UGS.

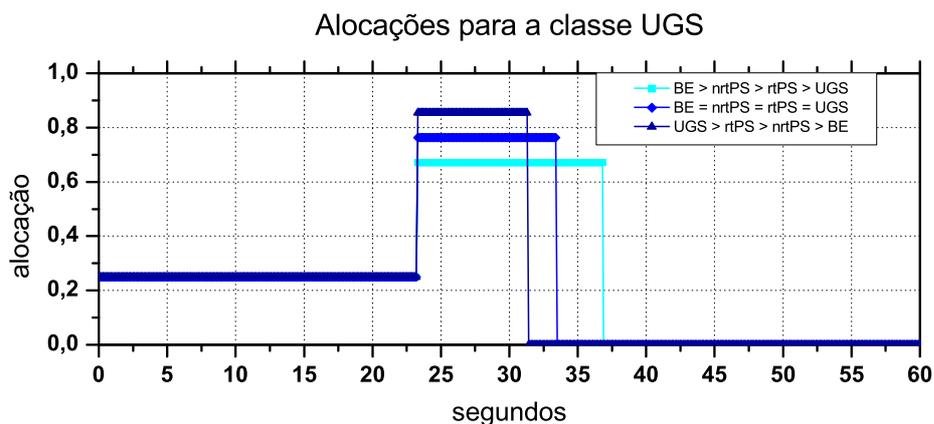


Figura 23: Alocações atribuídas à classe UGS.

Analisando a Figura 24, pode-se notar que o nível de ocupação na fila BE alcança o limiar superior em instantes diferentes, dependendo da configuração de prioridade. Isso ocorre, pois o alocador *fuzzy* é acionado anteriormente devido ao evento caracterizado pela ultrapassagem do limiar *Upper_UGS*. A partir desse instante, a simulação transcorre com alocações diferentes. Inclusive, em 23,2 s, momento desse primeiro acionamento, pode-se notar um ponto de quebra no gráfico de ocupação da fila BE. A partir desse ponto, a velocidade com que a ocupação da fila BE ocorre depende da quantidade de banda alocada a essa classe. Quando a prioridade de BE é maior, linha azul clara, a alocação para essa classe também é maior. Com mais recursos à disposição, o nível de ocupação na fila cresce em menor velocidade, alcançando o limiar superior mais tarde. O mesmo raciocínio pode ser estendido às outras duas configurações.

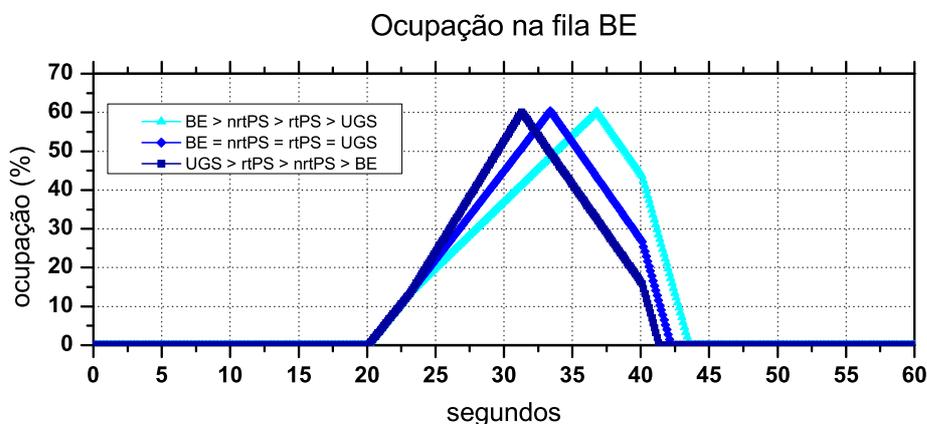


Figura 24: Ocupação na fila BE.

As Figuras 24 e 25 mostram que o segundo acionamento ocorre nos instantes 31,3 s, 33,4 s e 36,8 s: pontos referentes às linhas azul escuro, azul e azul claro, respectivamente. É importante destacar que o tráfego UGS cessou a partir do segundo 30, portanto, quando o alocador foi acionado novamente, toda a largura de banda foi destinada ao atendimento da fila BE, como pode ser verificado na Figura 25. Por essa razão, a taxa com que a ocupação cai é a mesma para as três simulações. Um outro ponto merece atenção: a partir de 40 s, a ocupação cai em maior velocidade, evidenciado pelo ponto de quebra nos gráficos. Isso ocorre, pois o tráfego BE cessa exatamente no quadro MAC 4000, correspondente ao segundo 40.

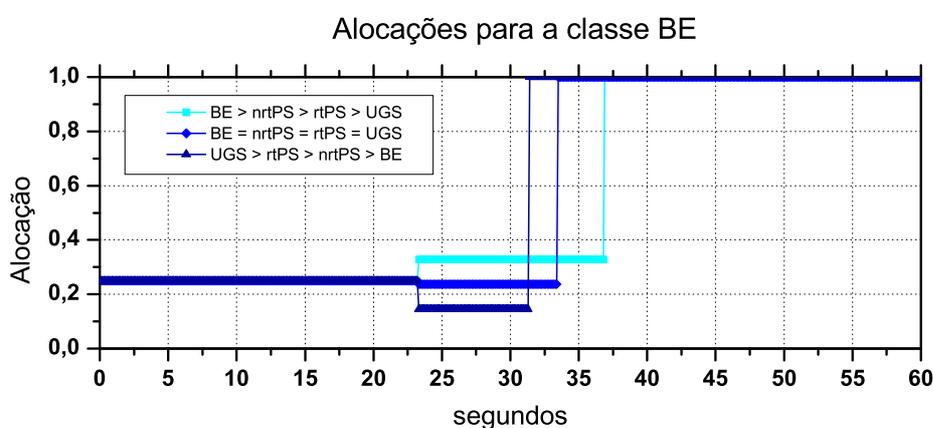


Figura 25: Alocações atribuídas à classe BE.

De acordo com os dados apresentados, pode-se afirmar que o sistema *fuzzy* proposto aloca mais recursos à classe com maior prioridade, tal como esperado. É importante notar também que o nível de prioridade atribuído a cada classe pode ser configurado não só na base de regras, mas também nas funções de pertinência. O termo nebuloso “prioridade” pode ser interpretado de infinitas maneiras, e essa subjetividade pode ser reproduzida nas funções de pertinência do espaço de saída, por exemplo. Em síntese, o ajuste pode ser feito conforme as especificidades do ambiente para o qual o sistema foi planejado.

6.4 Cenário 3

No cenário 3, a estratégia *fuzzy* é submetida, durante alguns instantes de tempo, a uma demanda superior à capacidade de transmissão. Dito de outro modo, a soma de todos os tráfegos gerados pelas fontes excede 4,23 Mbps. Essa situação pode ocorrer em uma operação real, caso seja adotada a abordagem estatística para garantia de QoS, ou

seja, apesar de, na média, o tráfego não ser maior que a soma de recursos, em alguns momentos, esse limite pode ser superado. Nesses casos, deve-se garantir atendimento às classes de maior prioridade, evitando o problema de *starvation* nas classes menos prioritárias. Portanto, o objetivo deste cenário é avaliar como a estratégia *fuzzy* proposta atua na situação de garantia estatística de QoS. A rede construída é apresentada na Figura 26 e a Tabela 7 traz a matriz de tráfego. A composição detalhada do tráfego pode ser consultada no Apêndice D.

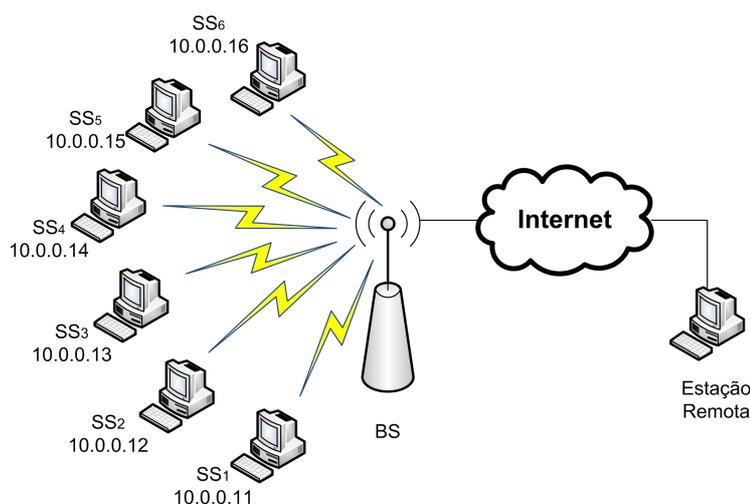


Figura 26: Rede para o cenário 3.

Tabela 7: Parâmetros de tráfego para o Cenário 3.

<i>Parâmetros de Tráfego</i>				
Classe	Aplicação	Destino	Início-Fim (s)	Demanda
UGS	VoIP	SS_2	10 – 40	25%
	VoIP	SS_4	20 – 40	15%
nrtPS	Transferência de arquivos	SS_2	0 – 60	11%
	Transferência de arquivos	SS_3	0 – 60	8%
	Transferência de arquivos	SS_5	0 – 60	11%
BE	HTTP	SS_1	0 – 60	11%
	HTTP	SS_4	0 – 60	19%
	HTTP	SS_6	0 – 60	10%

De acordo com a Tabela 7, durante toda a simulação, fontes nrtPS e BE consomem, em média, 70% da largura de banda disponível. No intervalo entre 10 e 40 s de simulação, uma fonte UGS demanda 25% dos recursos e, entre 20 e 40 s, outra fonte UGS necessita de outros 15%, totalizando uma demanda de 110% nesse último intervalo. Do mesmo modo que no cenário anterior, a simulação se inicia com os recursos distribuídos igualmente

entre as classes de serviço, ou seja, 25% da largura de banda para cada uma. Entretanto, nesse cenário, o mecanismo de histerese atua observando todas as categorias de eventos enumeradas na seção 5.2.2.

A Figura 27 apresenta os valores de alocação para cada classe durante toda a simulação. Observe que, antes mesmo de compor o primeiro quadro para transmissão, o módulo *Fuzzy Allocator* é acionado para rearranjar a distribuição de banda, haja vista que o gerenciamento de filas percebeu a inatividade em duas classes: UGS e rtPS. Nas medidas seguintes, a ocupação nas filas BE e nrtPS alternam, entre si, valores iguais a zero e diferente de zero. Quando uma das duas classes apresenta ocupação zero, a outra recebe a totalidade dos recursos, uma vez que também não há fluxos UGS e rtPS. O resultado perdura até o início do fluxo UGS, no segundo 10.

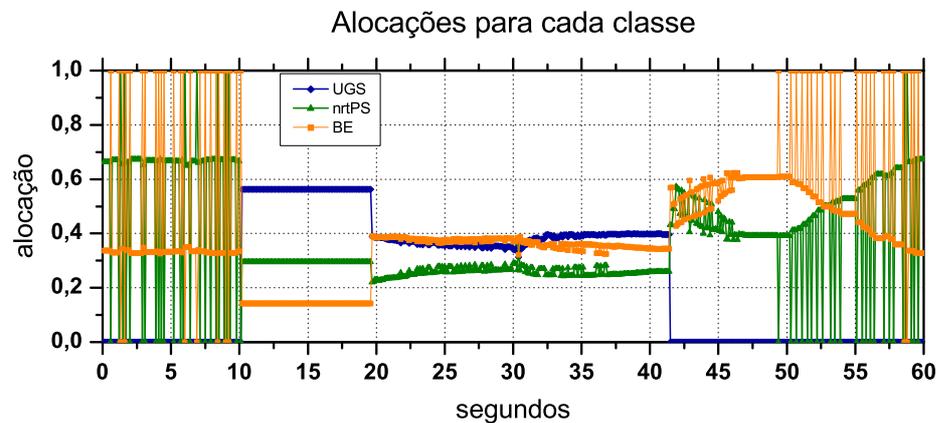


Figura 27: Pesos associados às classes durante a simulação.

Quando se inicia o tráfego UGS, um novo evento é detectado: filas UGS passam de uma média de ocupação 0% para diferente de 0%. Por isso, ocorre uma nova configuração de pesos: UGS recebe 56,3% dos recursos; nrtPS, 29,5%; BE, 14,2%, dados extraídos da Figura 27. Os recursos para nrtPS praticamente cobrem sua demanda de 30% e, por isso, quase não se percebem alterações nos seus valores de ocupação nesse momento, como observado na Figura 28. Já o parâmetro na fila BE, de acordo com a Figura 29, cresce de forma acentuada. Como resultado, a Figura 27 indica que um novo ajuste é feito no instante 19,7 s, devido à detecção de um evento de ultrapassagem de limiar inferior (BE se encontra entre os dois limites enquanto as demais classes estão abaixo do limiar inferior), resultando no seguinte arranjo: UGS possui 39%; nrtPS, 22%; BE 39%. Cabe ressaltar que BE recebe os mesmo recursos que UGS, mesmo tendo prioridade menor, fato motivado essencialmente pela avaliação mais crítica do parâmetro monitorado para BE frente aos das demais classes. Visualmente, de acordo com a Figura 29, a ocupação em

BE cresce em ritmo bem mais devagar após esse novo arranjo.

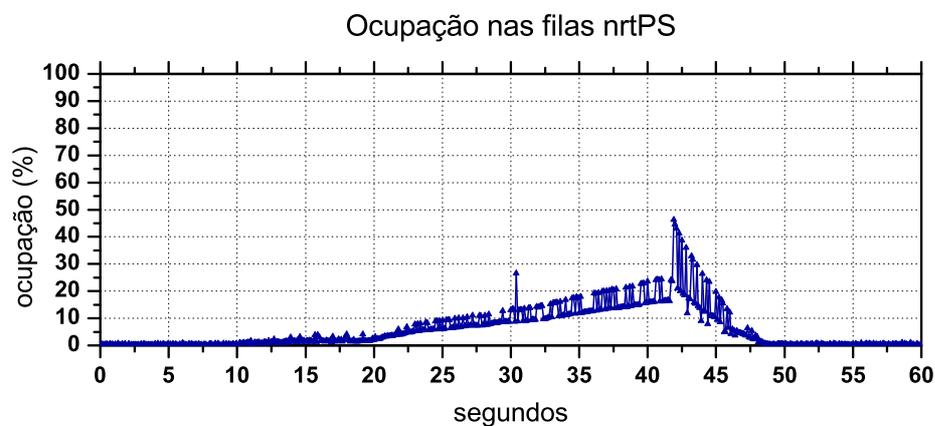


Figura 28: Ocupação média nas filas nrtPS.

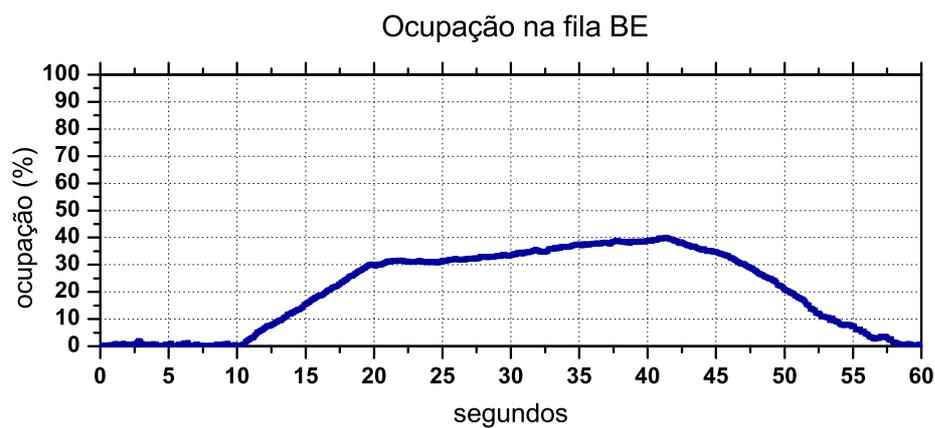


Figura 29: Ocupação média na fila BE.

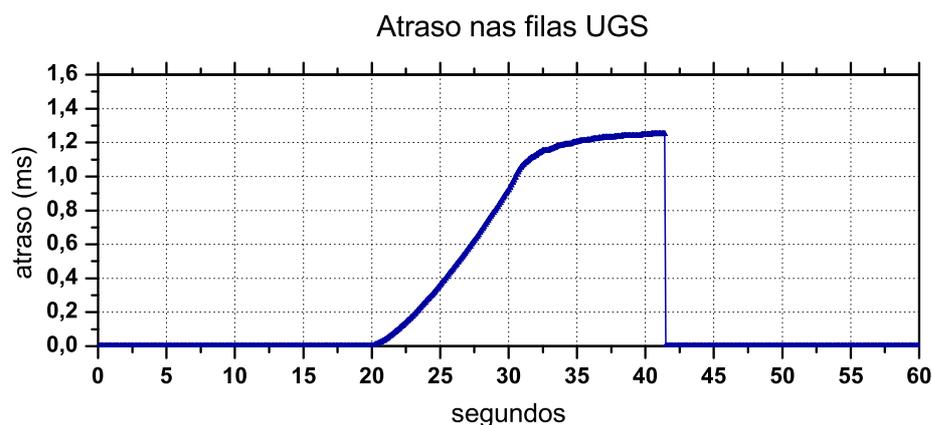


Figura 30: Tempo médio de espera nas filas UGS.

O valor de 39% seria suficiente para cobrir a demanda da fonte UGS para SS_2 (25%) se não tivesse início a segunda fonte, agora para SS_4 , com demanda de mais 15%. Por essa razão, de acordo com a Figura 30, o atraso médio nas filas UGS começa a crescer. Além disso, nas alocações seguintes, os recursos de UGS diminuem ainda mais, aumentando o ritmo de crescimento do atraso. Esse fato, por parecer incoerente a princípio, merece algumas explicações adicionais. Quando a segunda fonte UGS se inicia, pode-se extrair a seguinte fotografia das filas de atendimento, segundo as respectivas funções de pertinência apresentadas na seção 5.2.3: atraso UGS completamente compatível com o termo “*low*”; ocupação nrtPS, 0,964 “*low*” e 0,036 “*medium*”; ocupação BE, 0,398 “*low*” e 0,602 “*medium*”. Para esse cenário, no processo de inferência, são ativadas as seguintes linhas da base de regras mostrada no Apêndice B: 1, 2, 4, 5. Em todas as linhas, na correlação entre entrada e saída UGS, o termo *fuzzy* de saída é o mesmo: “*low*”. Enquanto para as outras duas classes, em algumas linhas, o termo é diferente de “*low*”. Como consequência, a classe UGS perde um pouco de seus recursos para as demais. Não perde mais devido à configuração das funções de pertinência de saída, em que a interpretação dos termos beneficia a classe UGS em relação às demais. Por exemplo, para BE (Figura 19, letra ‘d’), 20% dos recursos é considerado “*medium*” com grau de pertinência 1, já para UGS (Figura 19, letra ‘a’), a mesma interpretação, somente é feita com 80% dos recursos.

Essa situação muda quando o atendimento UGS alcança um atraso médio de 1 ms, por volta dos 30 s de simulação, como mostra a Figura 30. Observando a função de pertinência associada à classe UGS percebe-se que a partir daquele valor de atraso, o parâmetro deixa de ser interpretado apenas como “*low*”, sendo mapeado também para o termo “*medium*”, e à medida que o parâmetro continua crescendo, também é traduzido

como “*high*”. Naturalmente, os recursos para essa classe começam a crescer, como pode ser visto por volta dos 30 s, na Figura 31. Mesmo assim, a soma de largura de banda para UGS não ultrapassa 40%, ficando bem próximo desse valor. Por essa razão a média do atraso continua subindo, mas com taxa bem menor - Figura 30. Por outro lado, os recursos de BE caem para valores inferiores à sua demanda. Imediatamente o nível de ocupação em sua fila inicia novo crescimento até o máximo de 40% no instante 41,3 s - Figura 29. Com o fim dos fluxos UGS aos 40 s de simulação, o atraso dessa classe cai verticalmente momentos depois, precisamente no segundo 41,5. Nesse mesmo ponto, a Figura 35 indica que uma realocação eleva os recursos para BE e a ocupação em sua fila volta a cair - Figura 29. Finalmente, como não há mais fluxo UGS após os 40 s de simulação, essa classe não é mais contemplada nas próximas distribuições de recursos. Não houve ocorrência de perda de pacotes UGS durante a simulação.

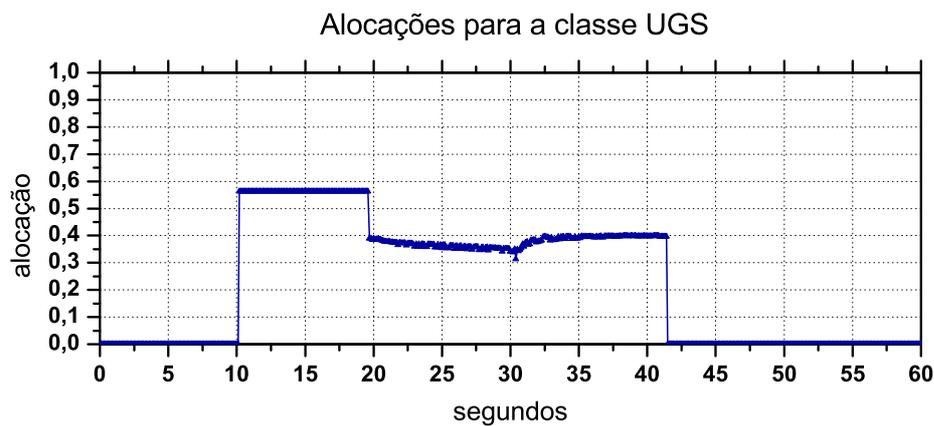


Figura 31: Pesos atribuídos à classe UGS.

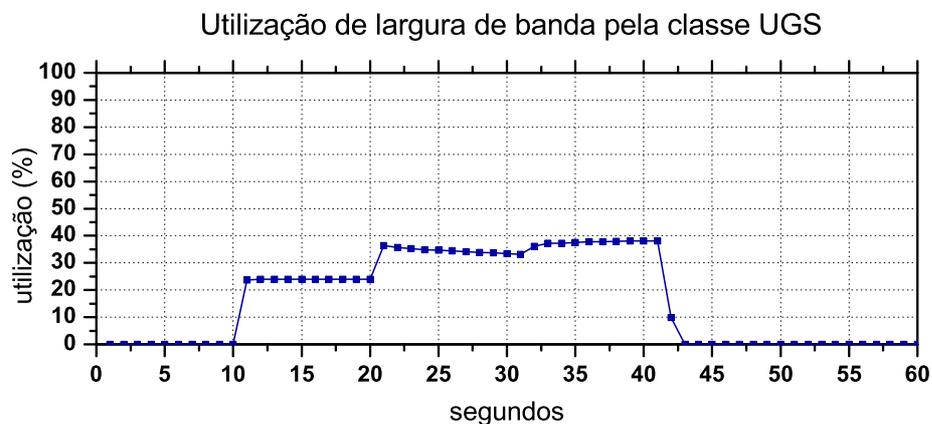


Figura 32: Utilização de largura de banda pela classe UGS.

Um aspecto interessante quando se comparam as Figura 31 e 32 é a semelhança entre as formas dos gráficos. Até os 10 s de simulação, não há tráfego UGS, portanto não há utilização de banda por essa classe. A partir de 10 s, são usados 25% da banda e essa utilização permanece até os 20 s, momento em que a demanda UGS cresce para 40% e a utilização alcança o nível de 35%. Dito de outro modo, a classe UGS aproveita toda largura de banda alocada e, mesmo assim, os recursos não são suficientes para atender a todos os pacotes; como resultado, o tempo médio de espera cresce, conforme esclarecido anteriormente. Quando mais recursos são alocados, a utilização também cresce, o atraso nas filas se equilibra e posteriormente cai a zero. Deve-se salientar que, quando a alocação chega próximo de 40%, a utilização também chega a esse patamar devido ao atendimento não só dos pacotes UGS que continuam chegando, como também daqueles que se acumularam nas filas da classe. Inclusive, o atraso cai verticalmente a zero porque o tráfego UGS cessa a partir da medida 400 e o atendimento que se segue ocorre exclusivamente devido aos pacotes remanescentes. Portanto, o último pacote atendido possui atraso diferente de zero.

O primeiro aspecto nos resultados da classe nrtPS que se destaca em relação aos da classe UGS é a grande variação nos valores de ocupação ao longo da simulação, como pode ser observado na Figura 28. Essa grande variabilidade se deve essencialmente às características do tráfego nrtPS: tamanho dos pacotes e tráfego com periodicidade variável. Naturalmente, o uso da média aritmética para obtenção dos resultados, conforme descrito na seção 5.2.2, também influencia nessa característica. Entretanto, essa variabilidade não se propaga com a mesma intensidade para as atribuições dos pesos, como pode ser comprovado na Figura 33. Isso ocorre por dois motivos: o uso de limiares inferior e superior amortece a variabilidade, uma vez que os valores podem excursionar dentro de uma faixa sem provocar realocação; é preciso grandes variações nos valores de ocupação das filas nrtPS para provocar diferenças significativas nos pesos atribuídos. A Figura 34 indica que a utilização da largura de banda pelas fontes nrtPS acompanham a alocação de recursos.

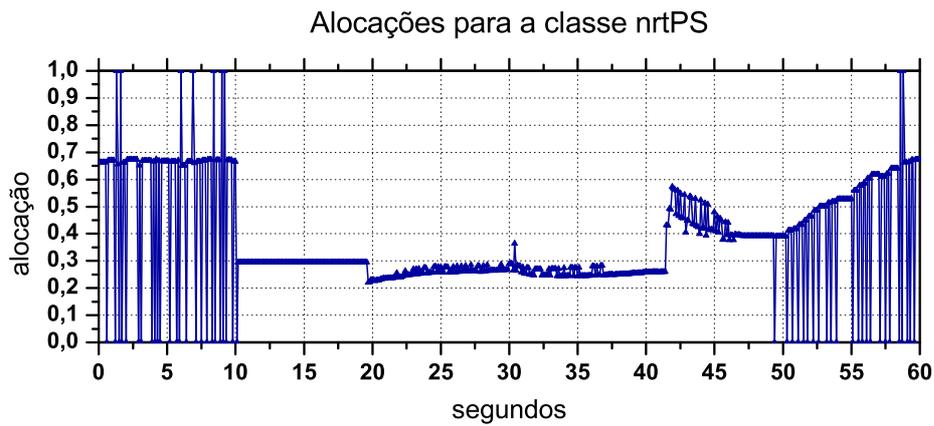


Figura 33: Pesos atribuídos à classe nrtPS.

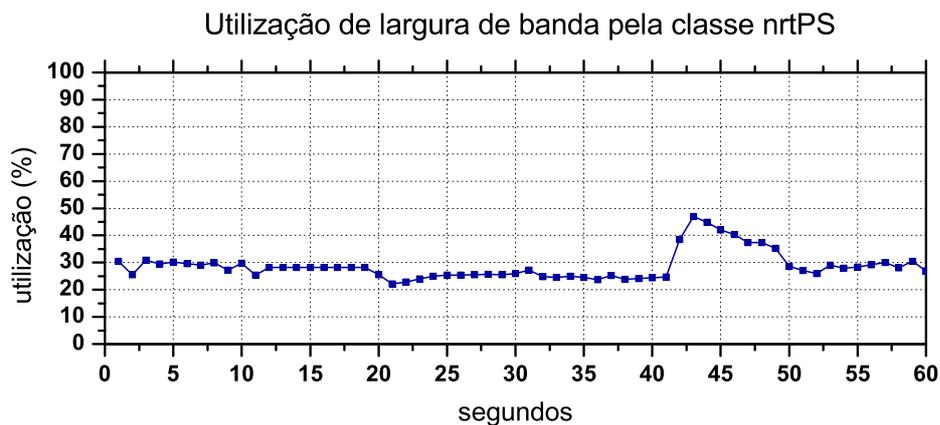


Figura 34: Utilização de largura de banda pela classe nrtPS.

Mesmo com o comportamento descrito, pode-se identificar uma tendência nas curvas apresentadas, ditada fundamentalmente pelos valores de recursos alocados. Por exemplo, conforme discutido anteriormente, quando o alocador *fuzzy* disponibiliza mais recursos para BE no instante 19,7 s, a reserva de banda nrtPS cai para cerca de 22% (Figura 27), portanto inferior à demanda de 30%. A partir dessa nova configuração, pode-se perceber na Figura 28 que a ocupação começa a crescer e volta a cair apenas por volta dos 42 s de simulação, momento em que alocações de até 56% superam novamente a demanda. Deve-se ressaltar que não ocorreu perda de pacotes nrtPS durante a simulação.

Embora a classe BE apresente maior demanda durante toda a simulação (40% da largura de banda disponível para DL), os recursos disponibilizados ficam abaixo dessa necessidade durante boa parte da simulação. Isso ocorre porque BE possui menor priori-

dade e a necessidade das demais classes implica em consumo de boa parte dos recursos. Ainda assim, BE recebeu uma soma de recursos que impediu que seus valores de ocupação chegassem ao limite da capacidade de sua fila. Adicionalmente, garantir uma fatia de recursos às classes de menor prioridade contribui para evitar o problema de *starvation*. Vale destacar que o maior valor de atraso na fila BE (8,19 ms) é bem maior que o atraso experimentado nas filas UGS (o maior chegou a 1,25 ms). Porém, apesar do atraso relativamente elevado, não houve ocorrência de perda de pacotes. Essas observações estão de acordo com os perfis de qualidade de serviço de cada aplicação.

Como explicado anteriormente, se inicia uma série de realocações a partir dos 41,8 s, que eleva os recursos de BE para aproximadamente 60%. Nessa mesma medida, foi finalizado o atendimento às filas UGS, permitindo que os recursos alocados a essa classe pudessem ser dirigidos às demais. Adicionalmente, o nível de ocupação nas filas nrtPS mantém um tendência de queda. Essa conjugação de fatores, associada ainda ao nível de ocupação alcançado pela fila BE, permitiu essa quantia de recursos para BE. Naturalmente, a utilização de largura de banda dá um salto, como pode ser verificado na Figura 36, e o nível de ocupação da fila BE, Figura 29, cai até o fim da simulação.

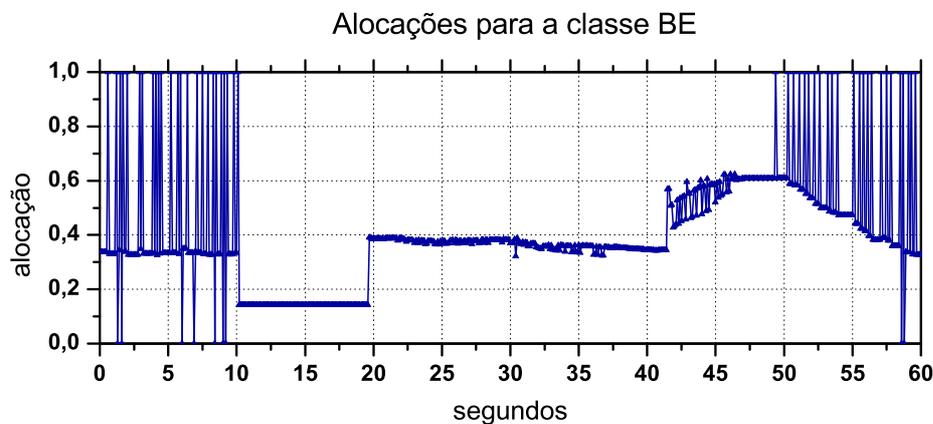


Figura 35: Pesos atribuídos à classe BE.

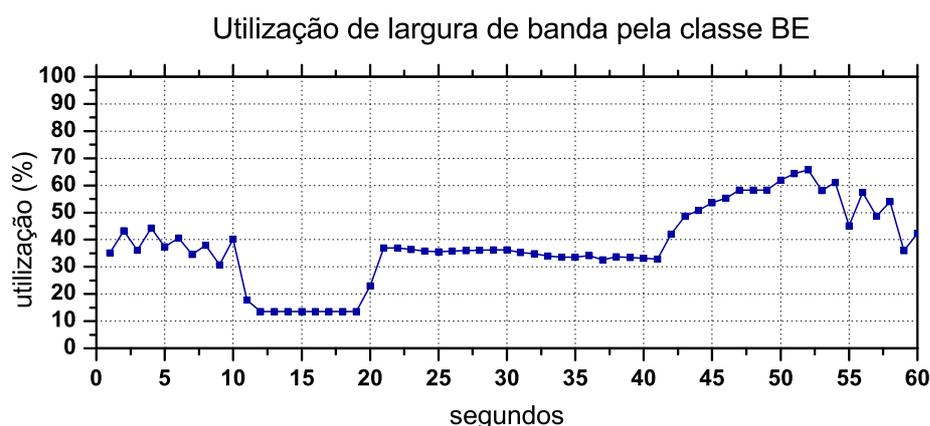


Figura 36: Utilização de largura de banda pela classe BE.

Uma das grandes vantagens da abordagem *fuzzy* na tarefa de alocação, evidenciada nos resultados apresentados, é propiciar o equilíbrio e justiça na distribuição de recursos. O interessante é que esses conceitos são subjetivos, assim como o termo prioridade. O entendimento do que venha a ser equilíbrio e justiça na distribuição dos recursos nos diversos cenários está no domínio do especialista humano. É exatamente esse conhecimento que o sistema *fuzzy* busca reproduzir quando, mesmo com demanda superior à capacidade durante um terço da simulação, minimiza o atraso nas filas UGS, e, ao mesmo tempo, mantém níveis de atendimento razoáveis para as classes de menor prioridade, evitando *starvation* e perda de pacotes. Por essas razões, pode-se concluir que a abordagem *fuzzy* obteve sucesso no objetivo de garantia de QoS estatístico e se mostra uma ferramenta poderosa na tarefa de provisionamento de recursos.

6.5 Cenário 4

No último cenário, são configurados representantes para as quatro classes IEEE 802.16. Durante toda a simulação, cada classe, através das aplicações, demanda igualmente 20% da largura de banda disponível. Classes UGS e BE, especificamente, demandam cada uma mais 20% em intervalos de tempo diferentes, indicando picos de utilização por um determinado tipo de aplicação. Dito de outro modo, em algum momento, a classe UGS terá demanda de 40% e em outro momento a classe BE terá uma demanda também de 40%. No restante da simulação, UGS e BE consomem 20% cada uma. Essas duas classes de serviços representam aplicações com requisitos de QoS diferentes e com prioridades diferentes. O objetivo, portanto, é verificar a atuação da estratégia *fuzzy* no atendimento

de serviços com características bem distintas quando esses apresentam picos de tráfego. Espera-se que os recursos necessários sejam destinados para as aplicações no momento em que apresentarem maior demanda, sempre observando a prioridade entre as aplicações, os recursos disponíveis e a situação das filas de atendimento. A Figura 37 ilustra a rede configurada para esse cenário e a Tabela 8 traz a matriz de tráfego. A composição detalhada do tráfego pode ser consultada no apêndice D.

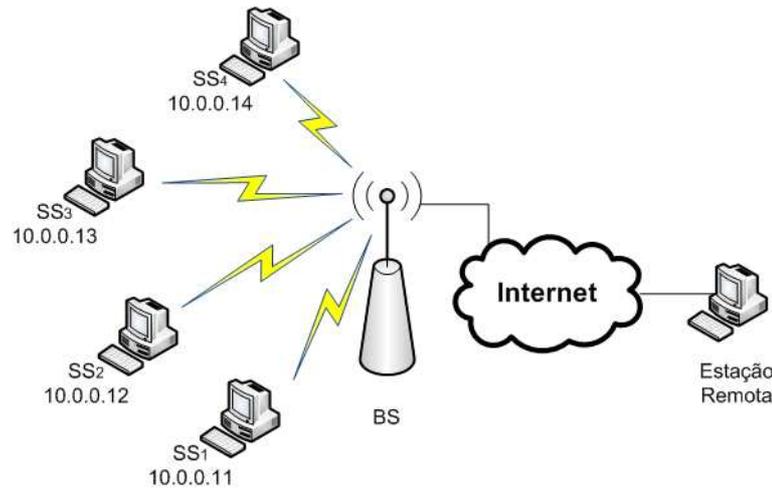


Figura 37: Rede para o cenário 4.

Tabela 8: Parâmetros de tráfego para o Cenário 4.

<i>Parâmetros de Tráfego</i>				
Classe	Aplicação	Destino	Início-Fim (s)	Demanda
UGS	VoIP	SS_1	0 – 60	20%
	VoIP	SS_3	30 – 50	10%
	VoIP	SS_4	30 – 50	10%
rtPS	MPEG Streaming Audio	SS_1	0 – 60	8%
	MPEG Streaming Audio	SS_2	0 – 60	12%
nrtPS	Transferência de arquivos	SS_3	0 – 60	15%
	Transferência de arquivos	SS_4	0 – 60	5%
BE	HTTP	SS_1	0 – 60	15%
	Terminal Remoto	SS_2	0 – 60	5%
	HTTP	SS_3	0 – 20	12%
	HTTP	SS_4	0 – 20	8%

Conforme dito anteriormente, cada classe demanda 20% da largura de banda durante toda a simulação. No intervalo entre 0 e 20 s, aplicações BE consomem além dos 20% outros 20%. O mesmo acontece com a UGS, no entanto entre 30 e 50 s de simulação. Dessa maneira, nesses dois intervalos, as aplicações requisitam 100% da largura de banda

disponível. À exemplo do que é feito nas simulações anteriores, no início dos testes cada classe recebe 25% dos recursos e o mecanismo de histerese atua observando todos os tipos de eventos.

Logo no início da simulação, quadro MAC 50, é medida ocupação 0 nas filas nrtPS, para em seguida o valor ser diferente de zero novamente. Pelo algoritmo, o alocador é acionado duas vezes seguidas: parâmetro diferente de zero vai para zero e parâmetro igual a zero vai para diferente de zero. Após o segundo acionamento, a nova configuração é a seguinte: UGS, 39,2%; rtPS, 29,4%; nrtPS, 20,3%; BE, 11,1%. Esses pesos permanecem durante várias medidas, como pode ser visto na Figura 38.

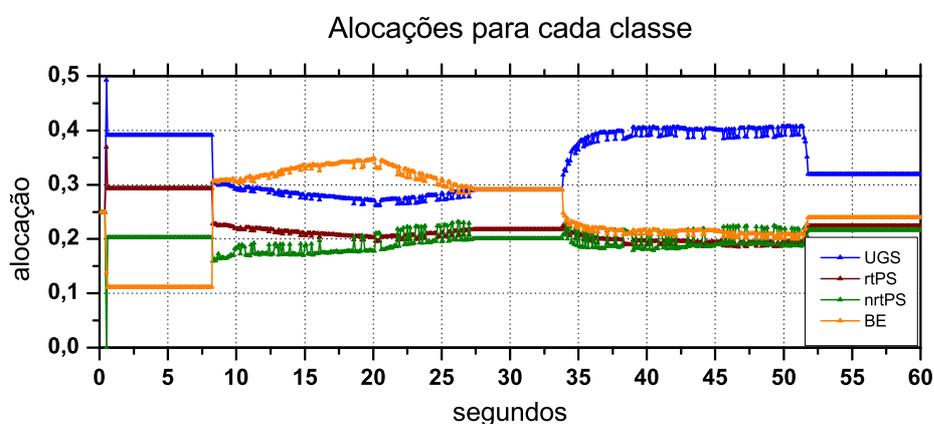


Figura 38: Pesos associados às classes durante a simulação.

Apenas na medida 83, essa configuração é modificada. Com uma diferença acentuada entre demanda e recursos, 40% e 11,1% respectivamente, a Figura 48 mostra que ocupação na fila BE cresce em ritmo acelerado. Um primeiro acionamento acontece aos 8,3 s de simulação, ou medida 83, e sucessivas alocações aumentam ainda mais os recursos para BE à medida que a ocupação vai crescendo. O maior valor de alocação, 34,9%, coincide com o maior valor de ocupação, 41,6%, aos 20,1 s. Esses recursos aliviaram a situação, mas ainda não eram suficientes, por isso a ocupação continuava crescendo, mesmo que em um ritmo menor. Após os 20,0 s, medida 200, se encerram os fluxos BE para SS_3 e SS_4 e a demanda dessa classe volta para 20%, por isso, com mais recursos, a ocupação começa a cair e com ela a reserva de BE também vai diminuindo na mesma proporção. Ao final dessa série de alocações no instante 27,4 s, a configuração fixada é a seguinte: UGS, 29%; rtPS, 21,8%; nrtPS, 20,1%; BE, 29,1%.

A soma de 29% dos recursos é suficiente para cobrir a demanda inicial de 20% da

classe UGS. Por isso, de acordo com a Figura 39, a média dos atrasos permanece 0 ms até o instante em que novas fontes adicionam mais 20% de demanda (medida 300 ou 30 s de simulação). Ao atingir o limiar inferior, fixado para a classe (1 ms), a Figura 38 indica uma redistribuição dos recursos que eleva a soma destinada a BS para 31,8%, ainda insuficiente. Apesar disso, o crescimento do atraso é menos acentuado e diminui conforme aumenta a soma de recursos com as sucessivas redistribuições. Quando a soma de recursos iguala a demanda, o atraso para de crescer e cai a 0 ms momentos depois com o fim do fluxo UGS para SS_3 e SS_4 .

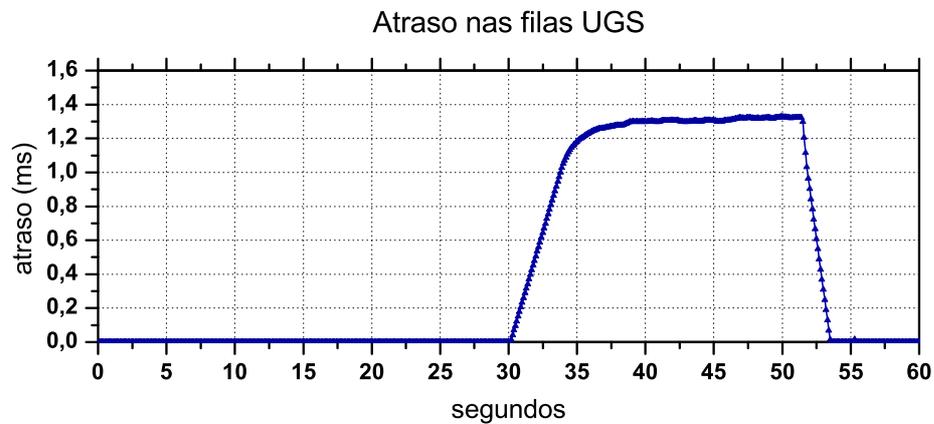


Figura 39: Tempo médio de espera nas filas UGS.

Até os 30 s de simulação, a alocação supera a demanda das fontes UGS. De acordo com a Figura 41, a utilização de recursos permanece exatamente em 20,0%, valor da carga UGS nesse intervalo. Após o início dos novos fluxos UGS, o tempo médio de espera cresce até ocorrer a nova alocação, instante próximo aos 35 s de simulação na Figura 40. Com demanda e recursos, a utilização cresce na mesma proporção.

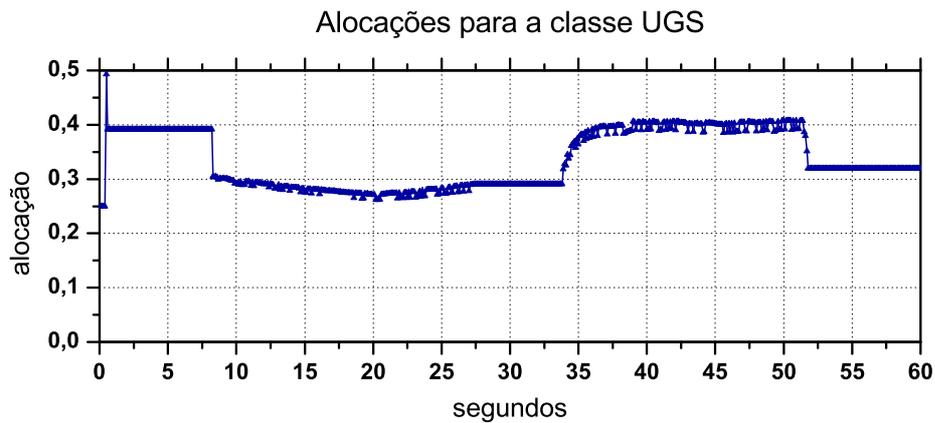


Figura 40: Pesos atribuídos à classe UGS.

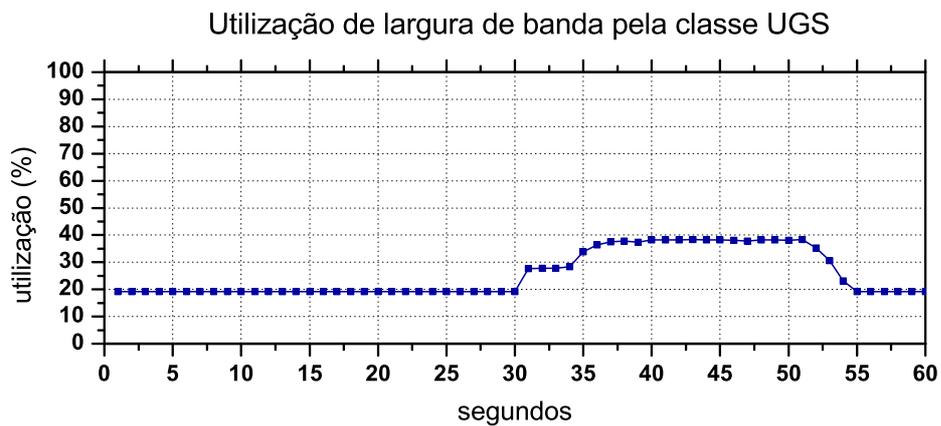


Figura 41: Utilização de largura de banda pela classe UGS.

A demanda rtPS permanece constante durante toda a simulação, 20% da largura de banda disponível. Durante quase todo o tempo, a quantidade de recursos alocados supera a demanda e por essa razão a média do atraso nas filas rtPS permanece dentro de uma faixa entre 0 e 0,13 ms de atraso, como mostra Figura 42. A exceção fica para o intervalo entre 39,0 s-51,5 s, período em que a soma de recursos fica ligeiramente abaixo da demanda, segundo a Figura 43. Nesse período, o atraso rtPS começa a crescer e atinge um pico de 0,74 ms. Vale destacar que o sistema *fuzzy* buscou destinar à classe rtPS a banda justa para a sua demanda, fato evidenciado na Figura 43. As vezes em que isso não ocorre, coincide com os maiores valores de alocação para UGS, classe de maior prioridade. A Figura 44 também confirma essa idéia, haja vista que a utilização permanece quase constante durante toda simulação.

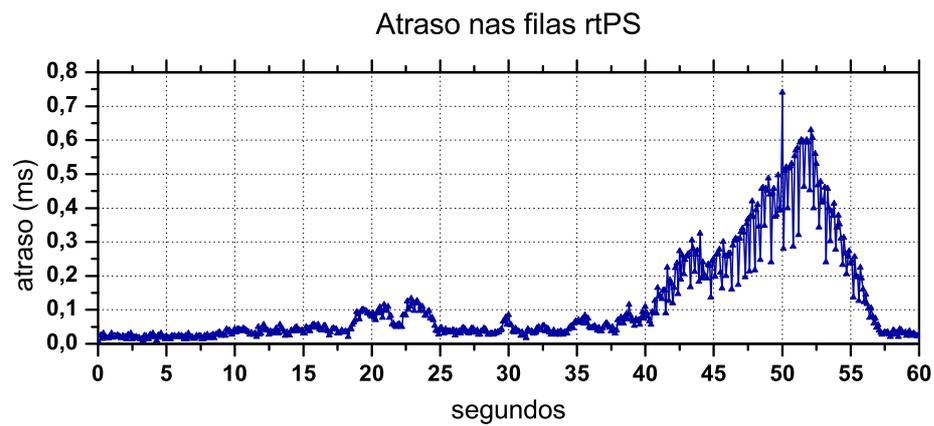


Figura 42: Tempo médio de espera nas filas rtPS.

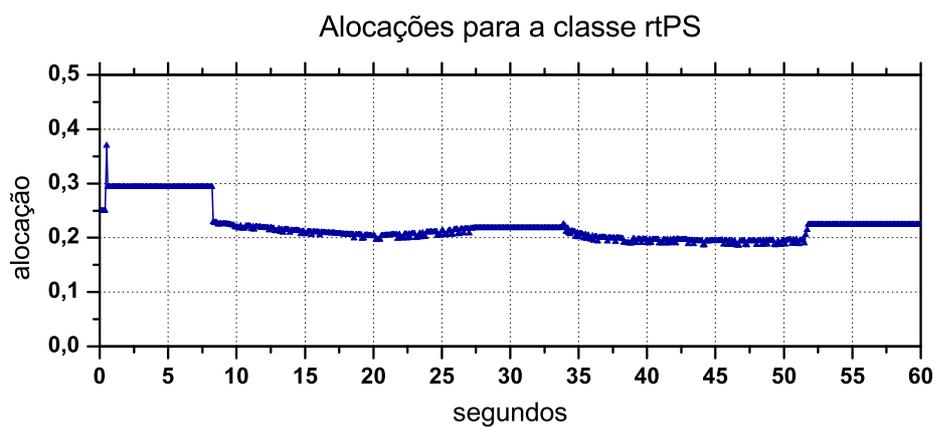


Figura 43: Pesos atribuídos à classe rtPS.

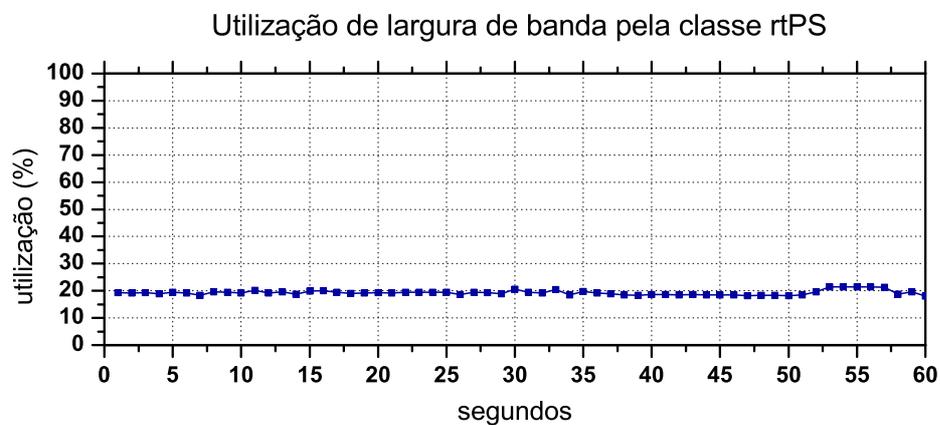


Figura 44: Utilização de largura de banda pela classe rtPS.

Da mesma forma que para classe rtPS, a estratégia buscou alocar os recursos necessários para nrtPS. Entretanto, por ter menor prioridade, essa classe acaba cedendo mais recursos nos momentos de maior demanda das demais com maior prioridade. Pode-se notar na Figura 45 dois intervalos em que o nível de ocupação nas filas nrtPS cresce para depois voltar a diminuir: entre os instantes medidas 8,3 s-27,4 s, e 33,8 s-51,5 s. Nesses mesmos intervalos, os recursos para nrtPS ficam abaixo de sua demanda de 20%.

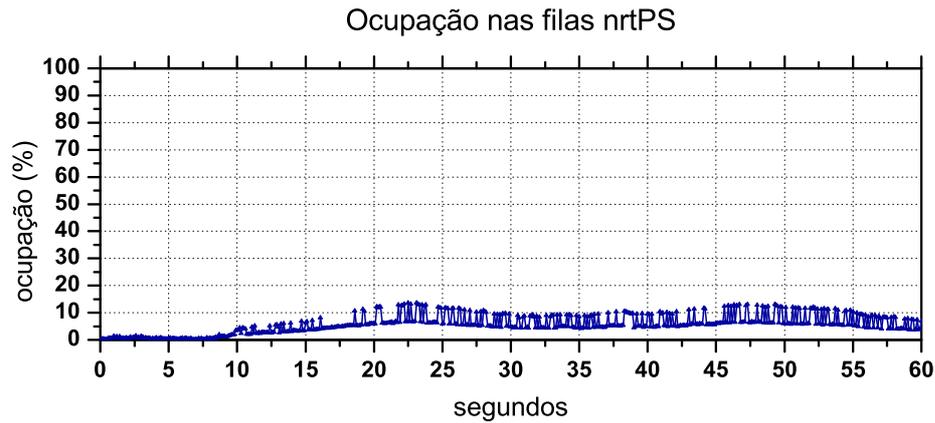


Figura 45: Ocupação média nas filas nrtPS.

A demanda dos fluxos nrtPS também são proporcionalmente constantes durante toda simulação, 20% de toda a carga. As Figuras 46 e 47 comprovam a alocação justa da solução. Os percentuais de alocação e de utilização acompanham os 20% de demanda.

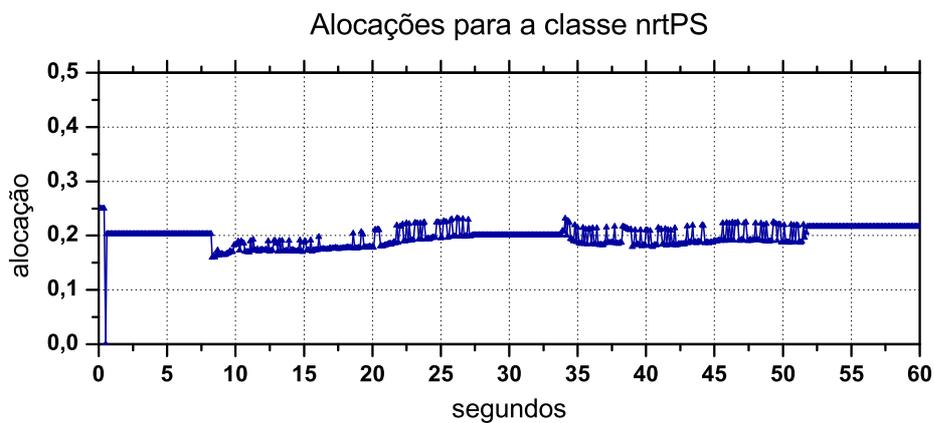


Figura 46: Pesos atribuídos à classe nrtPS.

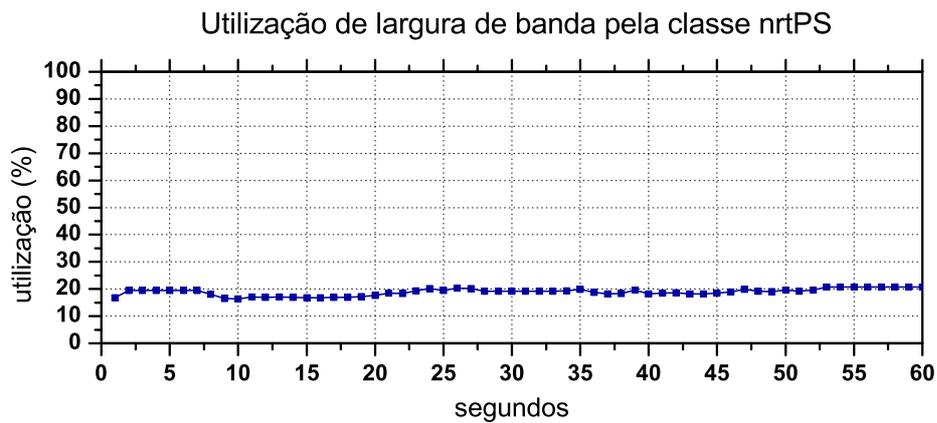


Figura 47: Utilização de largura de banda pela classe nrtPS.

Por ter menor prioridade, a classe BE é sempre penalizada quando as demais classes demandam recursos. Por essa razão, a ocupação na fila BE cresce até os 20,0 s de simulação, momento em que o fluxo BE retorna para 20% da largura de banda. Entretanto, deve-se ressaltar que, apesar da ocupação sempre crescer, as alocações realizadas garantem que esse crescimento não seja ainda mais acentuado. Inclusive, mesmo com menor prioridade, BE recebe a maior parte dos recursos no intervalo entre 8,3 s-27,4 s, segundo a Figura 38. Após a cessão dos tráfegos das fontes SS_3 e SS_4 , a ocupação cai até um determinado valor, que permanece constante até o fim da simulação. Isso ocorre, pois em realocações posteriores, os recursos para BE recuam para 20%, valor igual à demanda.

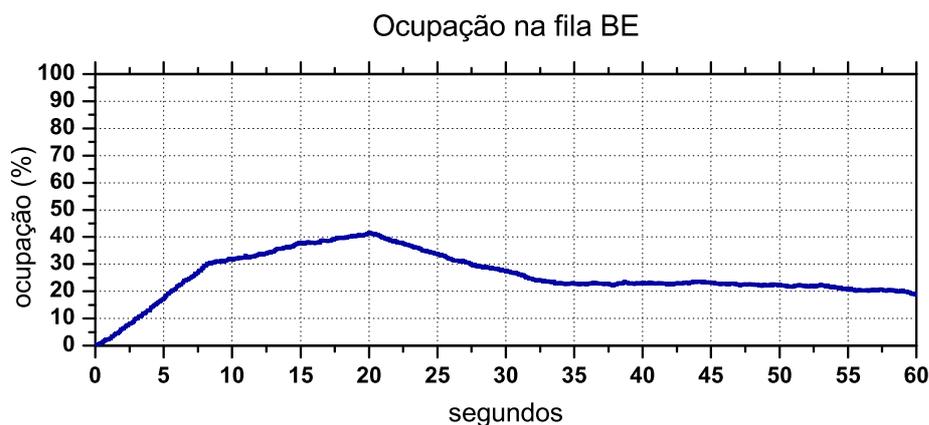


Figura 48: Ocupação média nas filas BE.

Tal qual nos casos anteriores, a utilização dos recursos BE, Figura 50, acompanha a alocação de recursos, Figura 49.

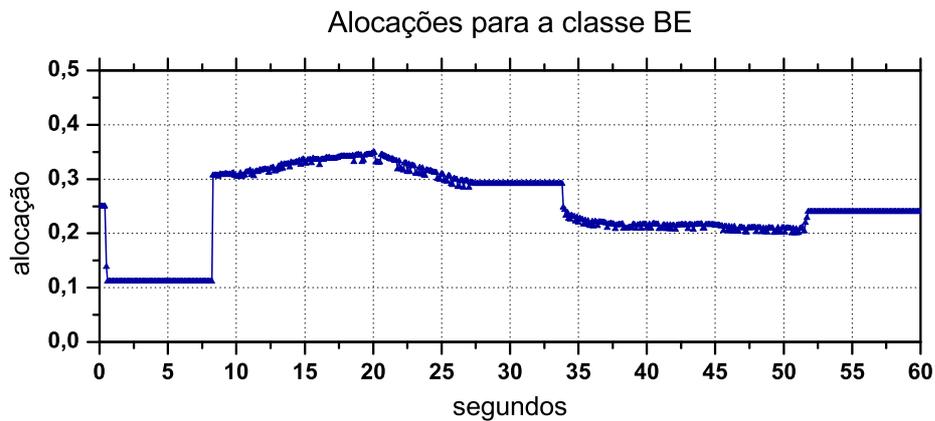


Figura 49: Pesos atribuídos à classe BE.

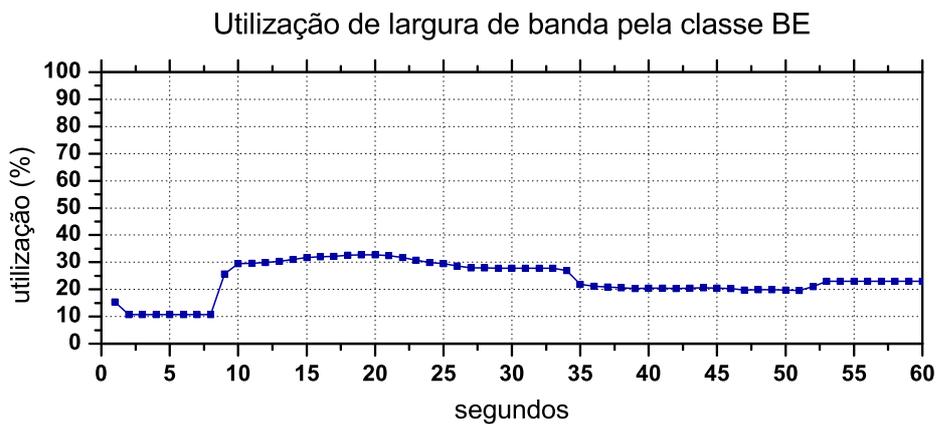


Figura 50: Utilização de largura de banda pela classe BE.

Considerando as análises gráficas anteriores, pode-se concluir que a estratégia *fuzzy* foi eficaz ao atender serviços distintos com variação de demanda. O sistema percebeu as mudanças nos perfis de tráfego e alocou os recursos dinamicamente, obedecendo à prioridade ponderada pela demanda entre as classes e pelos recursos disponíveis. Olhando para os gráficos, também é possível estabelecer um paralelo entre sistemas de controle moderno e sistema de controle *fuzzy*. Assim como no controle moderno, pode-se identificar um caminho de realimentação, ainda que indireto. A modificação da soma de recursos para cada classe se reflete no parâmetro de controle associado, ou seja no parâmetro de QoS de cada classe. Essa variação na medida é percebida pelo sistema e pode resultar em mais ou menos recursos em uma próxima realocação, a depender dos limiares estabelecidos para cada classe, nas funções de pertinência. As consecutivas realocações, motivadas

pela diferenciação entre parâmetro medido e limiar estabelecido, leva o sistema a alocar recursos justos à demanda, ou seja, o sistema busca a estabilidade. A estabilidade para o sistema significa todos os parâmetros dentro dos seus respectivos limites, ou todos os parâmetros abaixo dos seus respectivos limiares inferiores, ou, finalmente, todas as medidas iguais a zero. Caso todas as medidas sejam iguais a zero, significa que não há tráfego, de nenhuma classe, para ser encaminhado às SSs. Para qualquer situação distinta dessas enumeradas, o módulo *Fuzzy Allocator* será acionado para inferir uma nova configuração de pesos.

7 *Conclusões e Trabalhos Futuros*

*“Se não houver frutos,
valeu a beleza das flores...
Se não houver flores,
valeu a sombra das folhas...
Se não houver folhas,
valeu a intenção da semente.”*

Henfil

ESTE capítulo traz as considerações finais sobre o trabalho realizado, consolidando os resultados expostos anteriormente, na forma de conclusões e observações. Adicionalmente, as principais contribuições desta dissertação, bem como as propostas para trabalhos futuros são apresentados.

7.1 Conclusões

Este trabalho apresentou uma nova proposta, baseada em lógica *fuzzy*, para distribuição de recursos no canal DL em uma rede IEEE 802.16. Desde o princípio, o trabalho teve como objetivo geral contribuir para a arquitetura de recursos em redes IEEE 802.16. Para tanto, o trabalho buscou investigar a aplicabilidade de lógica *fuzzy* no provisionamento de QoS em redes WiMAX como alternativa às soluções tradicionais, baseadas em modelagem analítica. Essa idéia surgiu a partir da percepção de que as ferramentas matemáticas clássicas, como as equações diferenciais, não são suficientemente flexíveis para representar, ao mesmo tempo, eficaz e fielmente o cenário atual de redes convergentes.

A estratégia apresentada se baseou em outros trabalhos, ou por incorporar algumas de suas idéias ou por aperfeiçoar seus pontos fracos. Até onde alcança o conhecimento do autor, podem-se citar também pontos originais, como os algoritmos para detecção de eventos e a interpretação do conceito de prioridade através das funções de pertinência do sistema especialista *fuzzy*. Além disso, a estratégia foi elaborada visando sua inserção na arquitetura de QoS desenvolvida em [6], por isso adotou diversas considerações feitas nesse trabalho, incluindo o esquema de filas proposto para a BS 802.16.

Para validar a estratégia e avaliar seu desempenho, o desenho esquemático bem como os algoritmos apresentados no Capítulo 5 foram implementados em C++ e inseridos em um ambiente de simulação com as principais funcionalidades de uma rede WiMAX. Foram quatro cenários elaborados com objetivos específicos.

Os resultados obtidos e discutidos comprovam o potencial do uso da ferramenta *fuzzy* na tarefa de provisionamento de recursos em redes IEEE 802.16. A abordagem *fuzzy* se destacou principalmente pela sua flexibilidade e habilidade para perceber e se adaptar a mudanças na dinâmica da rede. Os gráficos mostram que o sistema *fuzzy* busca seguir a demanda de cada classe e atender os requisitos de QoS estabelecidos para cada uma. As funções de pertinência associada à base de regras refletem com bastante propriedade o conhecimento subjetivo sobre o que venha a ser justiça, equilíbrio e prioridade na distribuição de recursos. Além disso, a qualidade nos resultados sugere que não há necessidade de se aumentar a complexidade do sistema, inserindo novas partições *fuzzy*. A suposta melhoria que se obtém com a inserção de novas funções e novas regras não compensa a maior complexidade e carga computacional. Pode-se entender essa afirmação como um compromisso entre eficiência e simplicidade.

Como resultado desta dissertação, foi produzido o artigo "A Fuzzy Approach to Pro-

vide QoS in IEEE 802.16“. O artigo foi aceito e apresentado no International Workshop on Telecommunications - IWT09, realizado em São Paulo, em fevereiro de 2009.

Finalmente, considerando todo o trabalho desenvolvido, pode-se concluir que os objetivos iniciais, referidos na seção 1.3, foram alcançados. Diversos pontos, entretanto, podem ser explorados e investigados em maior profundidade.

7.2 Trabalhos Futuros

Podem ser citados como trabalho futuro os seguintes temas:

- Desenvolvimento de algoritmos mais sofisticados para detecção de eventos na rede.
- Uso de algoritmo genético para otimização da base de regras.
- Investigação de outras variáveis características para o sistema *fuzzy* proposto.
- Adaptação da estratégia para sua aplicação também no canal *uplink* IEEE 802.16.
- Estudo da aplicabilidade de lógica *fuzzy* também para o módulo de controle de admissão.
- Implementação de mecanismos para priorização de filas dentro de uma mesma classe.

Referências

- [1] BANDARA, J.; SHEN, X.; NURMOHAMED, Z. Resource allocator for non real-time traffic in wireless networks using fuzzy logic. *Wirel. Pers. Commun.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 21, n. 3, p. 329–344, 2002.
- [2] CHEN, J.; JIAO, W.; WANG, H. A service flow management strategy for IEEE 802.16 broadband wireless access systems in TDD mode. In: *IEEE International Conference on Communications*. Seoul, Korea: IEEE Press, 2005. v. 5, p. 3422–3426.
- [3] SWEENEY, D. *WiMAX Operator's Manual: Building 802.16 Wireless Networks*. Second edition. Spring Street, 6th Floor, New York, NY 10013: Apress, 2006. 210 p.
- [4] TANENBAUM, A. S. *Redes de Computadores*. Tradução da 4ª ed. Rio de Janeiro, RJ: Editora Campus, 2004.
- [5] IEEE Std. 802.16, IEEE Standard for Local and Metropolitan Area Network Part 16: Air Interface for Fixed Broadband Wireless Access System. Out 2004. [S.l.].
- [6] MALINI, R. Q. *Uma Arquitetura de QoS e Modelo de Simulação para o Padrão IEEE 802.16d como Suporte a Serviços Multimídia Distribuídos. 66f*. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo, Vitória, ES, 2007.
- [7] ZADEH, L. A. Fuzzy sets. *Information and Control*, v. 8, n. 1, p. 338–353, 1965.
- [8] LEE, C. Fuzzy logic in control systems: Fuzzy logic controller. ii. *Systems, Man and Cybernetics, IEEE Transactions on*, IEEE Press, v. 20, n. 2, p. 419–435, Mar/Apr 1990.
- [9] LIN C. T.; LEE, C. S. G. *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*. Upper Saddle River, NJ: Prentice-Hall PTR, c1996. 797 p.
- [10] CARVALHO, G. H. de. *Desenvolvimento do Módulo de Requisição de Banda do Padrão IEEE 802.16*. 2007. Monografia (Ciências da Computação) - Departamento de Informática, Universidade Federal do Espírito Santo. Orientador: Anilton Salles Garcia.
- [11] BERNARDO, L. A. K. *Desenvolvimento do Algoritmo de Alocação de Banda de Uplink para o Padrão IEEE 802.16*. 2008. Monografia (Ciências da Computação) - Departamento de Informática, Universidade Federal do Espírito Santo. Orientador: Anilton Salles Garcia.
- [12] CROZIER, E.; KLEIN, A. WiMAX's technology for LOS and NLOS environments. *WiMAX Forum*, p. 1–10, Ago 2007.
- [13] NAIR, G. et al. IEEE 802.16 medium access control and service provisioning. *Intel Technology Journal*, Intel Corp., USA, v. 08, n. 03, p. 213–228, 2004.

- [14] Senza Fili Consulting. *Fixed, nomadic, portable and mobile applications for 802.16-2004 and 802.16e WiMAX networks*. [S.l.], Nov 2005. Disponível em: <http://www.wimaxforum.org/news/downloads/Applications_for_802.16-2004_and_802.16e_WiMAX_networks_final.pdf>.
- [15] CHO, D.-H. et al. Performance analysis of the IEEE 802.16 wireless metropolitan area network. In: *DFMA '05: Proceedings of the First International Conference on Distributed Frameworks for Multimedia Applications*. Washington, DC, USA: IEEE Computer Society, 2005. p. 130–137.
- [16] CHU, G.; WANG, D.; MEI, S. A QoS architecture for the MAC protocol of IEEE 802.16 BWA system. In: . [s.n.], 2002. v. 1, p. 435–439 vol.1. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1180654>.
- [17] SAYENKO, A.; ALANEN, O.; HAMALINEN, T. Scheduling solution for the IEEE 802.16 base station. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 52, n. 1, p. 96–115, 2008.
- [18] ZADEH, L. A. Outline of a new approach to the analysis of complex system and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 3, n. 1, p. 28–44, 1973.
- [19] REZENDE, S. O. *Sistemas Inteligentes - Fundamentos e Aplicações*. 1. ed. SP: Manole, 2003.
- [20] SANDRI, S.; CORREA, C. Lógica nebulosa. In: *Anais da V Escola de Redes Neurais*. São José dos Campos, SP: ITA, 1999. p. c073–c090. Disponível em: <<http://www.ele.ita.br/cnrn/minicursos-5ern/log-neb.pdf>>. Acesso em: 23 Ago 2007.
- [21] LEE, K. O.; HAHM, J. H.; KIM, Y. S. QoS application method in portable internet. In: *Proceedings of Asia-Pacific Conference on Communications*. Perth, Australia: IEEE Press, 2005. p. 237–239.
- [22] WiMAX Forum. *WiMAX Forum Overview*. Jun 2001. Disponível em: <<http://http://www.wimaxforum.org/about/>>. Acesso em: 11 Ago 2008.
- [23] NUAYMI, L. *WiMAX: technology for broadband wireless access*. Chichester, GB: John Wiley & sons, 2007.
- [24] ANDREWS, J. G.; GHOSH, A.; MUHAMED, R. *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*. 1st. ed. Upper Saddle River, NJ, USA: Prentice Hall, 2007.
- [25] MAHESHWARI, S. *An Efficient QoS Scheduling Architecture for IEEE 802.16 Wireless MANs*. Dissertação (Mestrado) — Indian Institute of Technology, Bombay, India, 2005.
- [26] WALDBUSSER, S. *Remote Network Monitoring Management Information Base*. United States, 1995.
- [27] KAMAL, G.; MOUNIR, M. A.; ANNIE, G. Dynamic resource reservation in IEEE 802.16 broadband wireless networks. In: . Yale University, New Haven, CT, USA: [s.n.], 2006. v. 1, p. 140–148.

-
- [28] RESENDE, R. A. *Qualidade de serviço em rede IP utilizando lógica fuzzy*. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas., Campinas, SP, 2006.
- [29] SIRIPONGWUTIKORN, P.; BANERJEE, S.; TIPPER, D. Adaptive bandwidth control for efficient aggregate QoS provisioning. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 32, n. 3, p. 19–19, 2002.
- [30] ITU-T Recommendation G.114. *One way transmission time*. Geneva, Mai 2003.

Apêndice A - Inferência Fuzzy

Em [20], os autores resumem em alguns passos o processamento de inferência para o modelo clássico, entre eles o Mamdani. Esses passos foram adaptados para as escolhas adotadas neste trabalho e são apresentados em seguida. Considerando as regras codificadas como:

$$R_j : \text{Se } x_1 \text{ é } A_{1j} \text{ e } \dots \text{ e } x_n \text{ é } A_{nj} \text{ Então } y_j \text{ é } C_j \quad (7.1)$$

Passo 1: Seja x_i uma variável característica do problema, definida no universo X_i . A realização de x_i é definida como o valor $x_i^* \in X_i$ que esta assume em X_i em um dado momento. Em outras palavras, x_i^* é o valor medido por sensores e X_i é a faixa de valores que a variável característica x_i pode excursionar.

Passo 2: A compatibilidade da i -ésima premissa da j -ésima regra com x_i^* , ou seja, a compatibilidade de x_i^* , $1 \leq i \leq n$, com A_{ij} da regra R_j , $1 \leq j \leq m$, é definida por:

$$\alpha_{i,j} = \mu_{A_{i,j}}(x_i^*), \quad 1 \leq i \leq n, 1 \leq j \leq m \quad (7.2)$$

Passo 3: A compatibilidade global α_j da regra R_j , $1 \leq j \leq m$, com os x_i^* , após a avaliação individual das premissas, tal como feita no passo 2, é determinada pela equação 3.3, ou seja, por uma operação *min*.

$$\alpha_j = \min[\alpha_{1,j}, \dots, \alpha_{n,j}], \quad 1 \leq j \leq m \quad (7.3)$$

Passo 4: O α_j obtido é relacionado com o respectivo conjunto nebuloso C_j do conseqüente da regra R_j , dando origem a um conjunto C'_j , $1 \leq j \leq m$, através do operador de implicação definido na Equação 3.8 do seguinte modo:

$$\mu_{C'_j}(y) = \min[\alpha_j, \mu_{C_j}(y)], \quad \forall y \in Y \quad (7.4)$$

Passo 5: O operador *max* faz a agregação das contribuições das várias regras acionadas C'_j em um único conjunto nebuloso C' da forma descrita na Equação 7.5.

$$\mu_{C'}(y) = \max(\mu_{C'_1}(y), \dots, \mu_{C'_m}(y)), \quad \forall y \in Y \quad (7.5)$$

Para ilustrar o funcionamento de um processo de inferência, considere o seguinte exemplo com duas variáveis características, x e y , e uma variável de decisão, z . Cada variável lingüística, nesse exemplo, possui dois termos lingüísticos, ou partições *fuzzy*: A_1, A_2, B_1, B_2 e C_1, C_2 . Sendo dois a granularidade de cada variável na entrada, o número de entradas na base de regras deve ser quatro. No entanto, nesse exemplo, serão definidas apenas duas delas, apenas para ilustrar.

Regra 1: **Se** x é A_1 e y é B_1 **então** z é C_2

Regra 2: **Se** x é A_2 e y é B_2 **então** z é C_1 .

Considere que x' e y' são os valores medidos por sensores que devem ser instanciados como variáveis lingüísticas x e y , respectivamente.

$$\mu_{A_1}(x) = \begin{cases} \frac{x-3}{3} & \text{se } 3 \leq x \leq 5; \\ \frac{7-x}{3} & \text{se } 5 < x \leq 7. \end{cases} \quad \mu_{A_2}(x) = \begin{cases} \frac{x-5}{3} & \text{se } 5 \leq x \leq 7; \\ \frac{9-x}{3} & \text{se } 7 < x \leq 9. \end{cases}$$

$$\mu_{B_1}(y) = \begin{cases} \frac{y+5}{3} & \text{se } -5 \leq y \leq 0; \\ \frac{5-y}{3} & \text{se } 0 < y \leq 5. \end{cases} \quad \mu_{B_2}(y) = \begin{cases} \frac{y}{3} & \text{se } 0 \leq y \leq 5; \\ \frac{10-y}{3} & \text{se } 5 < y \leq 10. \end{cases}$$

$$\mu_{C_1}(z) = \begin{cases} \frac{z-1}{3} & \text{se } 1 \leq z \leq 3; \\ \frac{5-z}{3} & \text{se } 3 < z \leq 5. \end{cases} \quad \mu_{C_2}(z) = \begin{cases} \frac{z-4}{3} & \text{se } 4 \leq z \leq 6; \\ 1 & \text{se } 6 < z \leq 8. \end{cases}$$

Passo 1:

$$x' = 6, 4 \text{ e } y' = 2, 0$$

Passo 2:

$$\mu_{A_1}(x' = 6, 4) = 0, 3 \quad \mu_{A_2}(x' = 6, 4) = 0, 7 \quad \mu_{B_1}(y' = 2, 0) = 0, 6 \quad \mu_{B_2}(y' = 2, 0) = 0, 4$$

Passo 3:

$$\text{Regra 1: } \min[\mu_{A_1}, \mu_{B_1}] = \min[0, 3, 0, 6] = 0, 3$$

Regra 2: $\min[\mu_{A_2}, \mu_{B_2}] = \min[0,7, 0,4] = 0,4$.

Passo 4:

$$C'_2 = \min(0,3, \mu_{C_2}(z)), \quad \forall z \in Z \quad C'_1 = \min(0,4, \mu_{C_1}(z)), \quad \forall z \in Z$$

Passo 5:

$$C' = \max(C'_1, C'_2), \quad \forall z \in Z$$

As figuras abaixo ilustram os resultados das operações acima sobre os conjuntos *fuzzy*.

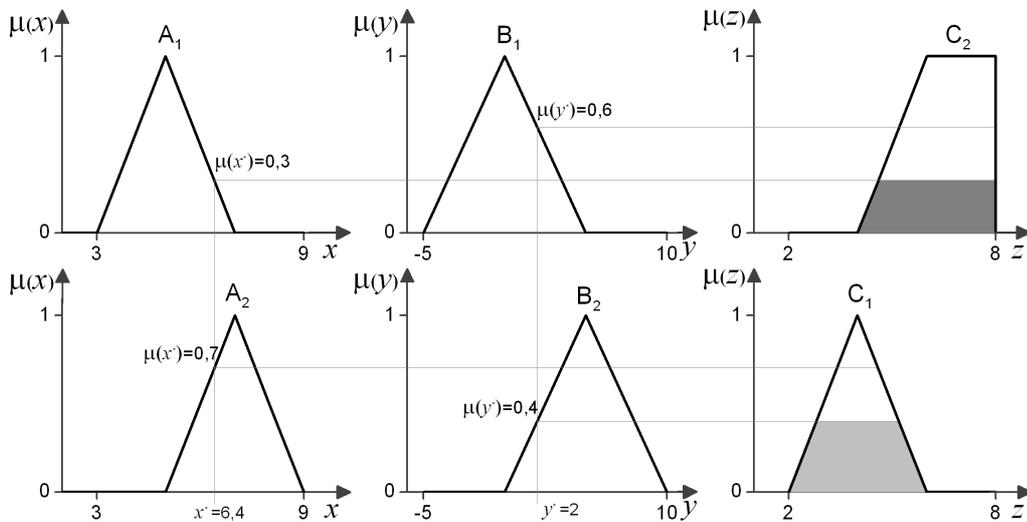


Figura 51: Processo de inferência ilustrado no Apêndice A.

A Figura 52 mostra o resultado do passo 5. Os conjuntos C'_1 e C'_2 compõem, segundo a operação *max*, o conjunto resultante do processo de inferência. No passo de defuzzificação é adotado o método COG, portanto calcula-se o centro de gravidade do conjunto *fuzzy* de saída obtido. O valor escalar produzido pelo sistema especialista é a abscissa do ponto encontrado como centro de gravidade, nesse caso 5.

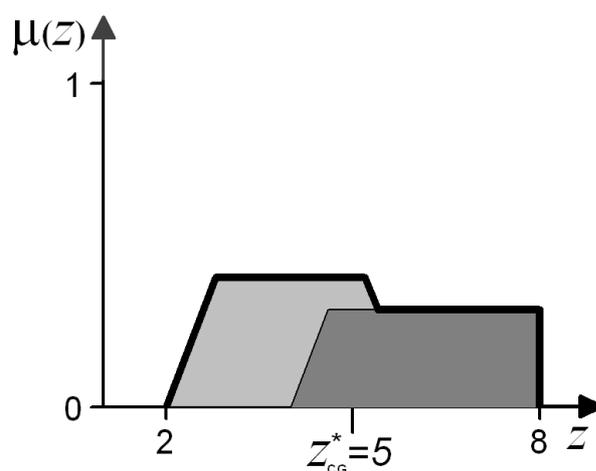


Figura 52: Resultado após a defuzzificação no processo de inferência ilustrado no Apêndice A.

Apêndice B - Base de Regras

Adotada para a Estratégia Fuzzy

	\overline{W}_{q_UGS}	\overline{W}_{q_rtPS}	\overline{N}_{nrtPS}	N_{BE}	BW_{UGS}	BW_{rtPS}	BW_{nrtPS}	BW_{BE}
1	low	low	low	low	low	low	low	low
2	low	low	low	medium	low	low	low	medium
3	low	low	low	high	low	low	low	high
4	low	low	medium	low	low	low	medium	low
5	low	low	medium	medium	low	low	medium	medium
6	low	low	medium	high	low	low	medium	high
7	low	low	high	low	low	low	high	low
8	low	low	high	medium	low	low	high	medium
9	low	low	high	high	low	low	high	high
10	low	medium	low	low	low	medium	low	low
11	low	medium	low	medium	low	high	low	medium
12	low	medium	low	high	low	medium	low	high
13	low	medium	medium	low	low	medium	medium	low
14	low	medium	medium	medium	low	high	medium	low
15	low	medium	medium	high	low	medium	medium	medium
16	low	medium	high	low	low	medium	high	low
17	low	medium	high	medium	low	medium	high	low
18	low	medium	high	high	low	medium	medium	medium
19	low	high	low	low	low	high	low	low
20	low	high	low	medium	low	high	low	medium
21	low	high	low	high	low	high	low	high
22	low	high	medium	low	low	high	medium	low
23	low	high	medium	medium	low	high	medium	low
24	low	high	medium	high	low	high	medium	medium
25	low	high	high	low	low	high	high	low

	\overline{W}_{q_UGS}	\overline{W}_{q_rtPS}	\overline{N}_{nrtPS}	N_{BE}	BW_{UGS}	BW_{rtPS}	BW_{nrtPS}	BW_{BE}
26	low	high	high	medium	low	high	medium	low
27	low	high	high	high	low	high	medium	medium
28	medium	low	low	low	high	low	low	low
29	medium	low	low	medium	high	low	low	medium
30	medium	low	low	high	medium	low	low	high
31	medium	low	medium	low	high	low	medium	low
32	medium	low	medium	medium	high	low	medium	low
33	medium	low	medium	high	high	low	low	medium
34	medium	low	high	low	medium	low	high	low
35	medium	low	high	medium	medium	low	high	low
36	medium	low	high	high	medium	low	medium	medium
37	medium	medium	low	low	high	medium	low	low
38	medium	medium	low	medium	high	medium	low	low
39	medium	medium	low	high	medium	medium	low	medium
40	medium	medium	medium	low	high	medium	low	low
41	medium	medium	medium	medium	high	medium	medium	low
42	medium	medium	medium	high	medium	medium	low	medium
43	medium	medium	high	low	medium	medium	medium	low
44	medium	medium	high	medium	high	medium	medium	low
45	medium	medium	high	high	medium	medium	medium	medium
46	medium	high	low	low	medium	high	low	low
47	medium	high	low	medium	medium	high	low	low
48	medium	high	low	high	medium	high	low	medium
49	medium	high	medium	low	medium	high	medium	low
50	medium	high	medium	medium	medium	high	medium	low
51	medium	high	medium	high	medium	high	medium	medium
52	medium	high	high	low	medium	high	medium	low
53	medium	high	high	medium	medium	high	medium	low
54	medium	high	high	high	medium	high	medium	medium
55	high	low	low	low	high	low	low	low
56	high	low	low	medium	high	low	low	medium
57	high	low	low	high	high	low	low	high

	\overline{W}_{q_UGS}	\overline{W}_{q_rtPS}	\overline{N}_{nrtPS}	N_{BE}	BW_{UGS}	BW_{rtPS}	BW_{nrtPS}	BW_{BE}
58	high	low	medium	low	high	low	medium	low
59	high	low	medium	medium	high	low	medium	low
60	high	low	medium	high	high	low	medium	medium
61	high	low	high	low	high	low	high	low
62	high	low	high	medium	high	low	high	low
63	high	low	high	high	high	low	medium	medium
64	high	medium	low	low	high	medium	low	low
65	high	medium	low	medium	high	medium	low	low
66	high	medium	low	high	high	medium	low	medium
67	high	medium	medium	low	high	medium	low	low
68	high	medium	medium	medium	high	medium	low	low
69	high	medium	medium	high	high	medium	low	medium
70	high	medium	high	low	high	medium	medium	low
71	high	medium	high	medium	high	medium	medium	low
72	high	medium	high	high	high	medium	medium	medium
73	high	high	low	low	high	high	low	low
74	high	high	low	medium	high	high	low	low
75	high	high	low	high	high	high	low	medium
76	high	high	medium	low	high	high	low	low
77	high	high	medium	medium	high	high	low	low
78	high	high	medium	high	high	high	low	medium
79	high	high	high	low	high	high	medium	low
80	high	high	high	medium	high	high	medium	low
81	high	high	high	high	high	high	medium	low

Apêndice C - Funções de pertinência para o Cenário 2

As funções de pertinência associadas às variáveis do espaço de entrada são matidas iguais àquelas apresentadas na seção 5.2.3.

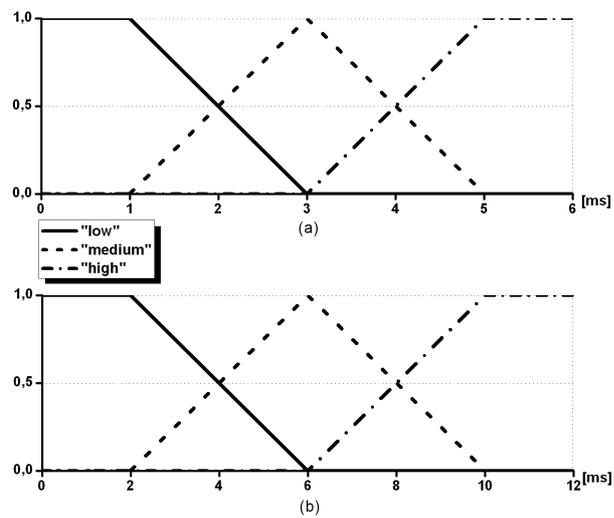


Figura 53: Partição *fuzzy* para as variáveis de entrada: (a) $\overline{W_{q-UGS}}$, (b) $\overline{W_{q-rtPS}}$

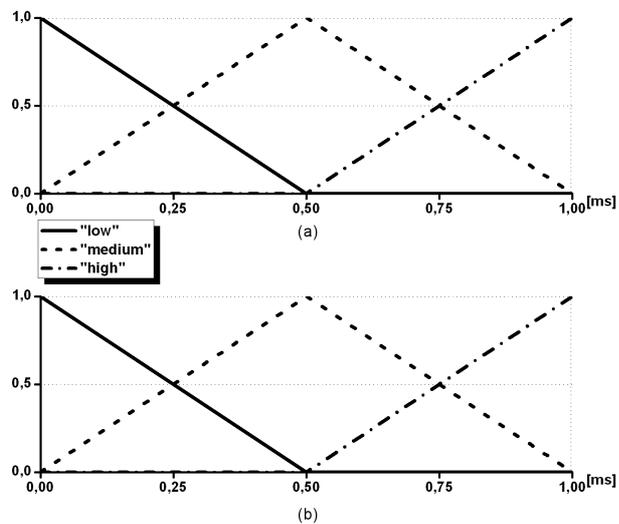


Figura 54: Partições *fuzzy* para as variáveis de entrada: (a) $\overline{N_{nrtPS}}$, (b) N_{BE}

As funções de pertinência do espaço de saída são ajustadas de maneira a representar diferentes interpretações para a expressão “prioridade”. Na primeira ordem, Figura 55, a classe BE (letra d) tem a maior prioridade. Observe que, para essa classe, 80% dos recursos é considerado alocação mediana, enquanto para a classe UGS, esse mesmo valor, é considerado bastante alto.

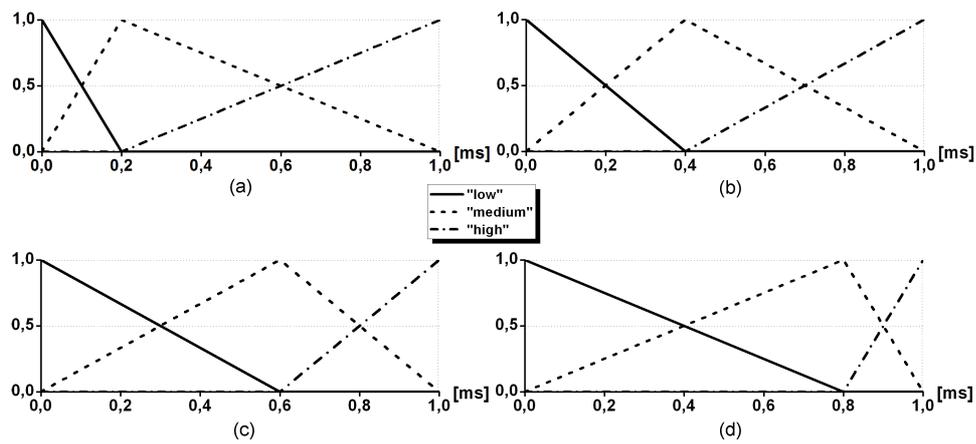


Figura 55: Prioridades:
 $(d)BW_{BE} > (c)BW_{nrtPS} > (b)BW_{rtPS} > (a)BW_{UGS}$.

Na segunda ordem, as prioridades são iguais.

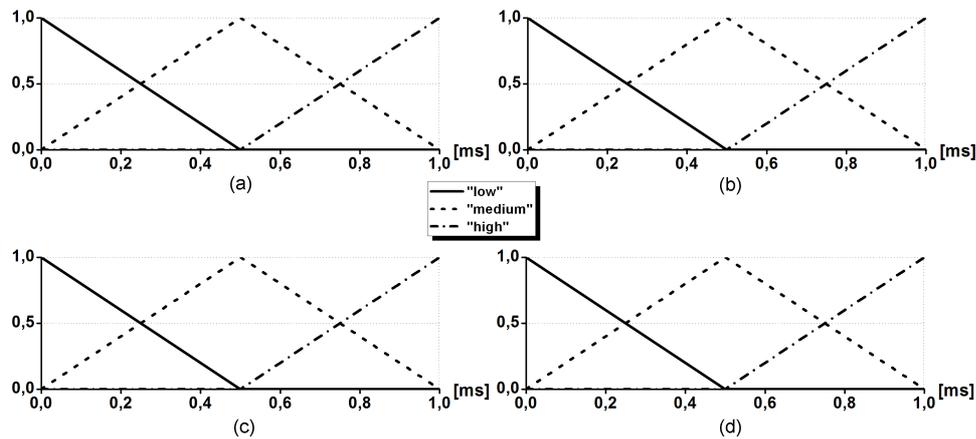


Figura 56: Prioridades:
 $(d)BW_{BE} = (c)BW_{nrtPS} = (b)BW_{rtPS} = (a)BW_{UGS}$.

Finalmente, a terceira ordem traz a classe UGS com a maior prioridade.

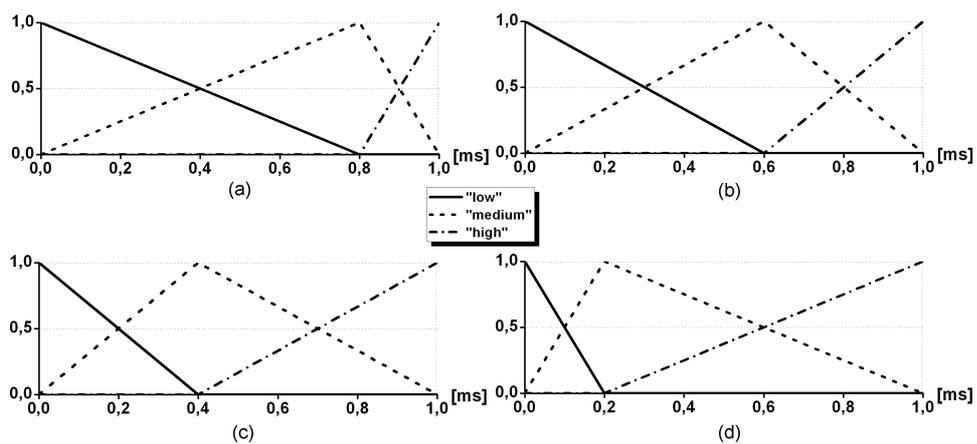


Figura 57: Prioridades:
 (a) $BW_{UGS} >$ (b) $BW_{rtPS} >$ (c) $BW_{nrtPS} >$ (d) BW_{BE} .

Apêndice D - Composição detalhada dos tráfegos de cada cenário de simulação.

D.1 Cenário 2

Tráfego UGS para SS_2

Envia 1 pacote de 608 bytes a cada quadro, durante o intervalo de atividade.

Tráfego UGS para SS_4

Envia 1 pacote de 976 bytes a cada quadro, durante o intervalo de atividade.

Tráfego BE para SS_4

Envia 1 pacote de 3168 bytes a cada quadro, durante o intervalo de atividade.

D.2 Cenário 3

Tráfego UGS para SS_2

Envia 1 pacote de 792 bytes a cada quadro, durante o intervalo de atividade.

Tráfego UGS para SS_4

Envia 1 pacote de 1320 bytes a cada quadro, durante o intervalo de atividade.

Tráfego nrtPS para SS_2

Envia, a cada 8 ± 3 quadros (valor de -3 a +3 escolhido segundo distribuição de probabilidades uniforme), 1 pacote de 4688 bytes em média, com distribuição de probabilidades normal de desvio padrão igual a $20\% \times 4688 = 938$.

Tráfego nrtPS para SS_3

Envia, a cada 10 ± 4 quadros (valor de -4 a +4 escolhido segundo distribuição de probabilidades uniforme), 1 pacote de 4102 bytes em média, com distribuição de probabilidades normal de desvio padrão igual a $20\% \times 4102 = 820$.

Tráfego nrtPS para SS_5

Envia, a cada 5 ± 2 quadros (valor de -2 a +2 escolhido segundo distribuição de probabilidades uniforme), 1 pacote de 2939 bytes em média, com distribuição de probabilidades normal de desvio padrão igual a $20\% \times 2939 = 588$.

Tráfego BE para SS_1

Envia, a cada 5 ± 2 quadros (valor de -2 a +2 escolhido segundo distribuição de probabilidades uniforme), n pacotes, $n \in \{0, 1, 2, 3, 4, 5\}$ e escolhido segundo distribuição de probabilidades uniforme. Cada pacote tem tamanho médio de 1200 bytes, com distribuição de probabilidades uniforme.

Tráfego BE para SS_4

Envia, a cada 8 ± 3 quadros (valor de -3 a +3 escolhido segundo distribuição de probabilidades uniforme), n pacotes, $n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ e escolhido segundo distribuição de probabilidades uniforme. Cada pacote tem tamanho médio de 1600 bytes, com distribuição de probabilidades uniforme.

Tráfego BE para SS_6

Envia, a cada 11 quadros, 1 pacote de 5632 bytes.

D.3 Cenário 4

Tráfego UGS para SS_1

Envia 1 pacote de 1056 bytes a cada quadro.

Tráfego UGS para SS_3

Envia 1 pacote de 528 bytes a cada quadro, durante o intervalo de atividade.

Tráfego UGS para SS_4

Envia 1 pacote de 528 bytes a cada quadro, durante o intervalo de atividade.

Tráfego rtPS para a SS_1

Envia, a cada 8 quadros, 1 pacote de 3380 bytes em média, com distribuição de probabilidades normal de desvio padrão igual a $20\% \times 3380 = 675$.

Tráfego rtPS para a SS_2

Envia, a cada 5 quadros, 1 pacote de 3168 bytes em média, com distribuição de probabilidades normal de desvio padrão igual a $20\% \times 3168 = 634$.

Tráfego nrtPS para SS_3

Envia, a cada 8 ± 3 quadros (valor de -3 a +3 escolhido segundo distribuição de probabilidades uniforme), 1 pacote de 6336 bytes em média, com distribuição de probabilidades normal de desvio padrão igual a $20\% \times 6336 = 1267$.

Tráfego nrtPS para SS_4

Envia, a cada 10 ± 4 quadros (valor de -4 a +4 escolhido segundo distribuição de probabilidades uniforme), 1 pacote de 2640 bytes em média, com distribuição de probabilidades normal de desvio padrão igual a $20\% \times 2640 = 528$.

Tráfego BE para SS_1

Envia, a cada 5 ± 2 quadros (valor de -2 a +2 escolhido segundo distribuição de probabilidades uniforme), n pacotes, $n \in \{0, 1, 2, 3, 4, 5\}$ e escolhido segundo distribuição de probabilidades uniforme. Cada pacote tem tamanho médio de 1584 bytes, com distribuição de probabilidades uniforme.

Tráfego BE para SS_2

Envia, a cada 17 quadros, com 70% de chance segundo distribuição de probabilidades uniforme, 1 pacote de 6411 bytes.

Tráfego BE para SS_3

Envia, a cada 8 ± 3 quadros (valor de -3 a +3 escolhido segundo distribuição de probabilidades uniforme), n pacotes, $n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ e escolhido segundo distribuição de probabilidades uniforme. Cada pacote tem tamanho médio de 1014 bytes, com distribuição de probabilidades uniforme.

Tráfego BE para SS_4

Envia, a cada 5 ± 2 quadros (valor de -2 a +2 escolhido segundo distribuição de probabilidades uniforme), n pacotes, $n \in \{0, 1, 2, 3, 4, 5\}$ e escolhido segundo distribuição de probabilidades uniforme. Cada pacote tem tamanho médio de 845 bytes, com distribuição de probabilidades uniforme.