

GERENCIAMENTO DE REDES EM ARQUITETURA DIFFSERV COM AUXÍLIO  
DE DATA WAREHOUSE

Gil Márcio Avelino Silva

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS  
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE  
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM  
ENGENHARIA ELÉTRICA.

Aprovada por:

---

Prof. Aloysio de Castro Pinto Pedroza, Dr.

---

Prof. Marcial Porto Fernandez, D.Sc.

---

Prof. Jorge Lopes de Souza Leão, Dr.Ing.

---

Prof(a). Luci Pirmez, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

FEVEREIRO DE 2005

SILVA, GIL MÁRCIO AVELINO

Gerenciamento de Redes em Arquitetura

DiffServ com auxílio do Data Warehouse

[Rio de Janeiro] 2005

XVII, 108 p. 29,7 cm (COPPE/UFRJ, M.Sc.,  
Engenharia Elétrica, 2005)

Tese - Universidade Federal do Rio de  
Janeiro, COPPE

1. Gerenciamento de Redes

2. Data Warehouse

I. COPPE/UFRJ II. Título (série)

## **Dedicatória:**

**À minha esposa Michele de Oliveira Ramos Silva, aos meus pais Gilson Oliveira da Silva e Selma Regina Avelino Silva, a minha irmã Márcia Regina Avelino Silva Maciel e a minha avó Aracy Oliveira da Silva.**

## **Agradecimentos:**

A Deus.

À minha esposa pelo apoio nas minhas decisões.

Aos meus pais, irmã e avó que me incentivaram.

Ao Professor Aloysio de Castro Pinto Pedroza, pelo incentivo, orientação, amizade e reconhecimento. Ao Prof. Marcial Porto Fernandez, pela ajuda, amizade e incentivo. Ao Prof. Jorge Leão, pelo auxílio. A Profa. Luci Pirmez, pela presença na banca e contribuição à tese.

Aos amigos da Embratel, Mauro, Rafael, Isac, Anderson e Nelson, pela amizade e apoio.

Aos amigos da Petrobras, Gustavo, Luis Eduardo e Luis Carlos, pela amizade e apoio.

Aos meus amigos do GTA, Fagundes, Gardel, Fabrício, Roberta, Pedro, Eric, Bicudo, Bernardo, Miguel e Guilherme, pelo apoio e amizade.

A todos os funcionários do Departamento.

Aos professores do GTA Jorge Lopes de Souza Leão, José Ferreira de Rezende, Otto Carlos Muniz Bandeira Duarte e Luís Henrique Maciel Kosmowski Costa que muito contribuíram para a conclusão deste trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## GERENCIAMENTO DE REDES EM ARQUITETURA DIFFSERV COM AUXÍLIO DE DATA WAREHOUSE

Gil Márcio Avelino Silva

Fevereiro/2005

Orientador: Aloysio de Castro Pinto Pedroza

Programa: Engenharia Elétrica

Parte dos trabalhos para garantir QoS emprega a técnica de Serviços Diferenciados (DiffServ), assumindo que a garantia por classes de serviço de forma estática proporciona qualidade de serviço. Porém, quando há agregações de fluxo isso pode falhar. Desta forma, faz-se necessário a utilização de mecanismos de provisionamento dinâmico de recursos para ajustar a rede de acordo com a variação do tráfego. Este trabalho apresenta uma metodologia para especificar um controlador fuzzy, capaz de reconfigurar os nós de um domínio de acordo com o tráfego entrante, partindo de políticas administrativas estabelecidas em contratos (SLAs). Além disso, propõe a utilização de uma ferramenta de Data Mart (DM) para fazer consultas aos dados históricos de uma empresa de telecomunicações que se encontram no Data Warehouse (DW), e assim, auxiliar no processo de tomada de decisão quanto aos dados de entrada no domínio DiffServ. A partir de então, os parâmetros do controlador fuzzy serão otimizados através dos algoritmos de Wang-Mendel e genético, para suportar da melhor forma o padrão do tráfego de entrada e assim obter a QoS desejada. As funcionalidades do modelo são demonstradas a partir de simulação de uma aplicação de Telefonia IP cruzando um domínio DiffServ.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

MANAGEMENT OF NETWORKS IN DIFFSERV FRAMEWORK WITH AID OF  
DATA WAREHOUSE

Gil Márcio Avelino Silva

February/2005

Advisor: Aloysio de Castro Pinto Pedroza

Department: Electrical Engineering

Part of the works to guarantee QoS uses Differentiated Services (DiffServ) technique, assuming that static guarantee for classrooms of service provides QoS. However, it can fail with data flow aggregation. Of this form, the use of dynamic provisioning mechanisms of resources becomes necessary to reconfigure node parameters according traffic changes. This work presents a methodology to specify a fuzzy controller, capable to reconfigure the nodes of a domain in accordance with the incoming traffic, leaving of established administrative politics in contracts (SLAs). Moreover, it proposes a Data Mart (DM) tool to make consultations the historical data of a telecommunications company that are located in Data Warehouse (DW), and then, to assist in the decision support process to choose the entrance data in the DiffServ domain. From now on, the fuzzy controller parameters will be optimized using the Wang-Mendel and the genetic algorithms, to support of the best form the standard of the entrance traffic and thus to achieve the desired QoS. The functionalities of the model are demonstrated by simulation of a IP Telephony application crossing a DiffServ domain.

# Sumário

<b>Resumo</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>Lista de Acrônimos</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xiii</b>
<b>Lista de Tabelas</b>	<b>xv</b>
<b>Lista de Algoritmos</b>	<b>xvi</b>
<b>Lista de Códigos</b>	<b>xvii</b>
<b>1. Introdução</b>	<b>1</b>
1.1. Proposta de trabalho .....	3
1.2. Trabalhos relacionados .....	3
1.3. Contribuições .....	4
1.4. Estrutura do trabalho .....	5
<b>2. Conceitos básicos de Data Warehouse</b>	<b>6</b>
2.1. O Ambiente de Data Warehouse .....	6
2.2. Tecnologia de Data Warehousing .....	7
2.3. Arquitetura do ADW .....	8
2.4. Componentes do ADW .....	9
2.4.1. Funções pessoais .....	9
2.4.2. Processos .....	11

2.4.3.	Dados .....	11
2.5.	Modelagem de Dados num ADW .....	12
2.5.1.	Visão Multidimensional .....	12
2.5.2.	Modelagem Multidimensional .....	13
2.5.3.	Modelo Estrela .....	14
2.6.	Modelagem do Data Mart (DM).....	16
2.7.	Comentários.....	17
<b>3.</b>	<b>Conceitos de Gerenciamento de Redes e Controlador Fuzzy</b>	<b>18</b>
3.1.	DiffServ .....	18
3.2.	Gerenciamento baseado em políticas .....	19
3.3.	Lógica Fuzzy .....	21
3.4.	Controlador Fuzzy .....	22
3.4.1	Interface de Fuzificação.....	23
3.4.2.	Procedimento de Inferência e base de regras.....	24
3.4.3.	Interface de Defuzificação.....	25
3.5.	Otimização com Algoritmo Genético.....	25
3.6.	Comentários.....	26
<b>4.</b>	<b>Metodologia para Gerenciamento de Redes na Internet com auxílio de DW</b>	<b>27</b>
4.1.	Arquitetura do nó DiffServ.....	28
4.2.	Proposta do trabalho .....	28
4.3.	Apresentação da arquitetura .....	29
4.4.	Apresentação da metodologia.....	31
4.5.	Especificação das políticas de QoS .....	35
4.6.	Mapeamento da especificação de políticas em lógica fuzzy .....	37
4.7.	Construção do Controlador fuzzy.....	39
4.7.1.	Controlador do Escalonador .....	39



4.7.2.	Controlador do Condicionador .....	41
4.7.3.	Algoritmo de Wang-Mendel.....	42
4.8.	Estrutura do DW .....	43
4.9.	Estrutura do DM.....	45
4.10.	Seleção dos valores para otimização do controlador fuzzy.....	49
4.11.	Otimização do controlador fuzzy através de Algoritmo Genético .....	51
4.11.1.	Funcionamento dos AGs .....	51
4.11.2.	Parâmetros de Configuração do AG.....	53
4.11.2.1.	Tamanho da população.....	53
4.11.2.2.	Taxa de cruzamento.....	54
4.11.2.3.	Taxa de mutação.....	55
4.11.2.4.	Número de gerações .....	55
4.12.	Comentários.....	56
<b>5.</b>	<b>Ambiente de Simulação e Resultados</b>	<b>57</b>
5.1.	Ambiente de simulação .....	58
5.1.1	Network Simulator - ns .....	58
5.1.2	Ponder Toolkit.....	59
5.1.3	JFS .....	59
5.1.4	Ferramenta Pilot .....	60
5.2.	Topologias de simulação .....	60
5.2.1	Topologia simples .....	61
5.2.2	Topologia complexa.....	62
5.3.	Modelo de tráfego de simulação.....	64
5.3.1	Modelo de fontes de tráfego para a topologia simples.....	64
5.3.2	Modelo de fontes de tráfego para a topologia complexa.....	68
5.4.	Medidas para avaliação .....	71

5.5.	Resultados.....	71
5.5.1	Resultados da topologia simples.....	72
5.5.1.1.	Retardo fim-a-fim da classe EF.....	73
5.5.1.2.	Descarte na classe EF.....	78
5.5.2.	Resultados na topologia complexa.....	82
5.5.2.1.	Retardo fim-a-fim da classe EF.....	83
5.5.2.2.	Descarte na classe EF.....	86
5.6.	Comentários.....	89
<b>6.</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>91</b>
	<b>Referências Bibliográficas</b>	<b>94</b>
	<b>Apêndice A - Tabelas do Modelo Estrela</b>	<b>101</b>
A.1.	Tabela de Fato.....	101
A.2.	Tabela de Dimensão.....	102
A.2.1.	Hierarquia de Dimensões.....	103
	<b>Apêndice B - Metadados</b>	<b>105</b>
	<b>Apêndice C – Procedimento de ETL</b>	<b>107</b>

## Lista de Acrônimos:

AAD:	Ambiente de Apoio a Decisão
ABD:	Administrador de Banco de Dados
AD:	Administrador de Dados
ADW:	Ambiente de Data Warehouse
AF:	Encaminhamento Assegurado ( <i>Assured Forwarding</i> )
AG:	Algoritmo Genético
BE:	Melhor escolha ( <i>Best Effort</i> )
CBR:	Taxa de Bits Constante ( <i>Constant Bit Rate</i> )
COPS:	<i>Common Open Policy Service</i>
DM:	<i>Data Mart</i>
DS:	Serviços Diferenciados ( <i>Differentiated Services</i> )
DW:	<i>Data Warehouse</i>
EF:	Encaminhamento Expresso ( <i>Expedited Forwarding</i> )
ETL:	Extração, Transformação e Carga ( <i>Extract, Transform and Load</i> )
FIFO:	<i>First In, First Out</i>
IIS:	Serviços Integrados na Internet ( <i>Internet Integrated Services</i> )
IP:	<i>Internet Protocol</i>
MIB:	<i>Management Information Base</i>
NS:	<i>Network Simulator</i>

ODS:	Armazenamento de Dados ( <i>Operational Data Storage</i> )
PHB:	Comportamento por nó ( <i>Per-Hop Behavior</i> )
QoS:	Qualidade de Serviço ( <i>Quality of Service</i> )
RTT:	<i>Round-Trip Time</i>
SGBD:	Sistema Gerenciador de Banco de Dados
SGBP:	Sistema de Gerenciamento Baseado em Políticas
SLA:	Acordo de Nível de Serviço ( <i>Service Level Agreement</i> )
SLS:	Especificação de Nível de Serviço ( <i>Service Level Specification</i> )
SNMP:	<i>Simple Network Management Protocol</i>
SO:	Sistemas Operativos
TCP:	<i>Transmission Control Protocol</i>
UDP:	<i>User Datagram Protocol</i>
WFQ:	<i>Weighted Fair Queuing</i>
WRR:	<i>Weighted Round-Robin</i>

## Lista de Figuras:

2.1: Arquitetura de um ADW .....	7
2.2: Arquitetura padrão do ADW. ....	9
2.3: Cubo do setor de vendas.....	13
2.4: Slice & Dice. ....	14
2.5: Esquema Estrela do setor de vendas.....	16
3.1: Arquitetura de Gerenciamento Baseado em Políticas. ....	20
3.2: Estrutura de um Controlador Fuzzy. ....	23
4.1: Diagrama de um nó DiffServ de borda. ....	28
4.2: Arquitetura do trabalho proposto. ....	29
4.3: Fluxograma da metodologia proposta. ....	32
4.4 Níveis da arquitetura de dados de uma empresa.....	44
4.5: Percentuais de telefonia (Acre) .....	46
4.6: Percentuais de telefonia (Estados do Brasil). ....	47
4.7 Modelo Estrela proposto.....	49
5.1: Volume de tráfego por Estado, de Setembro a Novembro de 2004. ....	61
5.2: Topologia simples (Estado de Pernambuco) .....	62
5.3: Topologia complexa (Estado de São Paulo).....	63
5.4: Quantidade de fontes de tráfego CBR (EF e BE) durante a simulação.....	65

5.5: Quantidade de fontes de tráfego On-Off (EF e BE) durante a simulação.....	67
5.6: Quantidade de fontes de tráfego CBR (EF e BE) durante a simulação.....	69
5.7: Quantidade de fontes de tráfego On-Off (EF e BE) durante a simulação.....	70
5.8: Retardo fim-a-fim da classe EF s/ o controlador.....	74
5.9: Retardo fim-a-fim da classe EF c/ controlador e Wang-Mendel.....	74
5.10: Retardo fim-a-fim da classe EF c/ controlador, Wang-Mendel e AG.....	75
5.11: Retardo fim-a-fim da classe EF s/ controlador.....	76
5.12: Retardo fim-a-fim da classe EF c/ controlador e Wang-Mendel.....	76
5.13: Retardo fim-a-fim da classe EF c/ controlador, Wang-Mendel e AG.....	77
5.14: Taxa de descarte da classe EF s/ controlador.....	78
5.15: Taxa de descarte da classe EF c/ controlador e Wang-Mendel.....	79
5.16: Taxa de descarte da classe EF c/ controlador, Wang-Mendel e AG.....	79
5.17: Taxa de descarte da classe EF s/ controlador.....	80
5.18: Taxa de descarte da classe EF c/ controlador e Wang-Mendel.....	80
5.19: Taxa de descarte da classe EF c/ controlador, Wang-Mendel e AG.....	81
5.20: Retardo fim-a-fim da classe EF s/ controlador.....	83
5.21: Retardo fim-a-fim da classe EF c/ controlador.....	84
5.22: Retardo fim-a-fim da classe EF s/ controlador.....	85
5.23: Retardo fim-a-fim da classe EF c/ controlador.....	85
5.24: Taxa de descarte da classe EF s/ controlador.....	87
5.25: Taxa de descarte da classe EF c/ controlador, Wang-Mendel e AG.....	87
5.26: Taxa de descarte da classe EF s/ controlador.....	88
5.27: Taxa de descarte da classe EF c/ controlador, Wang-Mendel e AG.....	88
A.1: Hierarquia de Tempo.....	103

## **Lista de Tabelas:**

4.1 Parâmetros utilizados no algoritmo genético.....	53
5.1: Relação dos segundos com horas do dia .....	66
5.2: Relação dos minutos com horas do dia .....	68
5.3: Retardo e variação de retardo da classe EF na topologia simples (ms).....	77
5.4: Taxa de descarte de pacotes na topologia simples .....	82
5.5: Retardo e variação de retardo da classe EF na topologia complexa (ms) .....	86
5.6: Taxa de descarte de pacotes na topologia complexa.....	89

# Lista de Algoritmos:

4.1: Algoritmo Wang-Mendel .....	42
----------------------------------	----



## **Lista de Códigos:**

4.1: Trecho exemplo de especificação Ponder do escalonador .....	36
4.2: Mapeamento do comando constraint.....	38
4.3: Mapeamento dos comandos event e oblig.....	38
4.4: Código JFS mapeado a partir do comando oblig.....	39

# Capítulo 1

## Introdução

As atuais redes de telefonia IP não oferecem recursos adequados para aplicações multimídia (Internet), devido a dificuldade do provimento de recursos demandados pelas redes que são aleatórios e imprevisíveis. A Diferenciação de Serviço (DiffServ) [1] surge como uma possível solução para este problema, pois segundo sua filosofia, os serviços podem ser providos de forma escalável a partir de diferentes classes de serviço. Soluções como a Integração de Serviços (IntServ), visam prover recursos fim a fim assegurando qualidade de serviço (QoS) para cada conexão, porém apresentam problemas de escalabilidade. Além destas, existem soluções baseadas em Engenharia de Tráfego que visam direcionar o fluxo de dados por um “melhor caminho”.

Basicamente existem duas técnicas que podem ser empregadas na tentativa de prover recursos com QoS para tráfegos na Internet, os quais apresentam característica aleatória com relação ao fluxo entrante. A primeira delas é o superprovisionamento de recursos da rede. Esta técnica embora possua como vantagem a facilidade de implantação, tem como grande desvantagem o seu elevado custo. A segunda técnica é o provisionamento dinâmico de recursos. Esta técnica possui uma complexidade adicional pois configura os recursos dos nós em uma rede dinamicamente, de acordo com aleatoriedade do fluxo entrante. A grande vantagem desta técnica está na utilização de canais de comunicação compatíveis com a demanda.

Tendo em vista a complexidade de gestão dos mecanismos de provisionamento dinâmico, grande parte das empresas de telecomunicação tem preferido o

superprovisionamento de recursos das redes para prover QoS. Porém, isto faz com que os custos relacionados à infra-estrutura sejam altos, pois durante grande parte do tempo as redes possuirão recursos ociosos. Além disso, tem-se como problema o planejamento do crescimento das redes, ou seja, deve-se construir uma infra-estrutura que leve em consideração estimativas de tráfegos futuros que normalmente é imprecisa.

No provisionamento dinâmico, o gerenciamento de recursos baseado em políticas tem se apresentado como uma técnica eficiente para coordenar a configuração de uma rede para obter a QoS desejada nas arquiteturas DiffServ e IntServ [2]. Apesar de não garantir a QoS em vários domínios, pode-se prover garantias de QoS no interior de um domínio DiffServ.

Atualmente, grande parte, senão todas as empresas de telecomunicações possuem um Data Warehouse (DW), normalmente usado por setores voltados para os negócios da companhia. Os DWs são grandemente empregados nas áreas de marketing, vendas, finanças, *e-commerce* [3, 4] entre outras. Segundo W. H. Inmon [55], um DW é uma coleção de dados orientada por assuntos, integrada, variante no tempo e não volátil que tem por finalidade dar suporte aos processos de tomada de decisão.

Apesar do DW apresentar-se útil no suporte à decisão em projetos de gerenciamento e planejamento de redes, não encontramos pesquisas integrando essas duas tecnologias. Isto ocorre devido a filosofia de utilização do DW, que tem como objetivo fornecer visões de tendências dos negócios de uma empresa baseado em dados históricos. Esta base de dados de uma empresa de telecomunicações possui um grande potencial, o qual pode ser usado como fonte provedora de dados para ferramentas de Data Mart (DM), que consiste em um repositório de dados menor que o DW e armazena dados relevantes para uma área específica de uma companhia, com o propósito de fornecer visões de tendências de tráfego. Desta forma, uma apresentação dos dados históricos trabalhados como visões de tráfego pode auxiliar consideravelmente no processo de tomada de decisão quanto aos dados que serão utilizadas no planejamento de uma rede que visa oferecer garantias de QoS.

### 1.1 Proposta de trabalho

Este trabalho apresenta uma proposta de inclusão de uma estrutura de suporte a decisão provida a partir de dados históricos armazenados no DW, os quais serão utilizados para medições de tendências de tráfego, em uma arquitetura de redes proposta por Fernandez [5] que visa garantir o controle de QoS dentro de um domínio DiffServ. Nesta arquitetura foi utilizado um controlador fuzzy que reconfigura dinamicamente os nós do domínio através do gerenciamento baseado em políticas. Foi aplicada lógica fuzzy no controlador em função da imprevisibilidade do tráfego de entrada. Desta forma, a utilização do controlador fuzzy se justifica pela ausência de um modelo matemático preciso para tratar estimativas de tráfego [6].

Embora os DWs sejam geralmente utilizados por setores relacionados a negócios, seus dados históricos armazenados podem auxiliar de forma expressiva nas tomadas de decisão relacionadas à configuração das redes para provimento de QoS. O trabalho proposto se divide em duas etapas. Na primeira etapa são realizadas consultas as tabelas do DW, com o objetivo de auxiliar o Gerente de Planejamento de Redes na tomada de decisão quanto aos dados a serem usados na configuração da rede. A segunda etapa consiste na configuração da rede propriamente dita, onde a rede será preparada para suportar da melhor forma o tráfego definido na primeira etapa.

Por fim, com o intuito de mostrar a viabilidade da arquitetura proposta, foram realizadas simulações em protótipos com dados de entrada típicos de uma Empresa de Telecomunicações que presta serviço para o mercado nacional. Onde foram avaliados alguns indicadores fundamentais para provisão de QoS, como: tempo de retardo, variação do retardo e descarte do fluxo de Telefonia IP.

### 1.2 Trabalhos relacionados

Na literatura, foram encontrados vários projetos de mecanismos de provisionamento dinâmico para melhoria da QoS. Como a proposta de Liao e Campbell [7, 8], que visa ajustar dinamicamente o escalonador de pacotes dos nós a partir de

medidas de descarte de pacotes. Fernandez [5] apresenta uma proposta que consiste no controle de QoS dentro de um único domínio DiffServ, onde a partir de um controlador fuzzy os nós da rede são reconfigurados dinamicamente através de um mecanismo de gerenciamento de políticas.

Também foram encontrados diferentes ramos de atuação para aplicações de DW, como em Ferreira [9] que propõe, a partir de uma estrutura de suporte a decisão, um modelo de integração capaz de mapear os conceitos relativos às análises dimensional e espacial. Moura [10] apresenta uma proposta de um projeto que tem por objetivo o estudo de Sistemas de Suporte a Decisão em ambientes de DW não convencionais, especificamente aqueles voltados para o suporte à análise de fenômenos em geral. Neste trabalho é apresentado uma modelagem dos ambientes operacional e analítico de banco de dados, para apoiar os estudos de interpretação de biofenômenos resultantes da ação de microorganismos presentes no processo de produção de petróleo.

Embora existam diferentes propostas utilizando funcionalidades de DW, não foram encontradas propostas para configuração de rede partindo de dados históricos providos por um DW para controle de QoS.

### 1.3 Contribuições

Este trabalho tem como principal contribuição, o desenvolvimento de um modelo de suporte a tomada de decisão quanto aos dados de entrada na arquitetura proposta por Fernandez [15, 16, 17], dados estes que serão de expressiva importância no processo de reconfiguração da rede para prover QoS para um padrão de tráfego pré-determinado. A proposta de Fernandez [15, 16, 17] consiste numa metodologia de um mecanismo de provisionamento dinâmico baseado em lógica fuzzy, incluindo técnicas de otimização baseadas em Algoritmos Genéticos (AG), comprovadamente eficiente para melhorar a QoS em uma arquitetura DiffServ com alta variação no tráfego de dados.

Neste trabalho foi construído um ambiente de processamento analítico específico para áreas de planejamento e gerenciamento de redes, ou seja, baseado num Ambiente

de DW foi construído um Data Mart (DM) específico para gerenciamento de redes. Kimball [18, 19] defende a idéia de que à medida que os analistas vão percebendo a potencialidade deste tipo de processamento, DMs podem ser criados ou alterados e inseridos num ambiente de DW.

### **1.4 Estrutura do Trabalho**

Esta dissertação encontra-se organizada em 6 capítulos. O capítulo 2 apresenta o Ambiente de Data Warehouse (ADW), a tecnologia de Data Warehousing, a arquitetura, os componentes e a modelagem de dados de um ADW e a modelagem do DM. No capítulo 3, são descritos um resumo sobre DiffServ, gerenciamento baseado em políticas, lógica fuzzy e Algoritmos Genéticos. O capítulo 4, por sua vez, apresenta a metodologia proposta para o ambiente de integração de um modelo de suporte a tomada de decisão para gerenciamento e planejamento de redes com QoS para a arquitetura DiffServ. O ambiente de simulação, as topologias e os modelos de tráfego aplicados nas simulações e os resultados obtidos são mostrados no capítulo 5. O capítulo 6 apresenta as conclusões deste trabalho e sugere temas para trabalhos futuros.

## **Capítulo 2**

### **Conceitos básicos de Data Warehouse**

A complexidade do gerenciamento de informações das grandes empresas resultou em necessidade de mudanças no sistema de manipulação dos dados, as quais foram atendidas com a tecnologia de Data Warehousing. Esta tecnologia é centrada na figura do Data Warehouse (DW), uma base de dados capaz de manipular um grande volume de informações com bom desempenho. A abordagem de DW resultou em um ambiente conhecido como Ambiente de Data Warehouse (ADW) [20], o qual permite melhorar a gerência, o controle e o acesso aos dados.

Esse capítulo está organizado em 6 seções. Na seção 2.1, é apresentado o ADW; na seção 2.2, é mostrado a tecnologia de Data Warehousing. Nas seções 2.3, 2.4 e 2.5 são apresentados a arquitetura, os componentes e a modelagem de dados do ADW. Por fim, na seção 2.6, é descrito a modelagem do Data Mart (DM).

#### **2.1 O Ambiente de Data Warehouse**

A tecnologia de Data Warehousing é considerada a evolução natural do Ambiente de Apoio à Decisão (AAD) [21]. Sua rápida absorção pelas empresas está relacionada à necessidade do domínio de informações para garantir respostas e ações rápidas, assegurando a competitividade de mercado. Dentre os fatores que contribuíram para essa absorção, merecem destaque: os avanços tecnológicos, as mudanças

organizacionais e estruturais nos negócios, a abertura de mercados e a globalização da economia. Com o advento da tecnologia de Data Warehousing, os Ambientes de Apoio à Decisão passaram a ser denominados Ambientes de Data Warehouse (ADW). Este novo ambiente contém como repositórios principais o DW e os Data Marts (DMs), os quais são melhor explicados na seção 2.4.3. A Figura 2.1 apresenta a arquitetura usual do ADW.

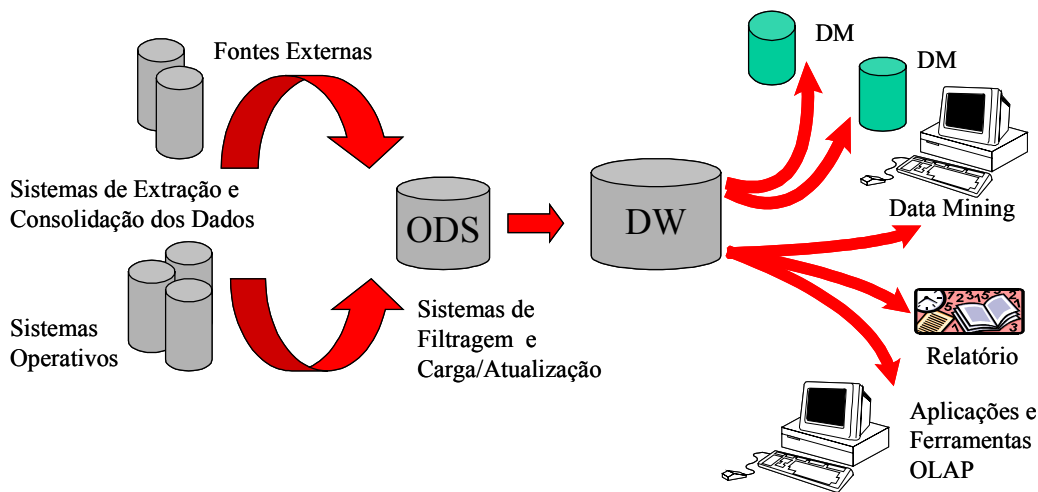


Figura 2.1: Arquitetura de um ADW

Nesta arquitetura do ADW os dados de entrada são providos a partir de fontes externas e sistemas operativos, onde são passados por processos de extração e consolidação dos dados. A partir de então esses dados são inseridos no armazenador de dados operativos (ODS), que funciona como um banco de dados intermediário entre o ambiente operacional e o DW. Antes de chegar ao DW, os dados sofrem processos de filtragem e limpeza. Uma vez no DW, os dados podem ser acessados livremente para geração de relatórios, carga de DMs, aplicações OLAP, “*data mining*” entre outras.

## 2.2 Tecnologia de Data Warehousing

A tecnologia de Data Warehousing tem o propósito de capacitar as empresas a acessarem seus dados de forma rápida e fácil, proporcionando um apoio aos tomadores



de decisão. Nesse contexto, o Data Warehousing proporciona ao AAD uma sólida e concisa integração dos dados da empresa para a realização de análises gerenciais. Ele habilita sistemas capazes de integrar e consolidar as informações de fontes heterogêneas e fontes externas, sumarizando, filtrando e limpando estes dados, preparando-os para análise e suporte à decisão. Essa tecnologia originou o ADW, que possui um conjunto de características, conforme apresentado em seguida, que o distingue de outros ambientes de sistemas convencionais [22].

- Extração de dados de fontes heterogêneas (existentes ou externas);
- Transformação e integração dos dados antes de sua carga;
- Utilização de máquina e suporte próprio;
- Visualização dos dados em diferentes níveis. Os dados do DW podem ou não ser extraídos para um nível mais específico, os DMs, e a partir deste para um banco de dados individual;
- Utilização de ferramentas voltadas ao acesso de informações com diferentes níveis de apresentação;
- Utilização de informações históricas constituídas de dados não voláteis.

### 2.3 Arquitetura do ADW

A arquitetura padrão do ADW, ilustrada na Figura 2.2, inclui, além da estrutura de dados, mecanismos de comunicação, processamento e apresentação da informação para o usuário final. De uma forma geral, as arquiteturas orientadas a este ambiente são constituídas por um conjunto de ferramentas que respondem desde a carga até ao processamento de consultas em repositórios de dados, como o DW e DM.

As ferramentas existentes podem ser divididas em dois grupos. As relacionadas à carga inicial e às atualizações periódicas do DW. As ferramentas do primeiro grupo são responsáveis pela extração dos dados de múltiplos sistemas operativos e fontes externas, e pela limpeza, transformação e integração dos dados. As do segundo grupo estão relacionadas às consultas, orientadas para os usuários finais, que são responsáveis pela

elaboração de relatórios, pesquisas informativas, análise e "*data mining*".

Quanto aos repositórios, o DW funciona como um grande centralizador dos dados, enquanto os DMs permitem uma visão mais direcionada de um problema, funcionando como repositórios menores, orientados a áreas específicas. A seguir são apresentados os principais componentes da arquitetura para este ambiente.

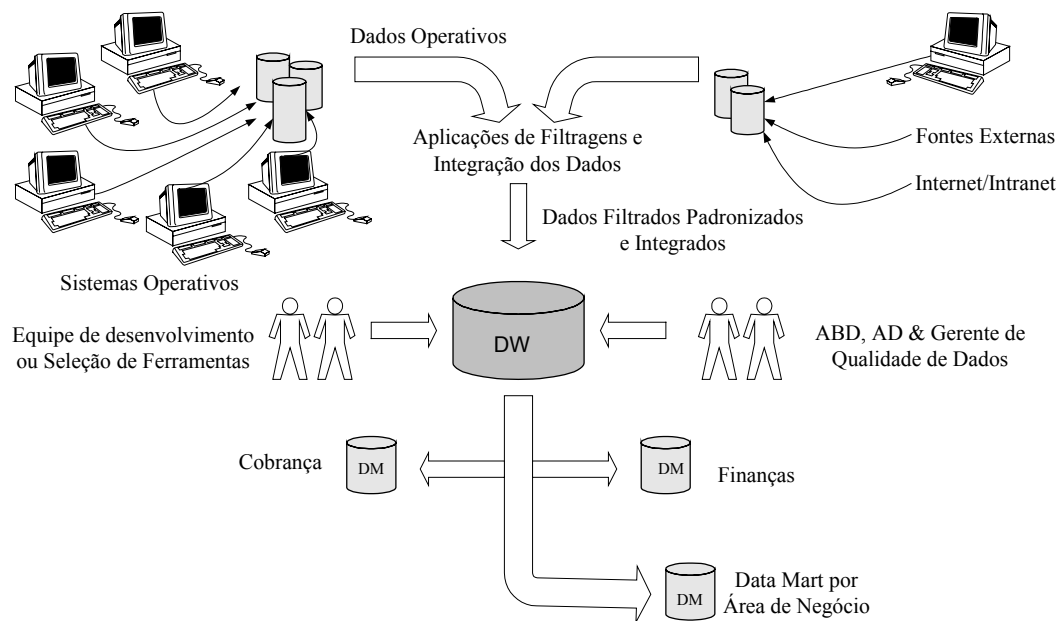


Figura 2.2: Arquitetura padrão do ADW.

## 2.4 Componentes do ADW

Nos tópicos a seguir, são apresentados os componentes do ADW relacionados a aspectos com as funções pessoais, os processos de utilização do ADW e os dados, principalmente quanto a sua forma de armazenamento.

### 2.4.1 Funções pessoais

Por sua abrangência, este ambiente envolve desde profissionais de

processamento de dados até analistas de negócios considerados como "usuários" do ambiente. Em outras palavras, este ambiente inclui: os administradores do projeto; os projetistas do banco; os Administradores de Bancos de Dados (ABD) dos sistemas operativos; os programadores e os analistas dos sistemas de conversão e dos aplicativos que acessarão as informações; e os usuários finais. Esta diversidade de "usuários" implica em uma preocupação maior do que a observada nos sistemas operativos. Nesses, os analistas e projetistas dos sistemas apenas atendiam ao grupo de usuários que efetivamente utilizariam o sistema. No ADW, estes "usuários" encontram-se agrupados por papéis como são descritos a seguir.

- Usuários responsáveis pela carga dos dados - são os programadores que necessitam conhecer o mapeamento entre o DW e os sistemas operativos, além de todos os requisitos necessários à filtragem e à integração dos dados.
- Usuários finais - são os especialistas, gerentes, executivos e analistas de negócio, que utilizam a informação para apoio à tomada de decisão. Estes usuários apresentam uma grande familiaridade com os termos do negócio e estão sempre em busca da solução de um problema ou de novas oportunidades. Estes usuários podem ser divididos em dois grupos: os usuários diretos e os usuários indiretos. Os usuários diretos são aqueles que acessam livremente o DW enquanto os indiretos acessam os DMs especializados [23].
- Usuários responsáveis pelo desenvolvimento e manutenção do DW e dos DMs - equivalem aos Administradores de Bancos de Dados (ABD) e Administradores de Dados (AD) dos Sistemas Gerenciadores de Bancos de Dados (SGBD) dos sistemas operativos. Estabelecem o nível de preocupação com os metadados (o conceito de metadados é apresentado no apêndice B), com a arquitetura de armazenamento e com a estrutura dos dados, visando, principalmente, melhorar o desempenho das consultas. É comum o estabelecimento de equipes diferentes. Para estes casos, um maior grau de rigidez se faz necessário na elaboração dos DM, evitando inconsistências entre as estruturas dos DM e do DW.

### 2.4.2 Processos

Os processos do ADW consistem na extração dos dados das fontes operativas, na organização e integração destes dados de forma consistente para o DW, e no acesso aos dados integrados de modo eficiente e flexível, tanto para consulta direta como para acesso por DM [24].

A extração, organização e integração dos dados devem ser realizadas com o propósito de garantir consistência e integridade das informações. Normalmente, faz-se necessário o desenvolvimento de sistemas ou avaliação de ferramentas para extração de dados e atualização do DW. Estes sistemas/aplicações são responsáveis pela filtragem, limpeza, sumarização e concentração dos dados espalhados pelas fontes externas e pelos sistemas operativos. Sua elaboração requer, dos analistas envolvidos, um razoável conhecimento tanto das bases de dados de onde as informações serão extraídas como da base onde serão armazenadas. No ADW, as ferramentas devem permitir um acesso intuitivo aos dados, possibilitando a análise daqueles mais significativos.

### 2.4.3 Dados

Neste ambiente os dados podem ser armazenados em diferentes níveis de agregação, que são: dados detalhados, configurando o nível operativo; dados levemente sumarizados; e dados altamente sumarizados [23]. Os dados encontram-se armazenados em repositórios que constituem uma das maiores preocupações deste ambiente. O emprego ou não de qualquer dos repositórios apresentados a seguir depende, exclusivamente, da arquitetura a ser adotada pela empresa. Assim, um ADW pode apresentar os repositórios de dados descritos a seguir.

- "Operational Data Storage" (ODS) - representa um armazenamento intermediário de dados, facilitando a integração dos dados do ambiente operativo antes da sua atualização no DW, ou seja, o ODS consiste num repositório temporário, que armazenava apenas as informações correntes, antes de serem carregadas para o DW.

- Data Warehouse (DW) - é a espinha dorsal deste ambiente. Ele representa uma grande base de dados capaz de integrar, de forma concisa e confiável, as informações de interesse para a empresa, que se encontram espalhadas pelos sistemas operativos e pelas fontes externas, para posterior utilização no suporte a decisão.
- Data Mart (DM) - representa um subconjunto de dados do DW. Permite acesso descentralizado e atualmente serve de fonte para os dados que compõem os bancos de dados individuais. Os dados do DM são direcionados a um departamento ou uma área específica do negócio. O DM, normalmente, é modelado em um esquema estrela (esquema apresentado na seção 2.5.3), de acordo com as necessidades específicas do usuário final [25]. Uma das principais vantagens de seu emprego é a possibilidade de retorno rápido, garantindo um maior envolvimento do usuário final, capaz de avaliar os benefícios extraídos de seu investimento.

## 2.5 Modelagem de Dados num ADW

A modelagem de dados consiste em um modo eficiente de compreender os dados, cujo objetivo é mapear aspectos do mundo real para um mundo particular. Segundo Batini em [39], um modelo de dados é uma coleção de conceitos que podem ser utilizados para descrever um conjunto de dados juntamente com as operações para sua manipulação. Mostramos a seguir conceitos sobre a Visão Multidimensional, a Modelagem Multidimensional e o Modelo Estrela.

### 2.5.1 Visão Multidimensional

Um modelo tridimensional pode ser representado por meio de um cubo. Entretanto, um modelo multidimensional, por normalmente possuir mais de três dimensões é definido por um hipercubo onde cada face representa uma dimensão. Geralmente, os usuários não trabalham com informações de uma face ou dimensão do

hipercubo específica, mas com o cruzamento delas. É apresentado na Figura 2.3 um modelo de um cubo dimensional para o setor de vendas de uma empresa.

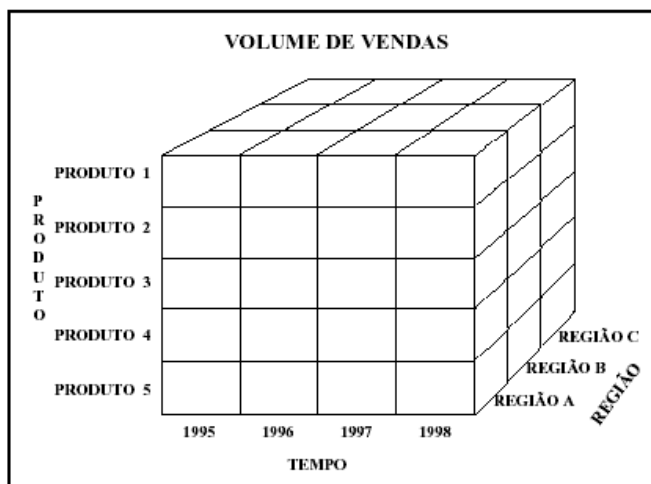


Figura 2.3: Cubo do setor de vendas

### 2.5.2 Modelagem Multidimensional

O principal objetivo deste modelo é a elaboração de um banco de dados consistente que forneça uma visão multidimensional ao usuário final do sistema. O modelo dimensional deve ser de fácil entendimento e interpretação pelos seus desenvolvedores e usuários finais. A modelagem independe do SGBD onde será implementada, ou seja, pode ser implementada num SGBD relacional ou multidimensional.

A modelagem multidimensional simplifica as operações realizadas pelas ferramentas de ADW. Em seguida, são descritas algumas operações como exemplo, as quais são detalhadas em Machado [40].

1. Drill-Down – é a capacidade que o sistema fornece ao usuário de poder navegar de um nível mais alto para um nível mais baixo de detalhe do dado.
2. Roll Up ou Drill Up – é a capacidade de percorrer o caminho inverso do “Drill-Down”, ou seja, permite ao usuário caminhar de um nível mais detalhado para

um nível menos detalhado.

3. Slice-Dice – é a operação de fatiar o cubo de informações onde o usuário pode realizar consultas que combinem e/ou separem dados. Nesta operação é possível fazer consultas desconsiderando algumas dimensões. Enfim, pode-se criar várias visões de consulta. A Figura 2.4 mostra um exemplo deste tipo de operação.

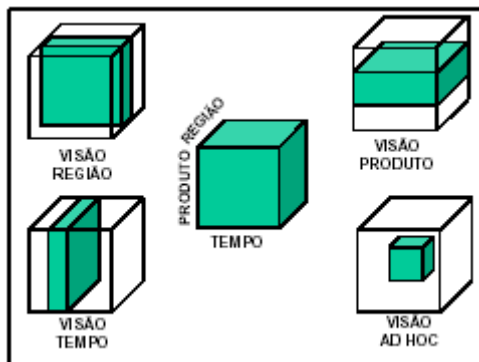


Figura 2.4: Slice & Dice

4. Pivoteamento – permite a reorganização das dimensões para novas visões de consultas.
5. Consultas ad-hoc – são consultas de acesso casual, geralmente executado de forma heurística. Deste modo, o usuário tem a possibilidade de gerar consultas de acordo com suas necessidades de cruzar as informações, para encontrar as respostas procuradas.

### 2.5.3 Modelo Estrela

Este modelo, também chamado de esquema estrela ou “*Star Schema*”, é conhecido por este nome por apresentar a tabela de fatos "dominante" no centro do modelo e as tabelas de dimensões nas extremidades. Conceitos sobre tabelas de fato e de dimensão são apresentados no apêndice A (Tabelas do Modelo Estrela). A tabela de fatos é ligada às demais tabelas por várias junções, uma junção para cada tabela de dimensão.

O esquema estrela possui uma estrutura simples, ideal para aplicações com

poucas tabelas e junções bem definidas. Segundo Poe [22], este esquema apresenta dificuldades se utilizado para aplicações com dimensões grandes ou com elevado número de dimensões. De forma geral, a utilização do modelo estrela proporciona algumas vantagens [22], dentre as quais pode-se listar:

1. facilidade de leitura e entendimento, principalmente para os usuários;
2. um projeto de banco de dados com uma visão mais próxima do usuário final;
3. criação de um banco de dados que permite consultas rápidas, realizadas de modo eficiente e fácil para o usuário final;
4. entendimento e “navegação” dos metadados pelos desenvolvedores e usuários finais;
5. disponibilidade de várias ferramentas de *frond-end*, desenvolvidas com o intuito de atender a este modelo.

A aplicação do modelo estrela pode representar tanto o modelo lógico como o modelo físico de um banco de dados [41]. Um modelo dimensional pode ser representado por uma ou mais tabelas de fato relacionadas com tabelas de dimensão. Entretanto, com o intuito de simplificação do modelo adota-se a visão de um esquema por vez, isto é, a representação mais simples de um modelo dimensional consiste em um modelo estrela com uma tabela central de fatos relacionada com tabelas de dimensões. A Figura 2.5 ilustra a representação de um esquema estrela relacionado ao setor de vendas de uma empresa.



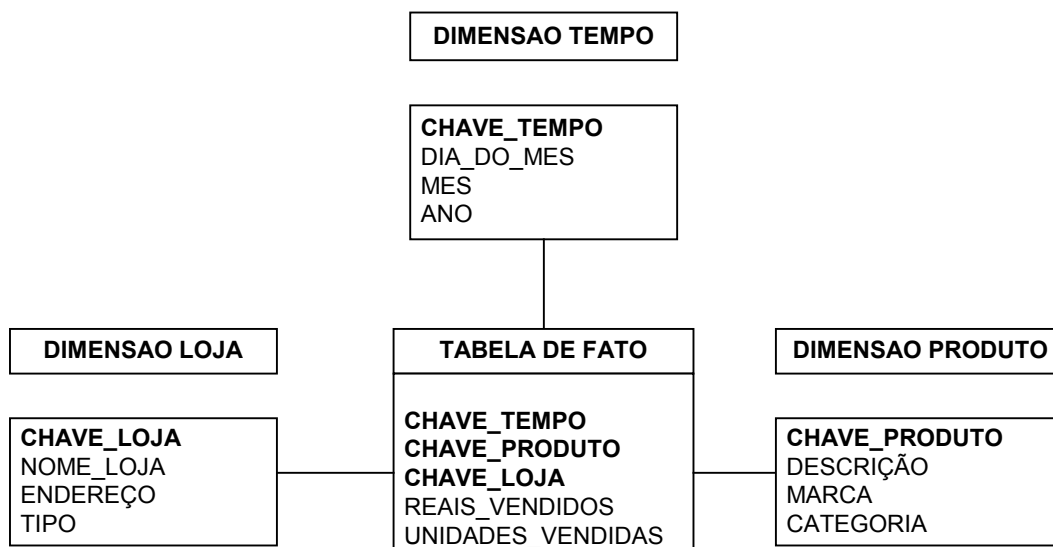


Figura 2.5: Esquema Estrela do setor de vendas

## 2.6 Modelagem do Data Mart (DM)

Os DMs, conforme mostrado na seção 2.4.3, são criados para atender a necessidade de um departamento específico. Assim, constrói-se diferentes DMs para diferentes setores de uma empresa a fim de atender seus objetivos específicos [45]. Costuma-se dar pouca atenção às diferenças entre os objetivos de projeto para um DM e os objetivos da empresa, que são representados pelo DW de uma forma generalizada. Para modelar bem um DM é necessário uma boa definição do seu escopo. Pode-se obter esta definição descobrindo, dentre outros, os problemas e as necessidades que o DM deve atender, as principais formas de utilização das informações pelos seus diferentes usuários finais e os benefícios que a empresa espera obter com sua implantação [46].

São listados em seguida, segundo [24, 44, 45], os principais requisitos que devem ser atendidos por um DM.

1. fornecer acesso aos dados de uma área de interesse.
2. garantir a consistência dos dados.
3. garantir a facilidade de emprego dos dados pelos usuários. A manipulação dos dados deve ser a mais intuitiva possível.

4. permitir as consultas padrão de um modelo multidimensional, como “Slice & Dice”, “Drill down” e “Drill up”.
5. ser compatível com as ferramentas para consulta, análise e apresentação de informações.

Basicamente existem duas propostas para construção de DMs, que são: a modelagem a partir do DW e a modelagem a partir dos sistemas operativos. A modelagem de dados a partir do DW garante a consistência e o mapeamento dos dados. Já a modelagem a partir de sistemas operativos pode gerar DMs independentes, ou seja, modelos sem uma estrutura padronizada de metadados, podendo assim gerar inconsistências. Na modelagem a partir de sistemas operativos deve-se adotar uma rígida política de controle e gerência de metadados, a fim de que sejam preservadas a integridade e consistência dos dados na criação dos DMs.

### **2.7 Comentários**

Neste capítulo foi descrito o ADW, utilizado em nossa proposta com o propósito de oferecer auxílio na tomada de decisão quanto aos dados a serem utilizados no planejamento das redes para provisão de QoS. Foram mostrados a estrutura do ADW, a tecnologia de Data Warehousing, a arquitetura do ADW, os componentes do ambiente, a modelagem de dados num ADW e a modelagem do DM.

A partir desta visão geral dos conceitos básicos de Data Warehouse, foi possível compreender a forma e o contexto de utilização do ADW. A seguir, será apresentado um resumo sobre os conceitos utilizados no trabalho de Fernandez [5] que serão utilizados nesta proposta.

## Capítulo 3

# Conceitos de Gerenciamento de Redes e Controlador Fuzzy

Neste capítulo são mostrados resumidamente alguns conceitos das tecnologias para Gerenciamento de Redes empregadas neste trabalho. É apresentada a Diferenciação de Serviços (DiffServ) na seção 3.1. Na seção 3.2 é introduzido o Gerenciamento baseado em Políticas. Já na seção 3.3, é descrita uma introdução sobre lógica fuzzy, enquanto na seção 3.4 é mostrado o controlador fuzzy. Por fim, na seção 3.5, é apresentada a otimização do controlador fuzzy por AG.

### 3.1 DiffServ

Os serviços providos pela Internet, rede por comutação de pacotes, não foram criados com o propósito de oferecer garantias temporais aos usuários, como: retardo e variação de retardo (*jitter*), estes tinham como objetivo oferecer apenas uma comunicação confiável. O crescimento da Internet despertou o interesse de criar mecanismos que possibilitassem oferecer recursos para permitir aplicações sensíveis ao tempo, em particular as aplicações multimídia.

Como proposta para resolver o problema de provisionamento de recursos nas redes, surge a Integração de Serviços (IntServ) que possibilita uma reserva de banda

passante e espaço em *buffer* para determinados usuários ou processos privilegiados ao longo do caminho, garantindo assim a QoS desejada. Esta solução, embora atenda as expectativas para o tráfego prioritário, apresenta grande problema de escalabilidade.

A Diferenciação de Serviços (DiffServ) [26, 27, 28] possibilita o oferecimento de serviços de forma diferenciada aos recursos de uma rede (Internet) sem problema de escalabilidade. Ela objetiva oferecer QoS diferenciada para diferentes fluxos de dados os quais são agregados em classes de serviços que possuem um padrão de QoS especificado.

A Arquitetura DiffServ utilizada neste trabalho, proposta por Fernandez [11, 12, 13, 14], possui como principal objetivo, melhorar o desempenho da rede mantendo a escalabilidade do sistema. Desta forma, foram utilizados dois tipos de roteadores (nós da rede):

1. Roteadores de borda: possuem a função de fazer a comunicação com os demais nós da rede, internos e externos ao domínio DiffServ. Eles implementam a complexidade de classificação, suavização e policiamento do tráfego entrante no domínio.
2. Roteadores de núcleo: nesses nós são realizadas as operações de classificação, escalonamento e gerenciamento de filas.

## 3.2 Gerenciamento baseado em políticas

O Gerenciamento baseado em políticas tem se mostrado como uma técnica eficiente para coordenar a configuração de uma rede com QoS. Sloman [29] descreve uma metodologia para especificar parâmetros de comportamento a serem cumpridos por cada elemento da rede da melhor maneira possível, levando-se em consideração as características particulares de cada elemento. Em [30], uma política é definida como *uma regra que direciona as opções de comportamento de um sistema de gerenciamento*.

Apresentamos na Figura 3.1, a arquitetura de um sistema de gerenciamento baseado em políticas proposto pelo IETF [31, 32]. Seus elementos possuem as seguintes funções:

- O PMT (*Policy Management Tool*) é uma ferramenta de gerenciamento de políticas que fica diretamente em contato com o operador de redes, que especifica os requisitos de serviço.
- O PR (*Policy Repository*) é o repositório (Banco de Dados), onde as políticas são armazenadas.
- O PDP (*Policy Decision Point*) é o ponto de decisão, local onde é feita a interpretação das políticas, e onde é feita a escolha da configuração dos equipamentos.
- O PIB (*Policy Information Base*) é a tabela que armazena todas as políticas do sistema.
- O PEP (*Policy Enforcement Point*) é o ponto no qual se faz cumprir (executa) as ações de configuração de rede determinadas pelo PDP.
- O LPDP (*Local Policy Decision Point*) é um componente, que contém seu próprio PDP e PIB, utilizado localmente na aplicação caso haja problemas com o PDP.



Figura 3.1: Arquitetura de Gerenciamento Baseado em Políticas

O elemento de destaque nesta estrutura de gerenciamento baseado em políticas é o PDP, pois nele está concentrada a responsabilidade pela escolha da configuração dos elementos da rede, ou seja, é responsável por grande parte do processamento de controle do sistema. O controlador fuzzy que usaremos, se baseia em políticas que realiza as funções do PDP/PEP na arquitetura de gerenciamento baseado em políticas.

### 3.3 Lógica Fuzzy

A lógica fuzzy foi desenvolvida com o objetivo de tratar de forma eficiente valores imprecisos. O aspecto mais notável dessa metodologia está na possibilidade de capturar, num modelo matemático, conceitos intuitivos como graus de satisfação, conforto, adequação e etc [33]. As teorias mais conhecidas para tratar do problema de imprecisão são a teoria dos conjuntos e a teoria da probabilidade. Embora muito utilizadas, essas teorias nem sempre conseguem captar a riqueza das informações fornecidas pelos seres humanos. A teoria dos conjuntos não é capaz de tratar o aspecto vago da informação e a teoria da probabilidade é mais adequada para tratar de informações relacionadas a freqüências com que os eventos ocorrem.

A teoria dos conjuntos nebulosos (lógica fuzzy) foi desenvolvida a partir de 1965 por Lofti Zadeh [34], sua principal característica consiste em poder representar uma variável dentro de um intervalo do conjunto representado por um valor  $x$  de pertinência. Desta forma, um elemento pode pertencer a um conjunto com um grau de pertinência entre 0 e 1, ou seja, diferentemente da Teoria de Conjuntos convencional onde um elemento pertence ou não pertence ao conjunto, na lógica fuzzy um elemento pode pertencer a um conjunto com grau de pertinência de 70% e não pertencer ao conjunto com grau de pertinência de 30%. Quando um elemento pertence completamente a um conjunto dizemos que ele pertence ao conjunto com grau de pertinência 1 ou 100% e não pertence ao conjunto com grau de pertinência 0 ou 0%. A função utilizada para o cálculo do grau de pertinência dos elementos é conhecida como função de pertinência, esta associa para todo o elemento do discurso um valor do intervalo  $[0,1]$ .

Uma característica importante da lógica fuzzy é o fato dela utilizar variáveis lingüísticas ao invés de variáveis numéricas. Assim, quando tratamos de sensação térmica por exemplo, podemos utilizar valores do tipo: “quente”, “frio”, “pouco quente”, “muito quente”, “pouco frio”, “muito frio” e etc, que são representados por conjuntos fuzzy. A idéia é de apresentar as variáveis e os sistemas de controle de forma semelhante a um operador humano. Desta forma, poderíamos utilizar a seguinte regra para o controle de temperatura: “Se está quente e a temperatura está aumentando

lentamente, então aumente um pouco o resfriamento”. Esta regra pode ser escrita em variáveis lingüísticas da seguinte maneira: “Se temp = MQ e variação\_temperatura = PP então variação\_resfriamento = PP, onde MQ = Muito Quente e PP = Pequeno Positivo.

A teoria de lógica fuzzy tem sido bastante empregada em sistemas que utilizam informações fornecidas por seres humanos para automatizar procedimentos como: controle de processos, auxílio à decisão e etc. Segundo Lee [6], a construção de um controlador fuzzy parte da idéia do armazenamento (incorporação) de “conhecimento especializado” baseado em experiências de um operador humano para obter estratégias de controle [35, 36] mais apropriadas.

Dissemos anteriormente que a lógica fuzzy foi desenvolvida objetivando tratar variáveis imprecisas de forma eficiente. Quando analisamos o tráfego de telefonia, observamos um alto grau de imprevisibilidade, fazer previsões de tráfego é uma tarefa bastante complicada, logo garantir QoS sem superprovisionamento de recursos tem sido um desafio para as empresas de telecomunicações. Apesar disso, existem pesquisas para provimento de QoS baseado em lógica fuzzy [5, 37], mais especificamente através do controlador fuzzy que tem a função de configurar dinamicamente a rede de acordo com o comportamento do tráfego, principalmente pelas características de incerteza e imprecisão do tráfego.

## 3.4 Controlador Fuzzy

Diferentemente dos controladores convencionais que utilizam algoritmos de controle baseados em equações algébricas e diferenciais de um modelo matemático, em lógica fuzzy são utilizadas regras no algoritmo de controle que visam descrever numa rotina a experiência humana, intuição e heurística para controlar um processo [34].

A Figura 3.2 apresenta a arquitetura de um controlador fuzzy genérico.

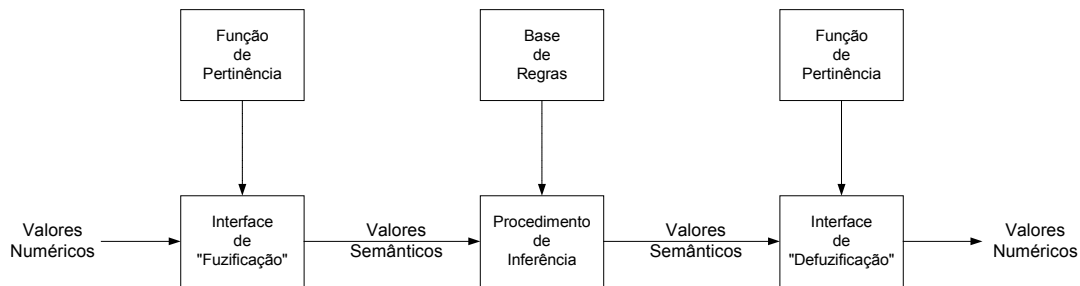


Figura 3.2: Estrutura de um Controlador Fuzzy.

As variáveis de entrada no controlador fuzzy são valores numéricos, chamados de valor *crisp*. Esses valores numéricos passam por funções de pertinência no processo de fuzificação onde são convertidos para valores semânticos. Normalmente, os valores semânticos advindos do processo de fuzificação adquirem um grau de pertinência o qual varia de 0 a 1, esses extremos correspondem respectivamente a total exclusão ou a total inclusão do valor na variável semântica.

Aos valores semânticos, são aplicadas bases de regras no procedimento de Inferência, que tem como saída, novos valores semânticos relacionados a atitudes a serem tomadas. A partir daí, é necessário converter esses valores semânticos em valores numéricos o que ocorre na fase de defuzificação.

#### 3.4.1 Interface de Fuzificação

Nesta fase, é feita a identificação dos valores das variáveis de entrada que caracterizam o estado do sistema (variáveis de estado). Em seguida, essas variáveis são normalizadas num universo de discurso padronizado. Estes valores são então “fuzificados”, com a transformação da entrada “crisp” em conjuntos nebulosos, afim de que estes possam se tornar em instâncias de variáveis lingüísticas.

No processo de normalização, os elementos do universo de discurso  $U$  são definidos como pares ordenados e são mapeados dentro do intervalo  $[0,1]$ . Os extremos 0 e 1 descrevem respectivamente total exclusão e total inclusão do elemento na variável semântica. Os valores intermediários do intervalo são o grau de pertinência dos elementos em relação a variável semântica. A possibilidade de possuir valores



intermediários, ou seja, um certo grau de incerteza, torna a teoria de lógica fuzzy eficiente para tratar de problemas reais com certo grau de complexidade.

#### 3.4.2 Procedimento de Inferência e base de regras

Conforme mencionado na seção 3.4, o que diferencia o controlador fuzzy do controlador convencional, é o fato de que no controlador fuzzy o conhecimento é expresso de forma intuitiva, onde são utilizadas regras com variáveis lingüísticas.

A base de regras é formada por estruturas do tipo:

**Se** <premissa> **então** <conclusão>

O procedimento de inferência tem a função de aplicar a base de regras nos dados de entrada, inferindo as ações de controle de acordo com o estado do sistema. Podemos dizer que o procedimento de inferência consiste em:

1. Verificar o grau de compatibilidade entre os fatos e as cláusulas nas premissas das regras;
2. Determinar o grau de compatibilidade global da premissa de cada regra;
3. Determinar o valor da conclusão, em função do grau de compatibilidade da regra com os dados e a ação de controle constante na conclusão;
4. Agregar os valores obtidos como conclusão das várias regras, obtendo desta forma uma ação de controle global.

Em um controlador fuzzy é fundamental existir tantas regras quantas forem necessárias para mapear totalmente as combinações dos termos das variáveis, ou seja, que a base de regras seja completa, garantindo que exista sempre uma regra a ser disparada para qualquer valor de entrada. Outro fator muito importante é a consistência, onde se procura evitar a possibilidade de contradições e a interação entre as regras.

### 3.4.3 Interface de Defuzificação

No controlador fuzzy, a Interface de Defuzificação é responsável por converter os valores semânticos obtidos após o processo de inferência em valores numéricos. Em seguida, são descritos os métodos de defuzificação mais utilizados.

1. primeiro máximo (SOM) - encontra o valor de saída através do ponto em que o grau de pertinência da distribuição da ação do controlador atinge o primeiro valor máximo;
2. método da média dos máximos (MOM) - encontra o ponto médio entre os valores que têm o maior grau de pertinência inferido pelas regras;
3. método do centro da área (COA) - o valor de saída é o centro de gravidade da função de distribuição de possibilidade da ação de controle.

Utilizamos, em nossa proposta, o método do centro da área.

## 3.5 Otimização com Algoritmo Genético

Inicialmente proposto por John Holland [38], os AGs receberam muitas melhorias ao longo das décadas de 80 e 90, acompanhando o crescimento da capacidade computacional de processamento. Eles são algoritmos de otimização global, baseados nas técnicas da seleção natural e da genética.

A estratégia empregada pelos AGs tem como premissa a pesquisa paralela e estruturada, voltada em direção à busca do reforço dos pontos de alta aptidão, isto é, pontos nos quais a função a ser otimizada possui valores comparativamente melhores. A busca por uma “solução ótima” não é aleatória ou não direcionada, na realidade esta busca pela solução ótima é feita através da exploração de informações históricas para encontrar novos pontos de busca onde são esperados melhores desempenhos. Isto é feito através de processos iterativos, onde cada interação é chamada de geração. Os AGs têm se mostrado como uma ferramenta muito eficaz no processo de otimização.

A otimização de um controlador fuzzy é um processo que visa encontrar a

melhor combinação de parâmetros para chegar a “solução ótima”. Tendo em vista que os parâmetros do controlador fuzzy são inter-relacionados, como os parâmetros de função de pertinência, as técnicas de AG têm se mostrado bastante útil, pois realiza uma heurística em todo o universo e produz resultados bons em tempo de processamento razoável.

## 3.6 Comentários

Foi apresentado neste capítulo, um resumo sobre os conceitos utilizados na proposta de Fernandez [5]. Tais conceitos, adicionados às técnicas de DW apresentadas no capítulo 2, serão utilizados na proposta apresentada nesse trabalho. Neste capítulo foram apresentados conceitos básicos sobre DiffServ, gerenciamento baseado em políticas, lógica fuzzy, controlador fuzzy e o uso do AG para otimização do controlador fuzzy.

Com base nos assuntos abordados neste capítulo e no anterior, será apresentada, no próximo capítulo, a metodologia proposta para integrar a tecnologia de DW a proposta de provisionamento dinâmico de recursos para melhoria da QoS de Fernandez [5].

# Capítulo 4

## Metodologia para Gerenciamento de Redes na Internet com auxílio de DW

Esse capítulo apresenta uma proposta de metodologia para provimento de QoS em um domínio DiffServ com auxílio de ferramentas de suporte a decisão. No capítulo 2, foi mostrado o comportamento de um ADW. No capítulo 3, foram apresentadas as tecnologias empregadas no trabalho de Fernandez [5] para provimento de QoS em um domínio DiffServ. O objetivo deste capítulo é apresentar uma proposta de usar o potencial das ferramentas de DW e DM para auxiliar no processo de gerenciamento de redes.

Na seção 4.1 é apresentada a arquitetura de um nó DiffServ. Nas seções 4.2, 4.3 e 4.4 são descritas respectivamente a proposta deste trabalho, sua arquitetura e metodologia. Nas seções 4.5 e 4.6, são mostradas a especificação das políticas de QoS e seu mapeamento em lógica fuzzy. Na seção 4.7, é apresentada a construção do controlador para provisionamento dinâmico de recursos. São descritas, nas seções 4.8 e 4.9, a estrutura de DW e DM, respectivamente. Por fim, nas seções 4.10 e 4.11, são apresentados a seleção dos dados para otimização, seguido do procedimento de otimização do controlador fuzzy através do AG.

## 4.1 Arquitetura do nó DiffServ

A função de controle numa arquitetura DiffServ é desempenhada por dois agentes com papéis distintos de acordo com sua localidade na arquitetura. Neste contexto, é feita distinção entre os nós de borda e os nós de núcleo. Apesar de possuírem diferentes funções de atuação, existem mecanismos comuns a esses dois tipos de nós. Eles são: uma fila distinta para cada classe de serviços, um classificador que coloca o pacote na fila correspondente e um escalonador que retira os pacotes das filas. Os nós de borda, além das funções anteriores, contêm um marcador que marca ou remarca cada pacote e um condicionador cuja função é controlar o fluxo de entrada. A arquitetura utilizada nesta proposta implementa dois controladores: um para o controle de filas e escalonador, usado nos nós de borda e de núcleo, e outro para o condicionador, utilizado apenas nos nós de borda. Na Figura 4.1, é mostrado o diagrama de um nó de borda do domínio DiffServ.

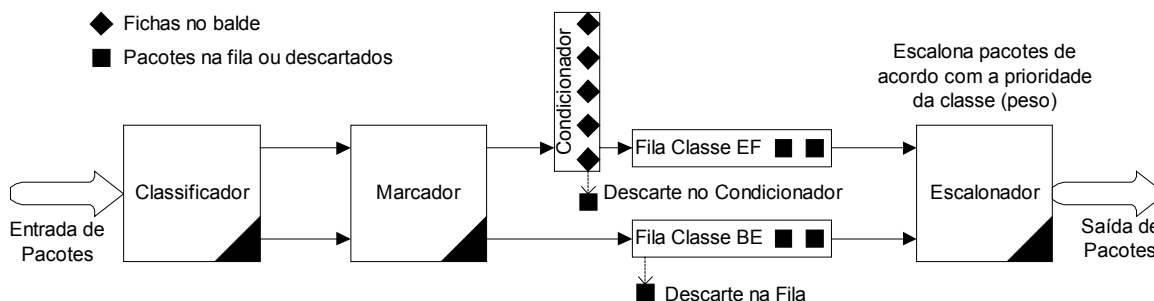


Figura 4.1: Diagrama de um nó DiffServ de borda

## 4.2 Proposta do trabalho

A arquitetura DiffServ visa fornecer QoS na Internet com escalabilidade para classes distintas de serviços oferecendo recursos de rede de forma diferenciada entre as classes. Embora tenha como objetivo garantir QoS para as classes mais prioritárias com escalabilidade, não se pode garantir QoS para todos os fluxos de uma mesma classe. Outro fator que dificulta o provimento de QoS é a aleatoriedade dos fluxos de tráfego na

Internet.

Apesar da imprevisibilidade do comportamento do tráfego na Internet, é conhecido que alguns fatores influenciam este fluxo, como por exemplo, o horário, o local, o dia, os fatores climáticos. Este trabalho se propõe a construir um modelo estrela com características de fluxo de dados de telefonia, que dão início a um DM para a área de Gerenciamento de Redes de uma empresa de Telecomunicações.

### 4.3 Apresentação da arquitetura

A arquitetura proposta, ilustrada na Figura 4.2, apresenta uma estrutura de auxílio à tomada de decisão a partir de um DM de Gerenciamento de redes.

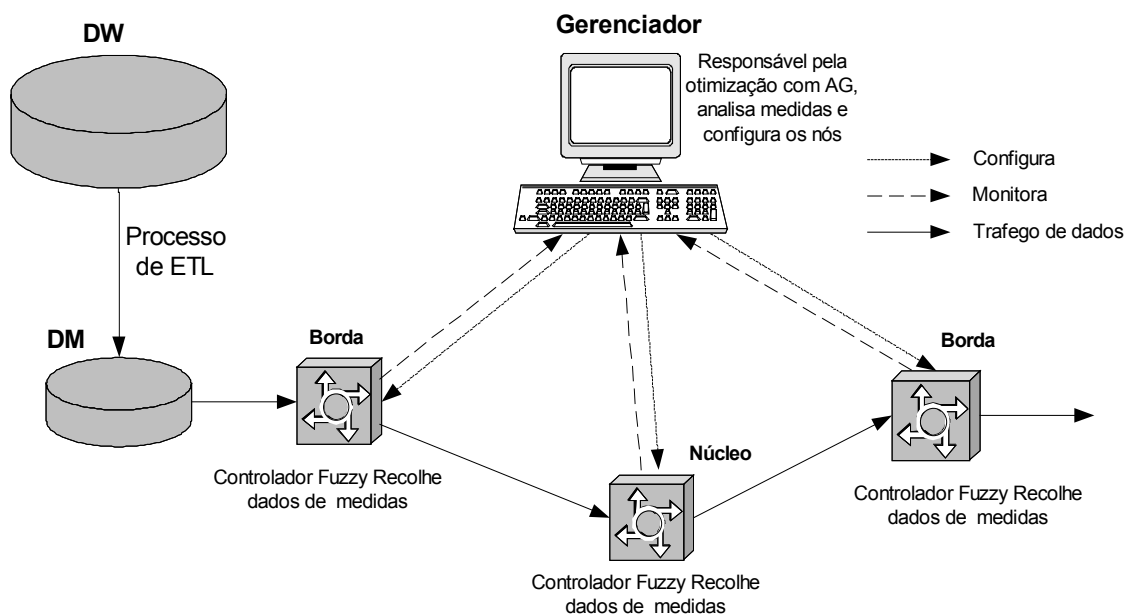


Figura 4.2: Arquitetura do trabalho proposto

Nesta proposta de arquitetura, o DW tem a função de fornecer dados históricos para o DM de gerenciamento de redes através do procedimento de ETL (*Extract, Transform and Load*), detalhado no Apêndice C. Desta forma, é de fundamental importância que a pessoa responsável por tomada de decisão especifique de forma clara e objetiva os requisitos que são relevantes para sua pesquisa, a fim de que esses dados sejam incorporados nas tabelas do DM para consultas.

Após ser efetuada a tomada de decisões com relação aos dados de entrada no domínio DiffServ, estes dados escolhidos são responsáveis pela geração do perfil de tráfego a ser usado na configuração da rede.

No domínio DiffServ, tanto os nós de borda como os de núcleo possuem controladores que realizam medidas do estado atual, calculam um novo valor de configuração utilizando lógica fuzzy, e aplicam um comando de controle no nó. A aplicação da lógica fuzzy é computacionalmente leve, desta forma pode ser executada em cada nó do domínio sem prejudicar no desempenho do roteador.

Cada nó do domínio é responsável por coletar as informações de estado do equipamento, como: retardo na fila e descarte de pacotes, e informá-las ao gerenciador, utilizando um protocolo de gerenciamento de redes, por exemplo, o SNMP (*Simple Network Management Protocol*).

O gerenciador recebe, dos nós, as informações do estado da rede, e então realiza cálculos a partir do AG para reconfigurar os nós do domínio. Desta forma, o objetivo do gerenciador é de configurar a rede para receber da melhor forma, ou seja, com os melhores parâmetros de QoS possíveis, um determinado perfil de tráfego informado previamente pelos nós do domínio DiffServ.

O AG é computacionalmente pesado, o que certamente implicaria em perdas de desempenho se fosse aplicado nos nós do domínio. Mas se aplicado apenas no gerenciador, em intervalos de horas ou dias, não afeta na performance desses nós, mantendo assim a escalabilidade mesmo para grandes domínios.

Como o gerenciador recebe dados de informações de estado provenientes dos nós do domínio DiffServ, e os nós de borda recebem como fluxo de entrada dados a partir de uma base de dados históricos, então ao findar o processo de otimização, teremos como resultado uma rede configurada para um perfil de tráfego específico. Exemplos de perfil de tráfego específico são: um dia de final de semana, um feriado de natal ou ano novo, uma determinada hora de tráfego intenso numa determinada campanha de promoção de produtos e etc.

## 4.4 Apresentação da metodologia

As empresas de telecomunicações utilizam bastante as ferramentas de DW e DM em seus modelos de negócio afim de, entre outras estratégias relacionadas ao negócio, fazer campanhas de marketing, identificar perfis de usuários importantes, identificar clientes mal pagadores “*black-list*”. Essas empresas, de forma geral, desprezam o potencial destas ferramentas para medição de tráfego de telefonia e, conseqüentemente, a utilização dessas medições de históricos de tráfego para auxiliar no processo de configuração da estrutura operacional das redes. Como conseqüência disso, as empresas adotam como solução do problema de provimento de QoS, o superdimensionamento das redes.

Este trabalho se propõe a apresentar uma metodologia de integração do potencial das técnicas e ferramentas de DW e DM com a tarefa de gerenciamento e configuração dos recursos das redes para provimento de QoS, evitando, assim, parte do desperdício de recursos ociosos das redes quando se usa a técnica de superdimensionamento.

O fluxograma da metodologia proposta é apresentado na Figura 4.3. A nomenclatura usada nessa figura foi: os retângulos representam os procedimentos da metodologia e os pentágonos representam as interações com os usuários ou com os equipamentos da rede. Posteriormente são apresentados os procedimentos da metodologia, a estrutura do DW e do DM usados e o processo de utilização do AG para otimização dos recursos da rede.



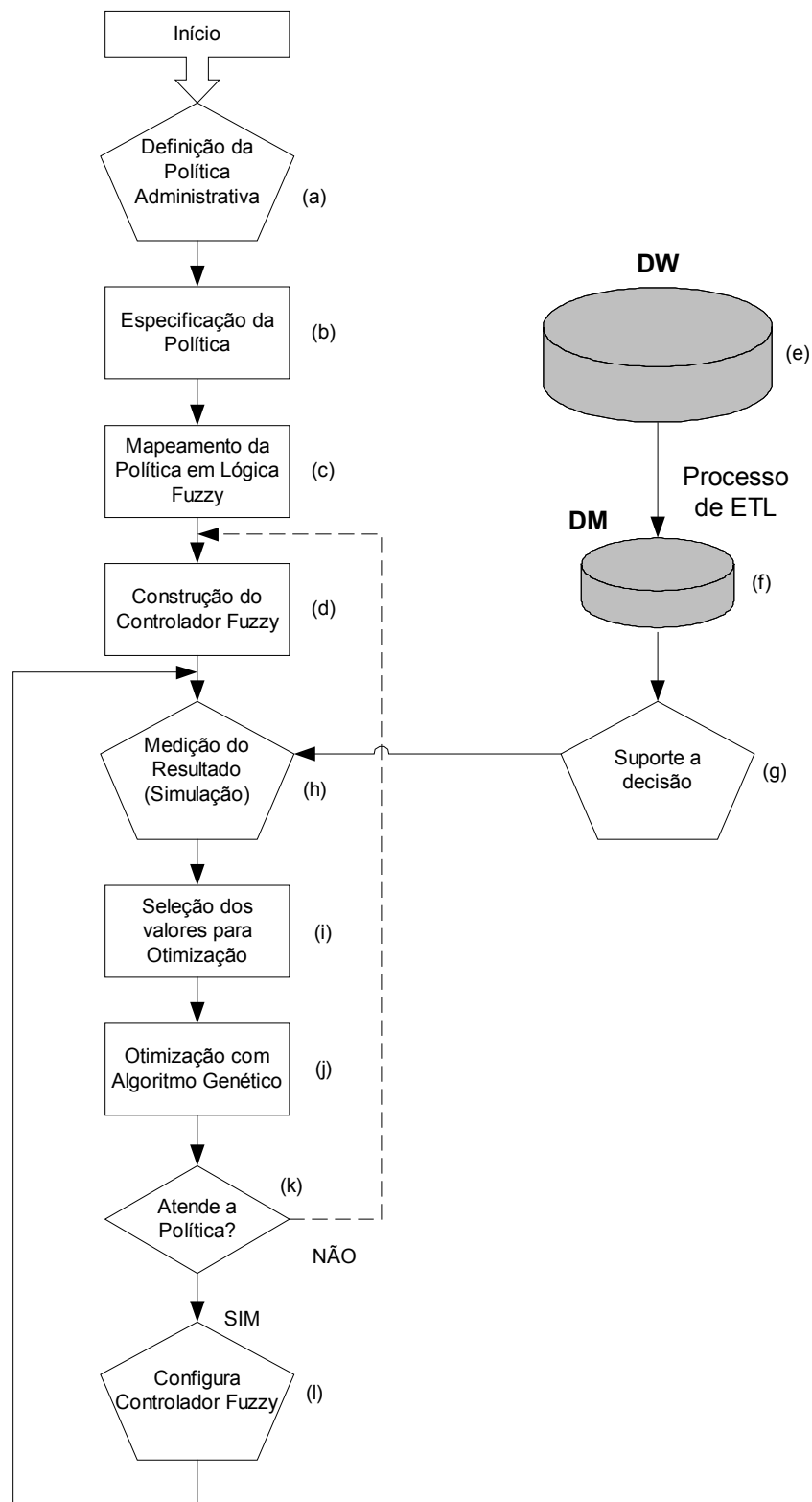


Figura 4.3: Fluxograma da metodologia proposta.

O início da proposta de integração se dá na definição das políticas administrativas de gerenciamento com base nos requisitos de QoS demandados, mostrado na Figura 4.3(a). Após esta fase, temos a especificação das políticas administrativas, mostrado na Figura 4.3(b). A Figura 4.3(c) apresenta o mapeamento das políticas administrativas em lógica fuzzy, mecanismo este que será utilizado nos controladores da rede.

A próxima fase é a construção do controlador. Nesta etapa são definidos os parâmetros do controlador fuzzy, como funções de pertinência e base de regras, a partir da especificação das políticas administrativas mapeadas anteriormente, esta fase é mostrada na Figura 4.3(d).

Deve-se ressaltar que, no processo de construção do controlador fuzzy, a definição de regras e parâmetros do controlador depende muito da decisão do projetista, podendo, na ocorrência de falhas, produzir resultados incorretos. Além disso, o atendimento as políticas não garante a eficiência do controlador, ou seja, não garante que este utilize os recursos da rede da melhor forma. Deste modo, é necessário realizar dois processos de otimização. No primeiro, é utilizado o algoritmo de Wang-Mendel, detalhado na seção 4.7.3, que visa construir uma base de regras correta, eliminando as inconsistências. No segundo, é usado o AG o qual será descrito na seção 4.11.

A Figura 4.3(e) mostra o início do processo de integração dos recursos de DW no gerenciamento das redes. Nesta fase, os dados armazenados no DW são explorados de forma a atender aos interesses dos responsáveis pela tomada de decisão. Fatores considerados nessa fase são: granularidade, periodicidade, temporalidade e regionalidade dos dados que serão estudados.

Após a etapa de estudos e definições dos dados que são relevantes para o processo de tomada de decisão, esses dados devem ser carregados em tabelas do DM específico, através do procedimento de ETL, para análise. Esta fase é mostrada na Figura 4.3(f).

A partir do DM carregado, mostrado na etapa anterior, os responsáveis pela tomada de decisão têm condições de preparar suas consultas com o intuito de procurar cenários passados que reflitam situações e acontecimentos semelhantes ao cenário pelo

qual se deseja provisionar QoS as redes. Fatores históricos, fatos climáticos e datas festivas como comemoração de Natal ou Ano Novo, são detalhes de grande importância para se identificar o perfil do tráfego. Apresentamos esta etapa na Figura 4.3(g).

Na Figura 4.3(h), é mostrada a etapa de medição dos resultados através de simulação. Nesta fase, são coletados os valores de entrada e saída nos controladores fuzzy e as medidas de desempenho desejadas, como retardo ou descarte de pacotes.

A Figura 4.3(i) mostra que de posse dos valores colhidos na etapa anterior, pode-se escolher as combinações de valores de entrada e saída que maximizam as medidas de desempenho desejadas para o tráfego em questão.

A Figura 4.3(j) mostra a aplicação do AG utilizado no processo de otimização dos parâmetros fuzzy, os valores selecionados na etapa anterior são utilizados nesta fase como referência para a função objetivo.

Embora seja de grande valia, a otimização por AG pode distorcer as funções de pertinência e regras do controlador, fazendo com que os resultados dos controladores deixem de cumprir com as políticas originais, apesar de estarem otimizados para as métricas de escolha dos valores. A etapa apresentada na Figura 4.3(k), mostra a necessidade de verificar se as novas funções de pertinência e bases de regras estão de acordo com a especificação das políticas.

Se o controlador apresentar um resultado condizente com as políticas originais, este deve ser utilizado nos equipamentos para funcionamento normal, situação mostrada na Figura 4.3(l).

Caso o controlador apresente um resultado em desacordo com as políticas originais, o controlador deve ser refeito na etapa da Figura 4.3(d), ilustrado com uma linha tracejada no fluxograma da metodologia proposta.

Ressalta-se que o processo de otimização, reconfiguração da rede, deve ser feito sempre que se desejar alterar o perfil da configuração da rede para suportar da melhor forma um tráfego previsto. Nesta fase de otimização, os controladores fuzzy se adaptam com as mudanças ocorridas na rede, como, ativação e desativação de um canal, procurando ajeitar a rede para o padrão do tráfego de entrada. Normalmente, as mudanças são pequenas e eventuais, desta forma o AG tem se mostrado eficiente na

otimização. O processo de reconfiguração da rede é mostrado na Figura 4.3 a partir de uma seta cheia saindo da fase (l) para (h), levando em consideração que as políticas administrativas não foram alteradas. Caso contrário, deve-se iniciar o processo a partir da etapa apresentada na Figura 4.3(a).

Em suma, neste trabalho é proposto a integração das tecnologias de DW e DM no auxílio ao planejamento operacional das redes de telefonia com suporte a QoS, ou seja, foi acrescentada uma estrutura de suporte a decisão quanto aos dados de entrada na metodologia definida por Fernandez [5], que projeta um controlador fuzzy com regras corretas e otimizadas para o tráfego entrante. Deste modo, foi apresentada uma metodologia desde o planejamento do perfil de tráfego entrante e a especificação de políticas de gerenciamento, até a reconfiguração dos parâmetros dos roteadores.

### 4.5 Especificação das políticas de QoS

Partindo dos requisitos administrativos, é possível especificar políticas de gerenciamento para provisão de QoS, ou seja, é possível produzir regras que direcionem o comportamento do nosso sistema de gerenciamento. Neste trabalho, a política definida é de que toda a prioridade deve ser concedida ao tráfego EF (*Expedited Forwarding*). Assim, a classe BE (*Best Effort*) terá sua prioridade reduzida sempre que houver queda na qualidade da classe EF. Foram definidas duas especificações de políticas: uma para o escalonador, que será aplicada em todos os nós do domínio DiffServ, e outra para o condicionar, aplicada apenas aos nós de borda.

A linguagem adotada neste trabalho para a especificação das políticas foi a linguagem Ponder [49]. Essa linguagem foi proposta por Damianou em [50] para especificar textualmente políticas de gerenciamento de acordo com as propostas de Sloman [29] e Lupu [51, 52]. Consiste numa linguagem declarativa, orientada a objetos, e oferece ao usuário uma interface simples para especificação de políticas, aproximando-se das regras de políticas abstratas. Essa linguagem define quatro políticas básicas, que são: (i) política de autorização, que pode ser positiva (auth+), permitindo o acesso a um determinado recurso, ou negativa (auth-), proibindo o acesso; (ii) política

de obrigação (oblig), que exige a execução de uma determinada operação autorizada anteriormente; (iii) política de proibição (refrain), que proíbe a execução de uma determinada ação; e (iv) política de delegação (deleg), que permite a um elemento delegar a outro o controle sobre um determinado objeto.

Além das políticas básicas, pode-se definir composição de políticas, como: (i) (group), agrupa um conjunto de políticas sob um mesmo nome, podendo ser estabelecidas várias políticas a partir de um único comando; e (ii) (role), que permite a substituição de um sujeito pertencente a um grupo de políticas, deste modo um usuário pode assumir o papel de outro.

É apresentado no código 4.1, um fragmento de programa que ilustra um exemplo de especificação de políticas do escalonador em Ponder.

Código 4.1: Trecho exemplo de especificação Ponder do escalonador

```
1 //Especificação do Controlador do Escalonador
2 //
3 //Define limites mínimos e máximos permitidos para o escalonador
4 const
5 minsched = 0.10 ; // Escalonador minimo 10%
6 maxsched = 0.90 ; // Escalonador maximo 90%
7 MaxDelay = 100 ; // Delay maximo 100 ms
8
9 //Define condições de restrição
10 constraint
11 bwMin = bwShare < minsched ;
12 bwMax = bwShare > maxsched ;
13 bwIncrease = bwShare < maxsched ;
14 bwDecrease = bwShare > minsched
```

Neste código, observa-se que entre as linhas 4 e 7 são definidos os valores máximos dos parâmetros de QoS para uma determinada classe de serviço (EF). De acordo com o código, o escalonador varia nos intervalos de 10% a 90% da banda de

saída e o retardo máximo é de 100 ms. Das linhas 10 a 14, são definidas as restrições baseadas nos valores previamente estabelecidos.

## 4.6 Mapeamento da especificação de políticas em lógica fuzzy

A especificação de políticas tem por objetivo transformar uma decisão administrativa em comando do sistema de gerenciamento. Como as regras de políticas são abstratas e próximas da percepção humana, é muito difícil mapeá-las em regras computacionais, absolutas e exatas por natureza. Desta forma, a aplicação da lógica fuzzy tem se mostrado uma ótima solução para este problema, apresentando-se apta a tratar variáveis semânticas com certo grau de imprecisão. Conseqüentemente, é praticamente intuitiva a aproximação das regras de especificação de políticas de gerenciamento dos atributos de um controlador fuzzy.

A linguagem Ponder apresentada na seção anterior possui a característica de especificar vários elementos de políticas de gerenciamento de forma textual. Com isso, é possível representar funções de restrição e evento através de funções de pertinência, enquanto funções de comando podem ser representadas por bases de regras.

Para exemplificar o mapeamento da especificação de políticas em lógica fuzzy, é apresentado resumidamente os comandos *constraint*, *auth-*, *event* e *oblig*. Maiores detalhes sobre esses comandos, bem como seus gráficos de funções de pertinência podem ser encontrados em Fernandez [5].

Os comandos *constraint* e *auth-* são utilizados para indicar o limite de restrição de uma autorização, normalmente definem um valor mínimo ou máximo para uma função. O objetivo da associação em operação fuzzy está em estabelecer o valor defuzificado da variável de saída igual ao valor desejado. É apresentado no código 4.2, uma especificação usando os comandos em questão. Pode-se observar que neste caso, a especificação da banda mínima do escalonador é de 0.10, ou seja, 10% da banda total. O comando *auth-* indica a não autorização de executar o comando de reduzir banda do escalonador, caso este possua como banda atual o menor valor autorizado.

Código 4.2: Mapeamento do comando *constraint*

```
1  constraint
2    bwmin = bwShare <= 0.10;
3  inst
4    auth- schedulerMin {
5      subject s;
6      target sched;
7      action sched.reduceBW( );
8      when bwMin;
9    }
```

O comando *event* tem a função de listar os eventos que disparam comandos de obrigação (*oblig*) e proibição (*refrain*). O comando *oblig* indica a obrigação de execução de uma ação caso uma determinada condição (evento) ocorra. É apresentado no código 4.3, o mapeamento dos comandos *event* e *oblig*.

Código 4.3: Mapeamento dos comandos *event* e *oblig*

```
1  event
2    RB = 0.1 ; // Retardo Baixo
3    RM = 0.5 ; // Retardo Médio
4    RA = 0.9 ; // Retardo Alto
5
6  inst
7    oblig aumentaEscalonador {
8      subject s;
9      target sched;
10     on classeEF.RA;
11     do sched.increaseBW()
12   }
```

A partir dos exemplos anteriores, apresenta-se no código 4.4, uma descrição JFS [53] da base de regras do controlador fuzzy. Neste exemplo de código JFS, pode-se observar que o mapeamento de um comando `increaseBW` foi separado em várias regras, `if <condição> then <ação>`, com o intuito de atender a todos os valores semânticos da função de pertinência.

Código 4.4: Código JFS mapeado a partir do comando `oblig`

```
1  program
2  if classeEF RM and sched PB then sched PM;
3  if classeEF RM and sched PM then sched PA;
4  if classeEF RM and sched PA then sched PA;
```

Observa-se que a partir da ação `increaseBW()` do código 4.3 em Ponder, são definidos três comandos JFS no código 4.4, por ser exigida a especificação de ação para cada valor semântico de entrada.

## 4.7 Construção do Controlador fuzzy

Uma vez realizado o processo de especificação das políticas administrativas para provimento de QoS e feito o mapeamento da especificação de políticas em lógica fuzzy, pode-se especificar o controlador fuzzy. Neste trabalho foi definido que toda a prioridade deve ser dada à classe EF. Desta forma, a classe BE será reduzida sempre que houver queda na qualidade da classe EF.

### 4.7.1 Controlador do Escalonador

O controle do escalonador está diretamente ligado a variável de saída do processo de configuração e esta depende do tipo de mecanismo utilizado. Objetivando prover uma capacidade mínima de 10% de banda para a classe menos prioritária, independente da situação da rede, é identificado que o escalonador que permite este



controle deve ser do tipo WRR (Weighted Round-Robin) ou WFQ (Weighted Fair-Queueing), onde as filas são servidas de acordo com o peso definido na configuração. Alterando-se este peso, pode-se modificar a QoS em cada classe.

São listadas abaixo as variáveis de entrada no modelo, que influem diretamente no valor da variável de saída.

1. Peso relativo atual do escalonador (EF/BE).
2. Retardo médio da classe EF.
3. Perda de pacotes na classe BE.

A primeira variável de entrada corresponde ao peso relativo das classes EF e BE no escalonador o qual será utilizado como referência para escolha do novo peso. A segunda variável corresponde ao retardo médio da classe EF, ou seja, é o tempo que o pacote fica na fila. Como os demais tempos do processamento do pacote são praticamente irrelevantes, considerou-se o tempo de espera na fila como o tempo total de permanência no nó. A terceira variável de entrada é a taxa de descarte por transbordamento da fila BE, indicando a exaustão de recursos. Podem ser usados mecanismos de gerenciamento ativo de fila para tentar minimizar o descarte, mas deve-se considerar que para chegar a este ponto existe grande escassez de recursos na rede.

Como variável de saída, temos apenas o novo peso relativo no escalonador da fila EF/BE.

Um fator importante que merece destaque, é como o valor de configuração (variável de saída) vai influenciar no funcionamento do roteador. O peso do escalonador (variável de entrada e saída) determina a quantidade de pacotes servidos de cada classe, definindo sua prioridade. Desta forma, a alteração do peso das classes no escalonador não influencia o estado corrente do roteador, isto é, apenas os pacotes que chegarem no nó após sua reconfiguração sofrerão o efeito da mesma.

## 4.7.2 Controlador do Condicionador

O condicionador se encontra apenas nos nós de borda, como dito no início deste capítulo, e possui como objetivo o controle da entrada de fluxos de dados no domínio, de forma que os fluxos que respeitam seus limites não sejam prejudicados em função de outros mal comportados. Normalmente, este mecanismo do condicionador é implementado seguindo a filosofia do balde furado, que consiste em um reservatório no qual são postas fichas numa taxa constante e do qual cada pacote que entra retira uma ficha. O condicionador policia a taxa de entrada, mas admite rajadas controladas.

São listadas abaixo as variáveis de entrada no condicionador:

1. Taxa atual de enchimento do balde do condicionador.
2. Retardo médio da classe EF nos nós de núcleo.
3. Descarte de pacotes da classe EF no condicionador.

A primeira variável de entrada é a taxa de enchimento do balde no condicionador a qual terá grande importância na escolha da próxima taxa. A próxima variável de entrada consiste no retardo médio da classe EF nos nós de núcleo. Caso este aumente muito, certamente será necessário reduzir a taxa de enchimento do balde afim de evitar descarte de pacotes no interior do domínio DiffServ. A terceira variável de entrada é o descarte de pacotes da classe EF no condicionador. O descarte ocorre quando um pacote chega ao condicionador e não encontra ficha no balde, ou seja, o tráfego entrante está maior que a taxa máxima permitida.

Tendo em vista que o condicionador tem como principal objetivo manter o tráfego suavizado e de acordo com as decisões contratuais, o mesmo não deve ter o valor de sua taxa de enchimento do balde alterada para suportar um tráfego além do contratado. Desta forma, todo o descarte de pacotes só deve ocorrer nos nós de borda, o que está de acordo com Jacobson [54], pois este diz que o PHB EF em um domínio DiffServ só deve descartar pacotes em seus nós de borda. Sendo assim, quando não houver mais recursos no núcleo do domínio, deve ser sinalizado para os nós de borda uma redução da taxa de entrada, através da redução da taxa de enchimento do balde.

Isto se dará da seguinte forma, o valor de retardo médio nos nós de núcleo é enviado ao gerenciador, este será encarregado de sinalizar a redução da taxa de entrada para os nós de borda.

Como variável de saída, temos apenas a nova taxa de enchimento do balde.

Da mesma forma que no controlador do escalonador, a mudança na taxa de enchimento do balde terá efeito apenas para os pacotes que chegarem após a reconfiguração da rede.

### 4.7.3 Algoritmo de Wang-Mendel

Em seções anteriores, foram apresentados os conceitos da lógica fuzzy bem como a construção de um controlador baseado nesta tecnologia. A lógica fuzzy possui a vantagem de produzir respostas eficazes, mesmo com entradas duvidosas e as vezes até ambíguas. Porém, um comportamento eficiente requer a definição de regras coerentes. Visando a produção das funções de pertinência utilizadas em lógica fuzzy e de um conjunto de regras coerente sem que isto esteja totalmente a cargo do projetista de sistemas, utilizamos o algoritmo de Wang-Mendel [57], que, partindo de um comportamento desejado, cria um conjunto de regras não contraditórias.

No experimento proposto, a grande aplicação deste algoritmo está na verificação da consistência do conjunto de regras criadas pelo projetista, identificando a ocorrência de regras contraditórias que acarretariam em resultados distorcidos e errados.

A implementação do Algoritmo de Wang-Mendel utilizada foi apresentada por Cox [58] e seu funcionamento básico é apresentado no algoritmo 4.1

**Algoritmo 4.1:** Algoritmo Wang-Mendel

**Entrada:** Base de regras não verificada

$contador(i) = 0$  {Inicia contador de posto}

$contradicao(i) = 0$  {Marca todas as regras contraditórias}

Normaliza as regras não verificadas na forma IF <variável> IS <adjetivo> AND <variável> IS <adjetivo> AND . . . THEN <variável> IS <adjetivo>.

**enquanto** Existe regra a ser verificada **faça**

se Regra já existe na base de regras **então**

Regra não é incluída e  $contador(i)$  é incrementado

**senão se** Regra ainda não existe e não contradiz nenhuma outra regra **então**

Regra adicionada à base de regras e  $contador(i)$  é incrementado

**senão se** Regra contradiz alguma regra existente **então**

Regra incluída na tabela de regras e marca  $contradicao(i) = 1$  e  $contador(i)$

é incrementado.

**fim se**

**fim enquanto**

Regras com  $contradicao(i) = 0$  são incluídas na base de regras final.

Regras com  $contradicao(i) = 1$  incluem a regra com  $contador(i)$  maior e descarta a outra.

### 4.8 Estrutura do DW

As variáveis de entrada no controlador, como vimos nas seções anteriores, influirão diretamente no processo de configuração da rede, a fim de que esta seja melhor adequada para os próximos fluxos de tráfego. Deste modo, é possível fazer consultas às bases do DW para planejar mudanças na configuração da rede, pode-se prever possíveis perfis de tráfego com base em dados armazenados no DW de uma empresa de telecomunicações, e assim, ajustar a estrutura da rede para melhor acomodá-lo, ajudando assim, a fornecer a QoS desejada. A importância de escolhermos bem os dados de entrada antes de reconfigurar a rede será apresentada em 4.10.

Segundo Inmon [55], os dados de uma empresa devem ser divididos em quatro níveis dentro de um ambiente arquitetural que são: operacional, atômico, departamental e individual. Esses diferentes níveis de dados perfazem a base para uma arquitetura maior conhecida como “*Corporate Information Factory*”. A mudança de níveis dos dados é ilustrada na Figura 4.4 e são descritos quais os tipos de dados que estão localizados em cada nível.

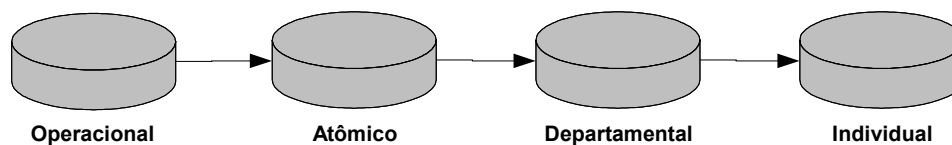


Figura 4.4 Níveis da arquitetura de dados de uma empresa

1. Nível Operacional - neste nível, os dados são acessados apenas por aplicações orientadas a dados primitivos e que necessitam de alto grau de performance, como aplicações on-line;
2. Nível Atômico - este é o nível do DW, onde os dados devem estar em seu menor nível de granularidade e bem integrados. Neste nível deve ser mantido o histórico de dados os quais não podem ser atualizados. Outro fator importante é que esta estrutura de armazenamento é variante no tempo;
3. Nível Departamental - este é o nível do DM, onde se localizam os dados derivados. Aqui os dados são organizados exclusivamente para atender demandas do usuário final de um departamento específico;
4. Nível Individual - neste nível são localizadas respostas às consultas do usuário, como: consultas ad hoc, tabelas temporárias, entre outras.

O DW é orientado e organizado por assuntos de maior interesse da companhia [55], onde os dados são fisicamente armazenados em tabelas que devem estar preparadas para atender requisições de diferentes perfis de usuários. Deste modo, devemos ressaltar que o processo de ETL dos Sistemas Operativos deve estar alinhado com as necessidades demandadas pelo setor de planejamento e gerencia de redes.

Dados sobre fatores climáticos normalmente não fazem parte do escopo do DW. Entretanto, esses dados seriam de grande valia ao nosso projeto, já que este tipo de informação certamente influencia significativamente no comportamento do tráfego de dados. Outro aspecto interessante seria o tratamento personalizado dos dados referentes a datas especiais, como dia das mães, dia do funcionário público, dia das crianças, além dos feriados regionais pelos quais podemos estimar um perfil de tráfego específico para a localidade em questão.

É bom lembrar que o processo de desenvolvimento do DW deve ser feito em espiral, onde pequenas partes do DW são desenvolvidas até completar o projeto, permitindo que futuramente novas partes possam ser desenvolvidas e incorporadas na estrutura inicial. Desta forma, empresas de telecomunicações que não possuem em seus DWs informações que possam auxiliar no processo de tomada de decisão para gerencia de redes, podem, desenvolver este projeto e incorporá-lo em seus DWs.

### **4.9 Estrutura do DM**

Nesta estrutura serão feitas as consultas às bases de dados para escolha dos dados de entrada na arquitetura DiffServ. Como foi mostrado no modelo estrela apresentado no capítulo 2, as definições das tabelas de dimensão e de fato estão inseridas no DM com o propósito de prover uma estrutura de modelo multidimensional para auxiliar no suporte a decisão.

Como ilustrado na Figura 4.4 da seção anterior, os dados que povoarão as tabelas de dimensão do DM procedem do DW. Desta forma, para construir o DM de gerenciamento de redes, deve-se conhecer bem as requisições que deverão ser atendidos pelo DM. Uma vez conhecidas essas requisições, deve-se estruturar as tabelas do modelo estrela para suportá-las.

No experimento proposto nesse trabalho, a ferramenta de DM utilizada foi o Pilot [56]. Esta ferramenta consiste em um banco de dados multidimensional, capaz de ler as tabelas de dimensão e de fato de um modelo relacional específico de uma aplicação, e a partir delas prover uma ferramenta de front & end capaz de fornecer ao usuário uma interface simples que pode ser composta de tabelas e/ou gráficos, onde se podem variar as visões de consultas a fim de se obter um resultado em diferentes ângulos, auxiliando significativamente no processo de tomada de decisão.

Ao se introduzir as tabelas no banco de dados do Pilot, deve-se especificar o nível de detalhe que é desejado nas consultas. Nesse trabalho, é adotada na dimensão tempo a hora como maior nível de detalhe e na dimensão região, é adotado o Estado como maior nível de detalhe. A partir daí, foi construído em Pilot, um painel de

#### 4. Metodologia para Gerenciamento de Redes na Internet com auxílio do DW

instrumentos “Dashboard” onde é possível visualizar em cada hora de um dia, o total de minutos de telefonia convencional e avançada, juntamente com seu percentual em relação ao total acumulado no dia para cada Estado.

É mostrado na Figura 4.5, numa tela customizada, o resultado de uma consulta feita à base de dados do Pilot.

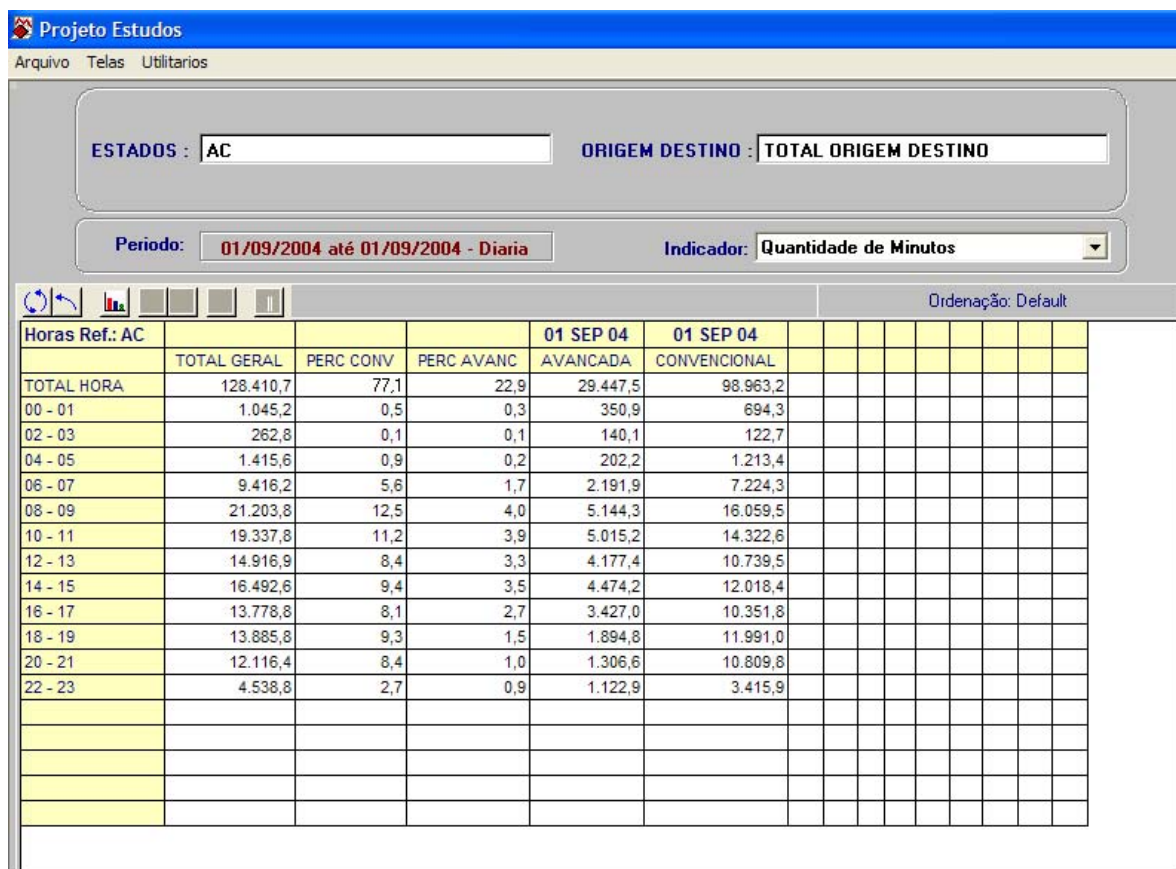


Figura 4.5: Percentuais de telefonia (Acre)

A Figura 4.5 mostra o resultado de uma consulta onde foram solicitados os percentuais de telefonia avançada e convencional, juntamente com a quantidade geral e em parte de minutos pelos quais a rede foi utilizada por estes diferentes tipos de tráfego no dia 01 de Setembro de 2004 no estado do Acre e na fração de duas em duas horas.

É apresentada na Figura 4.6, uma interface customizada resultante de outro exemplo de consulta utilizando a ferramenta Pilot.

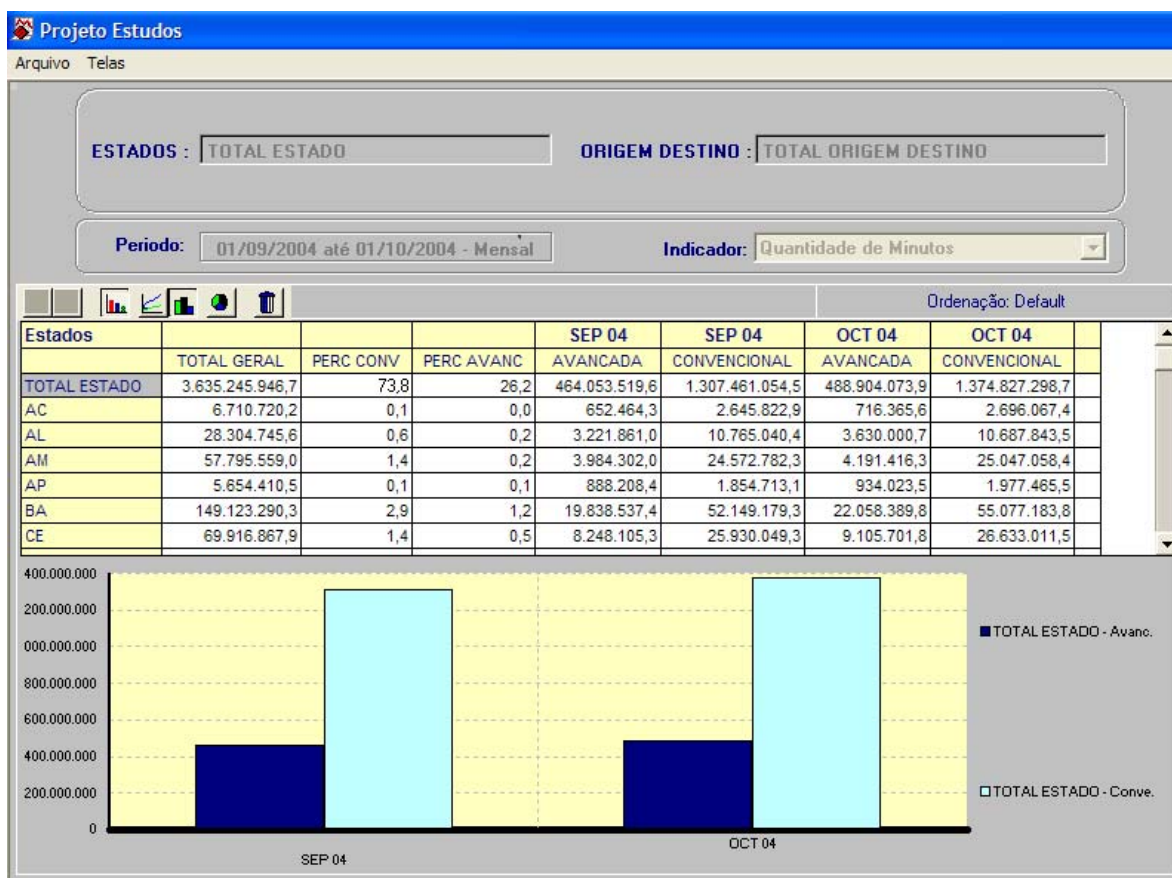


Figura 4.6: Percentuais de telefonia (Estados do Brasil)

A Figura 4.6 mostra o resultado de uma consulta onde foram solicitados os percentuais de telefonia avançada e convencional, juntamente com o total geral de minutos pelos quais a rede foi utilizada por estes diferentes tipos de tráfego no bimestre correspondente aos meses de Setembro e Outubro de 2004, em todos os Estados e em cada Estado do Brasil, bem como os totais gerais de telefonia avançada e convencional dos meses separadamente. Além disso, foi solicitada a construção do gráfico em barras do total geral de minutos de tráfego convencional e avançado dos meses que compõem o bimestre em questão.

Deve ser ressaltado que deveria ser utilizado pacotes de telefonia IP como tráfego na arquitetura DiffServ, mas por falta de informação quanto ao tráfego IP e pelo conhecimento adquirido quanto ao tráfego de telefonia em redes convencionais de telecomunicação, adotamos como tráfego de entrada em nosso domínio DiffServ valores percentuais típicos de uma empresa de telecomunicações. Esses valores serão



convertidos em fontes de pacotes IP através de um fator de ajuste, ou seja, multiplicamos o percentual de tráfego por um número inteiro ou fracionário para obter uma média de utilização da rede pelos diferentes tipos de tráfego.

É importante observar que, em nosso exemplo, a telefonia avançada compõe a classe EF, ou seja, a classe de maior prioridade, e a telefonia convencional foi tratada a partir da classe BE (menos prioritária).

Além das dimensões tempo e região mencionadas anteriormente, nosso modelo estrela possui a dimensão tipo de telefonia que pode possuir o valor convencional ou avançado e a dimensão tráfego que pode possuir o valor entrante, sainte ou em transito. Esta última serve para sinalizar que o tráfego apenas passa pela rede da região em questão, ou seja, esta região não originou o tráfego e nem tão pouco é a região a qual o tráfego se destina.

É bom lembrar que o objetivo do DM não é de prover uma solução direta para o problema de escolha dos melhores valores de entrada para configuração de uma rede com QoS, mas de auxiliar o gerente de redes a escolher os melhores dados, baseado numa série de consultas as bases de dados do DM.

O modelo estrela proposto para o DM de gerenciamento e planejamento de redes desenvolvido em nosso trabalho é ilustrado na Figura 4.7.

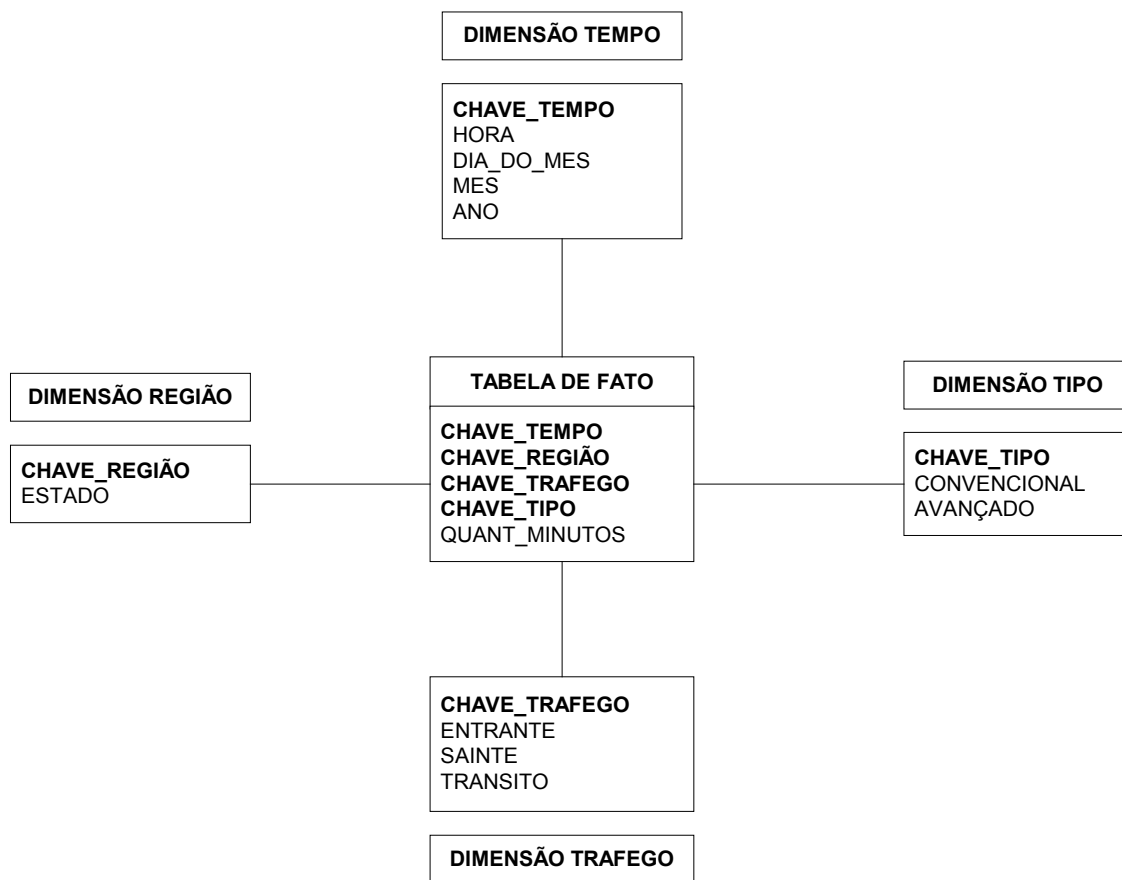


Figura 4.7: Modelo Estrela proposto

### 4.10 Seleção dos valores para otimização do controlador fuzzy

O processo de otimização por AG normalmente é feito através de uma função objetivo para onde a otimização deve convergir. Mas no projeto proposto há grande dificuldade em atingir esta função [59]. É possível notar esta dificuldade ao se levantar os requisitos do controlador do escalonador, onde tem-se como variáveis de entrada: o peso inicial do escalonador, o retardo na fila EF e o descarte na fila BE e, como variável de saída, o peso do escalonador. Entretanto, a variável a ser otimizada é o retardo da fila EF, que somente pode ser avaliado em simulação.

Por conta disso, é definida uma metodologia para obter os valores utilizados na otimização. Deste modo, são armazenadas todas as combinações de parâmetros de

entrada do controlador fuzzy juntamente com suas métricas de avaliação como, por exemplo, a taxa de descarte de cada classe e o valor de retardo obtido no período seguinte a aplicação das variáveis de entrada, ou seja, o resultado obtido pela aplicação de parâmetros na simulação a partir das funções de pertinência criadas pelo projetista.

Partindo dos valores resultantes da simulação, são escolhidas as combinações de parâmetros que produzem um baixo valor de retardo, aquelas que produzem uma maior redução no retardo dos pacotes em medidas consecutivas, e aquelas que produzem uma menor taxa de descarte agregado entre as classes EF e BE. De posse dessas combinações, é estabelecido um critério para ordenação das medidas, uma vez que as mesmas são independentes e algumas até contraditórias, como baixo descarte e baixo retardo. Assim, são ordenadas as medidas individualmente por ordem crescente para as métricas: baixo retardo e baixa taxa de descarte agregado; e por ordem decrescente para a métrica: maior redução no retardo. A partir daí, é calculado um valor de posto igual à média da posição de cada parâmetro, isto é, menor retardo absoluto, maior redução no retardo e menor descarte. Sabendo que as duas primeiras medidas estão relacionadas ao retardo e a terceira ao descarte, pode-se dizer que a otimização utilizou como medida de ponderação os seguintes pesos:  $2/3$  para o retardo e  $1/3$  para descarte. Desta forma, obtemos, segundo a ponderação escolhida, as melhores combinações de parâmetros para o problema em questão.

De posse dos valores previamente selecionados, eles são usados como base de conhecimento para a aplicação do AG, o qual se encarregará de produzir o melhor conjunto de parâmetros do controlador fuzzy, visando atingir a otimização desejada. Em nossa aplicação, é utilizado como referência para escolha os 20% melhores valores.

De acordo com o que foi dito nesta seção, foi evidenciada a importância dos valores de entrada na aplicação, provindos a partir de consultas aos dados históricos de uma empresa de telecomunicações - DW, visto que a otimização por AG será aplicada de modo a acomodar a rede do domínio DiffServ para atender da melhor forma o perfil do tráfego entrante, fornecendo assim, a QoS desejada para o padrão de tráfego de entrada na topologia corrente da rede.

## 4.11 Otimização do controlador fuzzy através de Algoritmo Genético

A partir das funções de pertinência criadas pelo projetista e com as bases de regras do controlador verificadas pelo algoritmo de Wang-Mendel apresentado na seção 4.7.3, é definido um controlador correto, porém sem garantias de eficiência. Objetivando otimizar o resultado do controlador, devem ser escolhidos os melhores parâmetros do controlador através da aplicação do AG.

### 4.11.1 Funcionamento dos AGs

Inicialmente o AG cria um conjunto de soluções, a partir dos valores inicialmente arbitrados que são chamados de indivíduos, para um determinado problema. Este conjunto de soluções é chamado de população. Cada indivíduo possui um posto que é obtido a partir da comparação do resultado do cálculo do teste da solução em relação ao valor desejado. Os indivíduos com posto mais baixo são substituídos por novos indivíduos criados a partir de indivíduos com postos maiores. Este processo persiste até que uma determinada condição de parada seja atingida, por exemplo, um determinado número de gerações ou até que a solução idealizada seja atingida. A solução para o problema consiste no indivíduo com maior posto alcançado. Em geral, um AG modela uma solução possível para o problema na forma de um vetor com um número fixo de posições (cromossoma) [60].

No exemplo desse trabalho, partindo da tabela com valores de entrada e saída desejados, escolhidos na seção 4.10, o AG cria um indivíduo, avalia se o resultado calculado se aproxima do valor desejado e pontua cada avaliação. A cada geração, os indivíduos mais aptos são selecionados e combinados por meio de *cruzamento*, que toma dois indivíduos selecionados e casando-os aleatoriamente gera indivíduos que tem características de ambos os pais, obtendo-se desta forma um conjunto de parâmetros otimizados para a tabela de valores desejados. Porém, ocasionalmente, os parâmetros adquiridos podem convergir para um bom valor local que não são os melhores para a

aplicação como um todo. Para evitar isso, aplica-se o processo da *mutação*, que inclui um valor aleatório totalmente novo no conjunto de avaliação, que pode indicar para uma nova região de otimização.

Após várias interações, é obtido um conjunto de parâmetros otimizados para o controlador. Destaca-se que a grande vantagem deste processo consiste da otimização automática dos parâmetros das funções de pertinência sem depender do critério do projetista. No método simples que tem funcionamento intermediário aos métodos geracional e em regime, a avaliação de pais e filhos são realizadas em paralelo, possibilitando uma melhor escolha dos indivíduos. Aplicando este método, tem-se a vantagem de obter uma rápida convergência para o caso da solução estar próxima ou distante do ponto de partida.

Uma grande vantagem que a utilização do AG apresenta, é a de poder ser executado continuamente, a partir de medidas de desempenho da rede durante o seu funcionamento normal. Além disso, pode-se fazer planejamentos da utilização da rede para tráfegos específicos, a partir da geração de tráfegos tendo por base os dados históricos armazenados no DW. Desta forma, pode-se ajustar os parâmetros do controlador quando ocorrerem modificações na topologia ou no padrão do tráfego de entrada, mesmo durante a operação da rede.

Deve-se ressaltar que, quando o AG é usado durante a operação da rede, este não provoca queda no desempenho da rede. Isto ocorre porque a otimização pode ser realizada em equipamentos dedicados e os roteadores da rede somente executam o controlador fuzzy.

É bom lembrar que a otimização não precisa ocorrer em períodos muito curtos, pois quando se trata planejamento de redes, estes ocorrem normalmente em dias específicos, como feriados nacionais ou municipais. Desta forma, as mudanças que poderiam causar alterações na rede ocorrem na ordem de dias ou semanas, no pior dos casos poderiam ocorrer na ordem de horas, que seriam as horas de pico, mesmo assim é possível fazer o planejamento da melhor configuração da rede na madrugada do dia que antecede o dia com horário crítico.

Os mecanismos de funcionamento do AG adotados no trabalho em questão,

detalhados em Michalewicz [61], foram: a codificação do cromossomo foi um número real com 4 casas decimais, o mecanismo de funcionamento foi o método simples, o método de seleção foi o da roleta e o mecanismo de cruzamento foi o de multi-ponto.

### 4.11.2 Parâmetros de Configuração do AG

Pode-se afirmar que a escolha correta dos parâmetros de configuração dos AGs se traduz num dos aspectos mais importantes na estratégia dos AGs. Não existe uma metodologia genérica, comum a todas as aplicações, muito pelo contrário, cada aplicação tem seus próprios parâmetros a serem otimizados. Também deve ser destacado como grande fator de importância, o tempo de execução do AG, bem como os recursos computacionais despendidos.

São mostrados na tabela 4.1, os parâmetros do AG utilizados em nosso trabalho na otimização do controlador fuzzy. Nas próximas seções serão apresentadas as faixas de valores mais usuais para esses parâmetros. Vale destacar que foram utilizados valores dentro desta faixa, com seus devidos ajustes para melhoria de eficiência e eficácia.

Tabela 4.1: Parâmetros utilizados no algoritmo genético.

Parâmetro	Valor
Tamanho da população	500
Taxa de cruzamento	65%
Taxa de mutação	5%
Critério de parada	90% da população igual
Número de gerações	600

#### 4.11.2.1 Tamanho da população

Esta medida afeta diretamente no desempenho e eficiência dos AGs. Se for admitido como parâmetro para o AG uma população pequena, esta fornecerá uma cobertura pequena do espaço de busca do problema. Por outro lado, uma população

grande fornecerá uma maior cobertura do espaço de busca, sendo mais eficaz na prevenção de convergências prematuras para soluções locais. Entretanto, quando são usadas grandes populações, deve-se levar em conta a necessidade de uma maior capacidade de recursos computacionais, além de tempos mais longos para resolução de problemas que poderiam, algumas vezes, ser caracterizados como simples.

Desta forma, pode-se afirmar que a escolha mais adequada para o tamanho da população, que produz respostas rápidas e corretas, depende do problema e da sua formulação. Normalmente é utilizado na população um tamanho proporcional ao tamanho do cromossomo, ou seja, quanto maior for o cromossomo maior deverá ser o tamanho da população para se manter uma diversidade razoável. A literatura sugere populações dentro do intervalo de 50 a 100 cromossomos como boa relação de cobertura e desempenho [62].

No experimento proposto nesse trabalho, foram utilizados 500 indivíduos, pois este valor se mostrou como uma boa relação entre o espaço de cobertura e o tempo de resposta, tendo em vista a capacidade computacional disponível.

##### **4.11.2.2 Taxa de cruzamento**

Este valor define a probabilidade de ocorrer um cruzamento “*crossover*”, isto é, quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Entretanto, dependendo da aplicação, uma alta taxa de cruzamento pode substituir estruturas com boas aptidões mais rapidamente, perdendo assim boas oportunidades de chegar em uma solução ótima. Porém, um valor baixo para esta taxa pode tornar o algoritmo muito lento, resultando numa demora para encontrar a solução ideal. Grande parte da literatura recomenda uma taxa de cruzamento entre 50% a 95% [62].

No experimento proposto, foi usado como taxa de cruzamento o valor de 65%, pois este se mostrou razoável para a aplicação.

### 4.11.2.3 Taxa de mutação

Define a taxa de ocorrência da mutação. Uma taxa de mutação muito alta torna a busca do AG numa busca aleatória. Porém, uma baixa taxa de mutação diminui a inclusão de indivíduos novos na população, podendo provocar o encontro de resultados baseados em mínimos locais, tendo em vista que o espaço de busca é restringido.

Na literatura foram encontradas recomendações para a escolha da taxa de mutação baseado no tamanho do cromossomo da população. De Jong em [63], sugere uma taxa de mutação inversamente proporcional ao tamanho da população. Grande parte da literatura recomenda uma taxa de mutação entre 0,1% e 5% [62].

No experimento proposto, foi usado um alto valor para a taxa de mutação (5%), porém dentro da faixa recomendada. A escolha deste alto valor se justifica em função do método de seleção utilizado (roleta), pois este método de seleção pode produzir uma baixa diversidade na população e assim, não atingir o valor máximo global da função.

### 4.11.2.4 Número de gerações

Este número consiste na quantidade de populações criadas até chegar à resposta final definida pelo AG. Se for adotado um valor baixo, é possível rapidamente encontrar a solução ótima local. Porém longe da solução ótima global. Já se for adotado um valor alto, temos a garantia de chegar à solução ótima global. Entretanto, o tempo de execução para alcançar esta resposta pode ser longo, e no caso de problemas simples, novas gerações são criadas desnecessariamente após a solução ótima ter sido alcançada. Desta forma, dizemos que o valor ideal depende do problema e da sua formulação.

No experimento proposto, foi estabelecido um valor limite, válido para os problemas apresentados. Foi adotado como condição de parada do AG quando 90% da população for constituída de apenas um indivíduo, que será a resposta.

O programa JFS [53] permite apenas duas formas de encerrar o processamento: pelo tempo decorrido ou pela quantidade de gerações. Assim, teve-se que executar uma grande quantidade de gerações e verificar se o critério de fim tinha sido atingido, ou



seja, se pelo menos 90% da população estava constituída de um mesmo indivíduo. Nos testes efetuados, foi observado que 600 gerações foram suficientes para atingir este resultado para as otimizações realizadas. Desta forma, foi aplicado como condição de parada para o programa JFS a quantidade de 600 gerações.

### **4.12 Comentários**

Neste capítulo, foi apresentada a metodologia proposta para utilização da base de dados do DW no auxílio ao gerenciamento e planejamento de redes para provimento de QoS numa arquitetura DiffServ. Foi mostrada inicialmente a arquitetura geral de um nó DiffServ. Logo após foram apresentadas a arquitetura e a metodologia proposta neste trabalho. Foi descrita a metodologia para construção do controlador fuzzy, juntamente com sua definição da política de gerenciamento, o mapeamento da política em parâmetros do controlador fuzzy e a construção do controlador propriamente dita. Foram apresentadas em seguida, as estruturas de DW e DM necessárias para prover suporte a decisão, e foi mostrada a utilização do AG, que traz como benefício a otimização dos parâmetros do controlador.

## Capítulo 5

### Ambiente de Simulação e Resultados

Este capítulo apresenta o ambiente de simulação utilizado para validar a metodologia descrita no capítulo anterior.

Nesse trabalho foi criada uma aplicação para duas classes de tráfego de telefonia convencional típicas de uma empresa de telecomunicações. É considerado como fator de divisão de classes uma *flag* “tipo” que assume um dos valores: avançado ou convencional. Foi considerado o tipo avançado como o tráfego pertencente à classe EF (*Expedited Forwarding*) e o tipo convencional como o tráfego da classe BE (*Best Effort*). Destaca-se que a classe EF é ideal para aplicações com restrição de tempo de propagação, pois esta oferece o menor retardo possível, sendo bastante aconselhável para acomodar tráfegos típicos de telefonia IP. Já a classe BE que no modelo desse trabalho suporta o tráfego convencional, não possui garantias de retardo, sendo bastante peculiar para representar o tráfego de dados de uma rede IP (*Internet*).

É bom lembrar que o foco deste trabalho é de priorizar o tráfego da classe EF procurando não afetar o tráfego BE, sendo este prejudicado apenas quanto houver escassez de recursos na rede.

Esse capítulo está organizado em 5 seções. Na seção 5.1 é apresentado o ambiente de simulação. Na seção 5.2 são mostradas as topologias utilizadas. Na seção 5.3, é descrito o modelo de tráfego aplicado em simulação. As medidas de avaliação são apresentadas na seção 5.4. Por fim, na seção 5.5 são mostrados os resultados obtidos.

## 5.1 Ambiente de simulação

Nesta seção são apresentadas as ferramentas utilizadas em nosso trabalho. Como simulador de redes, foi utilizado o Network Simulator – ns-2 [64]. A ferramenta PonderToolkit foi usada para especificar as políticas administrativas e para verificar a correção dessa especificação e a ocorrência de conflitos. Foi usada a ferramenta JFS [53] para construir o controlador fuzzy a partir da especificação de políticas e para realizar a otimização dos parâmetros. Como ferramenta de DM, foi adotado o Pilot [56] para fornecer suporte a decisão.

### 5.1.1 Network Simulator – ns

Para validar nosso modelo, foi usada a versão 2.1b8a [64] da plataforma de simulação *Network Simulator (ns)*, tendo em vista que esta versão disponibiliza alguns dos recursos de arquitetura DiffServ indispensáveis para o trabalho proposto, recursos estes originários do modelo proposto por Pedia e Ethridge [65].

Como funcionalidades do modelo DiffServ presentes nesta versão do ns, temos: PHBs, *Assured Forwarding (AF)*, *Expedited Forwarding (EF)*, *Class Selector* e *Best Effort (BE)*; condicionadores, *TSW 2 Color Market*, *TSW 3 Color Market*, *Single Rate 3CM*, *Two Rate 3CM* e *Token Bucket*; gerenciamento ativo de filas, *RIO-C (Coupled)*, *RIO-D (Decoupled)*, *WRED* e *DropTail*; escalonadores, *Weighted Round Robin (WRR)*, *Weighted Interleaved Round Robin (WIRR)*, *Round Robin (RR)* e *Priority (PRIO)*.

Para atender as necessidades da aplicação proposta foi preciso acrescentar novas funcionalidades ao modelo DiffServ. Essas novas funcionalidades, desenvolvidas por Fernandez em [5], permitem a alteração de alguns parâmetros de configuração durante a simulação, como o peso do escalonador. Na versão original, o peso do escalonador é definido obrigatoriamente na configuração do nó, não permitindo alteração após o início da simulação. As funcionalidades originais do modelo DiffServ utilizadas são as seguintes: escalonador WRR, filas *DropTail* em ambas as classes e condicionador

*Token Bucket* para os fluxos da classe EF.

### 5.1.2 Ponder Toolkit

Como mostrado na seção 4.5 do capítulo anterior, a linguagem Ponder [49, 50] é declarativa, orientada a objetos e oferece ao usuário uma interface simples para especificação de políticas. A ferramenta *Ponder Toolkit* é composta por um editor de políticas, um compilador da linguagem Ponder, um gerador de objetos em Java e uma interface de gerenciamento do repositório. Foram usados neste trabalho a versão 1.0.1 do Ponder Policy Editor e a versão 2.1 do Ponder Compiler.

Como não era objetivo deste trabalho, implementar uma plataforma de gerenciamento de políticas, foram utilizadas apenas as ferramentas de edição e compilação disponibilizadas no conjunto Ponder Toolkit. Desta forma, foi realizado manualmente o mapeamento de políticas em Ponder para as funções de pertinência e base de regras do controlador fuzzy.

### 5.1.3 JFS

A ferramenta JFS, de Mortensen [53], consiste num ambiente de programação para desenvolvimento e execução de programas com funções de lógica fuzzy e técnicas de aprendizagem.

Esta ferramenta foi utilizada como ambiente para desenvolvimento do controlador fuzzy, foi possível especificar as funções de pertinência, regras de inferência e defuzificador. Além disso, foram utilizadas as seguintes ferramentas de otimização disponibilizadas: algoritmo de Wang-Mendel e AG.

A implementação do controlador usando a ferramenta JFS resultou em uma biblioteca de funções em código C, que são integradas ao simulador. O código gerado por essa ferramenta possibilitou implementar as funções do controlador fuzzy com código eficiente.

### 5.1.4 Ferramenta Pilot

Como apresentado na seção 4.9 do capítulo anterior, o Pilot consiste numa ferramenta de DM que possibilita ao usuário realizar consultas a base de dados do DW. Neste trabalho, esta ferramenta se mostrou de grande valia, pois a partir dela foi possível identificar perfis de tráfego em diferentes períodos de tempos e em regiões distintas. Desta forma, pode-se realizar o planejamento das redes e assim configurá-las para suportar seus diferentes padrões de tráfego.

Foi escolhido o Pilot como ferramenta de DM pela disponibilidade cedida e por atender as requisições necessárias em nosso projeto de gerenciamento e planejamento de redes, como: boa performance no gerenciamento de informações e capacidade de realizar automaticamente atualizações regulares em sua base de dados [56].

## 5.2 Topologias de simulação

Neste ambiente de simulação, foram atribuídas diferentes topologias, fictícias, para suportar o fluxo de telefonia de determinados Estados do território nacional. Nesse trabalho são propostas duas topologias com diferentes quantidades de roteadores. A topologia simples, mostrada na seção 5.2.1, foi associada ao Estado de Pernambuco, tendo em vista que este é um Estado que não possui grandes demandas de tráfego. A topologia complexa, mostrada na seção 5.2.2, foi atribuída ao Estado de São Paulo, em função da sua grande demanda de recursos de rede.

A escolha dessas diferentes topologias associadas aos determinados Estados foi efetuada em função de uma consulta feita ao DM referente ao volume do fluxo de dados de telefonia em âmbito nacional, para o trimestre relativo aos meses de Setembro, Outubro e Novembro de 2004, separada pelas regiões (Estados). O resultado desta consulta está apresentado na Figura 5.1.

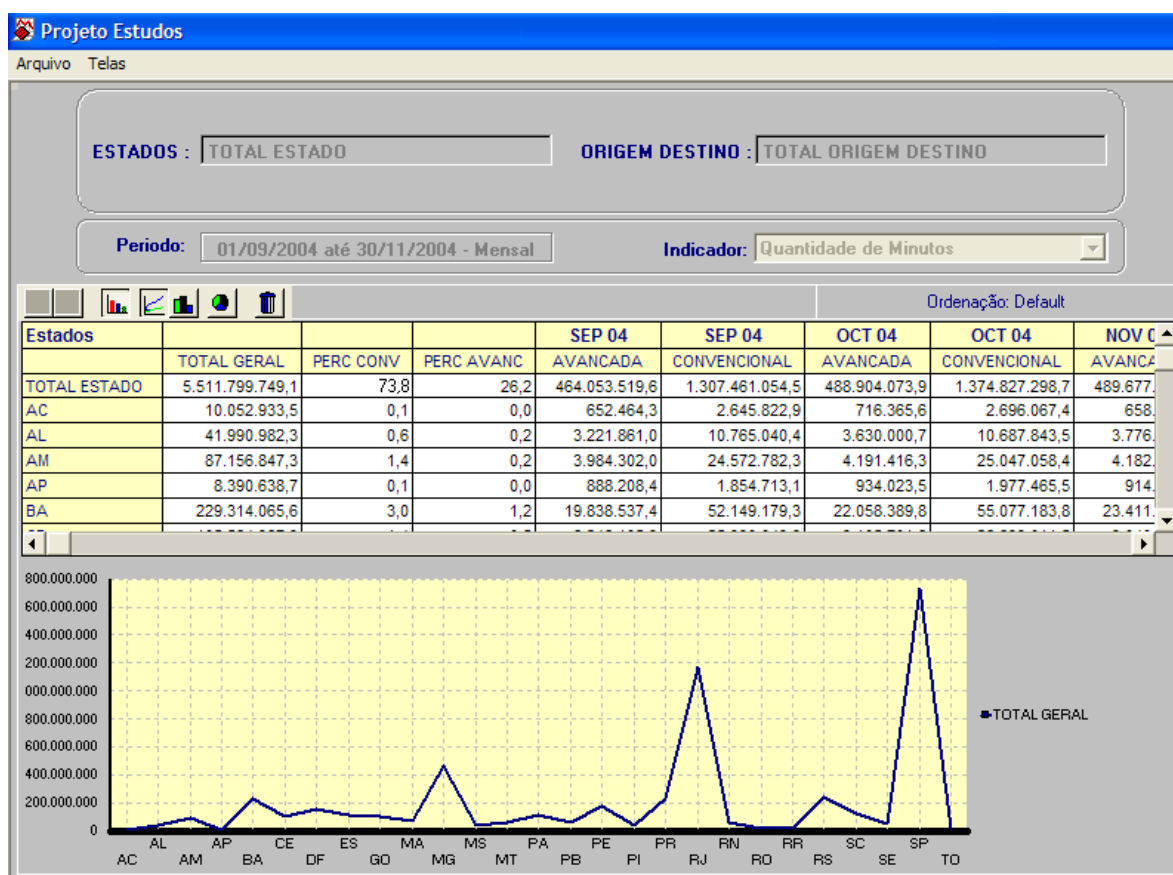


Figura 5.1: Volume de tráfego por Estado, de Setembro a Novembro de 2004

Foi tomado como base um determinado dia específico do ano para cálculo do número de fontes para as diferentes topologias, já que não havia um grande volume de dados históricos na base do DM, o que proporcionaria uma maior capacidade de pesquisa.

### 5.2.1 Topologia simples

A topologia simples mostrada na Figura 5.2, forma um domínio DiffServ com cinco nós, sendo três de borda e dois de núcleo. Nessa topologia existem dois pontos de congestionamento, o primeiro entre dois nós de borda e o segundo entre os nós de núcleo.

Na Figura apresentada, não foram representadas todas as fontes para não prejudicar a visualização. Foram utilizados como dados de entrada os padrões extraídos

do DM para o Estado de Pernambuco para um dia de semana com feriado, foi tomado como base o dia 07 de Setembro de 2004 (terça-feira).

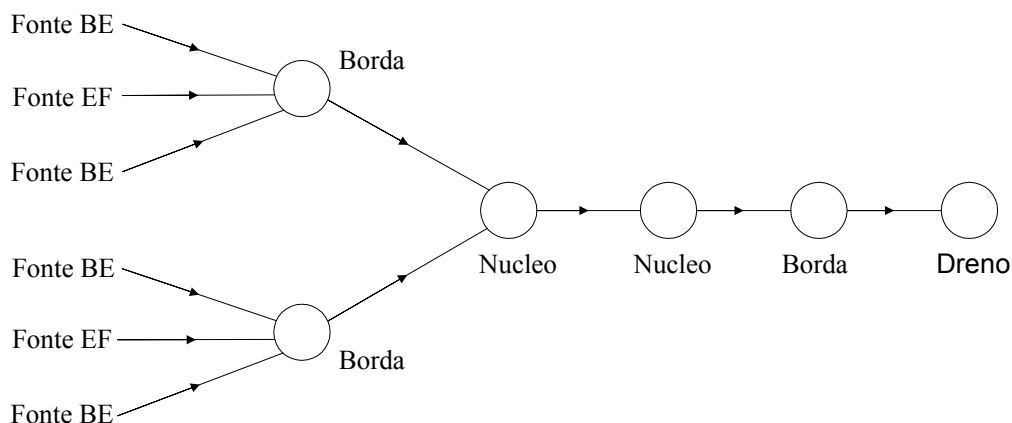


Figura 5.2: Topologia simples (Estado de Pernambuco).

### 5.2.2 Topologia Complexa

A topologia complexa mostrada na Figura 5.3, forma um domínio DiffServ com 40 nós, sendo 30 de núcleo e 10 de borda, onde temos 5 nós fonte e 5 nós de “dreno” os quais, juntos, constituem os 10 nós de borda. Esta topologia foi criada com o pacote gtm, incluído na distribuição do ns [64].

Zegura [66] mostra que se pode criar topologias randômicas com métricas de redes reais, porém com o tamanho e a complexidade desejados. Assim, foi adotado um tipo de distribuição para definir as posições dos nós e das conexões, onde a probabilidade de dois nós estarem conectados decresce conforme aumenta a distância entre eles. Conforme os trabalhos [67, 68], foram escolhidos os tipos de distribuição e parâmetros mais apropriados para modelagem da Internet. Assim, nesta topologia foram usadas as métricas a seguir.

1. **Tipo** - Esta métrica consiste no tipo da distribuição, isto é, no algoritmo utilizado para calcular a probabilidade de existir uma conexão entre dois nós. O algoritmo utilizado foi o Exponencial, detalhado em [69].

2. **Grau Médio** - É o grau médio de nós, ou seja, a relação entre nós e conexões no grafo. O valor utilizado foi 4.133.
3. **Diâmetro** - Consiste no diâmetro do grafo, ou seja, a distância do par de nós mais afastados. Três tipos de distância são avaliados, que são: quantidade de saltos (hh), distância Euclideana (ll) e distância total (hl), isto é, saltos mais distância, os valores utilizados foram 6, 95 e 102, respectivamente.
4. **Bicomp** - É o número de componentes biconectados, ou seja, a quantidade máxima de conexões entre três nós dentro de um mesmo círculo. O valor utilizado foi 3.

Maiores detalhes sobre métricas de topologias de redes podem ser encontrados em Zegura [66] e Calvert [68].

Nesta topologia foram usados como dados de entrada os padrões extraídos do DM para o Estado de São Paulo para um dia de semana comum, tomando como base o dia 01 de Outubro de 2004 (sexta-feira).

A Figura 5.3 mostra a topologia complexa associada ao Estado de São Paulo. Deve-se ressaltar que a posição real dos nós poderá não ser a mostrada na Figura, com o objetivo de melhorar a visualização da topologia.

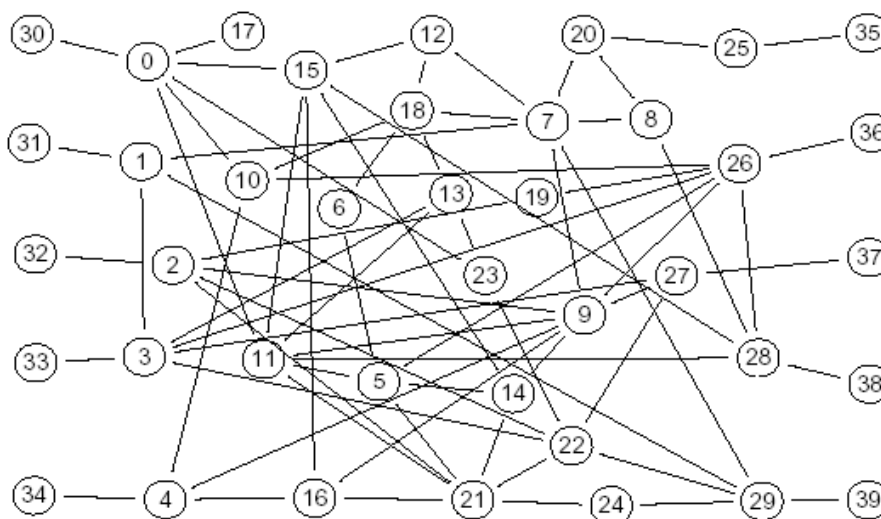


Figura 5.3: Topologia complexa (Estado de São Paulo).



## 5.3 Modelo de tráfego de simulação

Foi implementada a aplicação de telefonia IP com tráfegos CBR e On-Off Exponencial (OO), ambas sobre o protocolo UDP. O tráfego CBR tem comportamento determinístico e exige mais banda da rede, enquanto o tráfego On-Off Exponencial se aproxima mais de uma conversação normal. Sabendo que a taxa média das fontes On-Off é bem menor que do tráfego CBR, foi utilizado um fator de ajuste para o tráfego On-Off para obter o congestionamento desejado. Deste modo, foi direcionado o tráfego de voz para a classe EF e o tráfego concorrente para a classe BE, sendo ambos CBR/UDP.

Com o intuito de configurar as redes a partir de dados de entrada provindos da base de dados do DW, foi definida a quantidade de fontes de tráfego de maneira controlada para cada topologia, uma vez que já foram feitas as consultas necessárias para pré-determinar o perfil de tráfego a ser otimizado. Assim sendo, são mostrados nas seções 5.3.1 e 5.3.2 os gráficos relativos à quantidade de fontes utilizadas para cada topologia, apresentadas nas seções 5.2.1 e 5.2.2.

### 5.3.1 Modelo de fontes de tráfego para a topologia simples

É mostrada na Figura 5.4, a quantidade de fontes de tráfego CBR para as classes EF e BE respectivamente, utilizadas durante o período de simulação de 120 segundos. Da mesma forma, é apresentada a quantidade de fontes de tráfego On-Off na Figura 5.5, durante o período de 120 segundos de simulação para as classes EF e BE. Assim, foi caracterizado um perfil de tráfego a ser otimizado para a rede que afixamos para o Estado de Pernambuco.

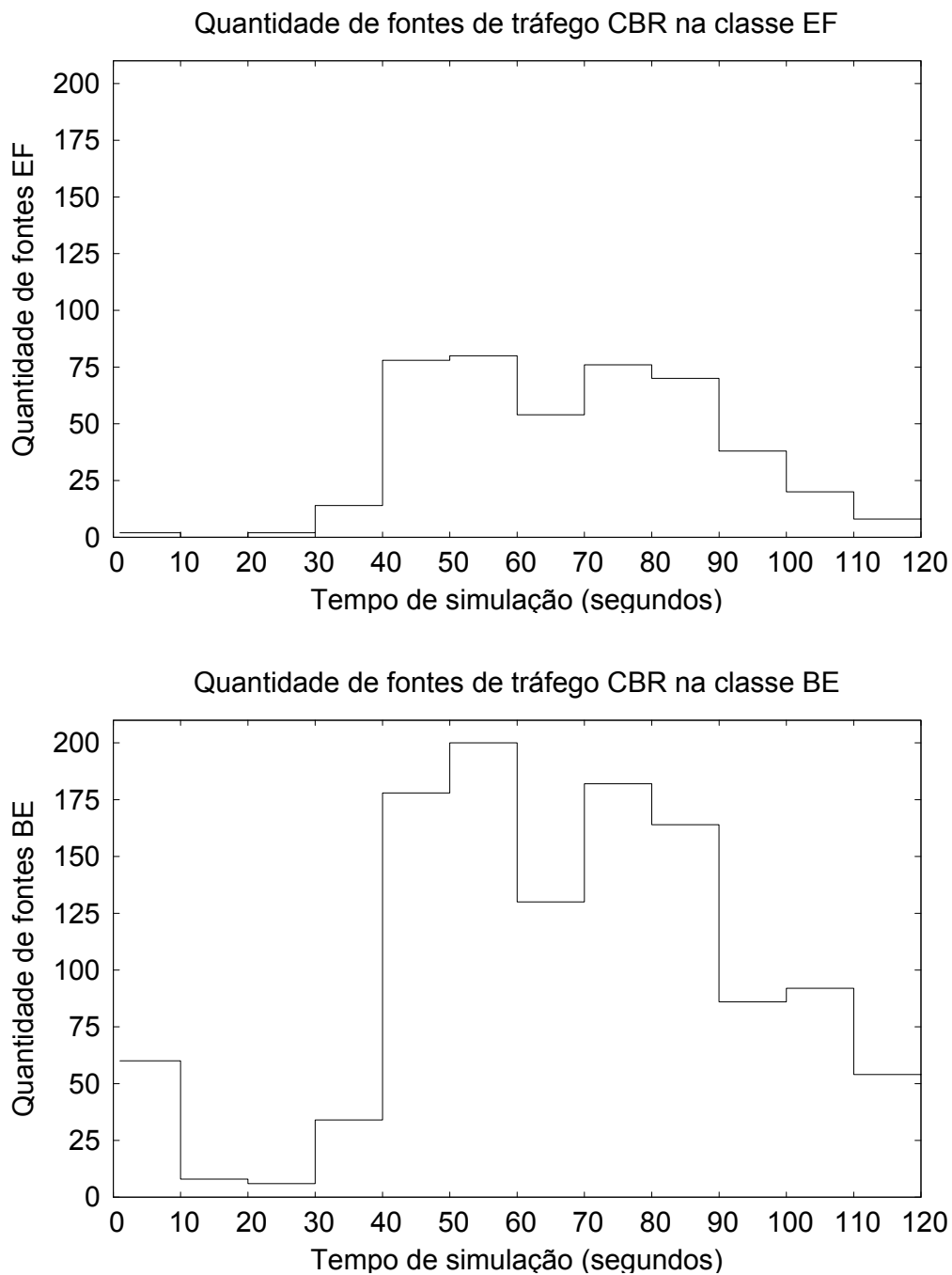


Figura 5.4: Quantidade de fontes de tráfego CBR (EF e BE) durante a simulação

É apresentada na tabela 5.1, a relação dos 120 segundos simulados com o período do dia correspondente (24 horas).

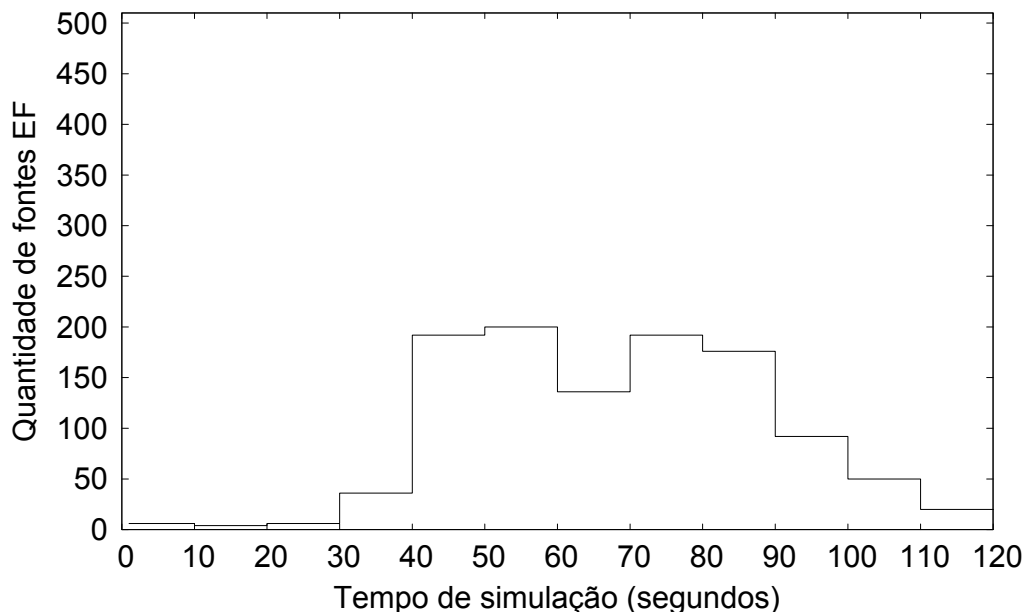
Tabela 5.1: Relação dos segundos com horas do dia

<b>Segundos (simulados)</b>	00...10	10...20	20...30	30...40	40...50	50...60
<b>Horas (associadas)</b>	00..01:59	02..03:59	04..05:59	06..07:59	08..09:59	10..11:59

<b>Segundos (simulados)</b>	60...70	70...80	80...90	90...100	100...110	110...120
<b>Horas (associadas)</b>	12..13:59	14..15:59	16..17:59	18..19:59	20..21:59	22..23:59

Cada fonte CBR foi definida com 64 Kbps. Para as fontes On-Off, foi usada uma taxa de 64 Kbps, com tempo de rajada de 400ms e tempo de silêncio de 600ms, representando uma taxa de 25,6 Kbps. Utilizamos pacotes de 576 bytes, que corresponde a 91,7% dos pacotes de um codificador G.711 (PCM) a 64 Kbps [70].

Quantidade de fontes de tráfego OO na classe EF



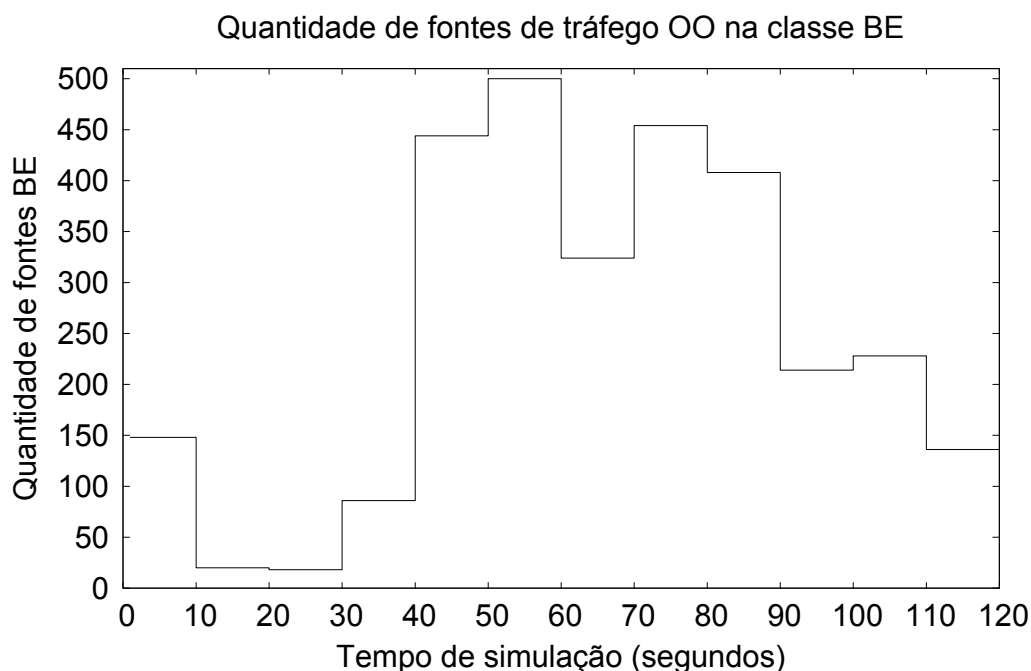


Figura 5.5: Quantidade de fontes de tráfego On-Off (EF e BE) durante a simulação

Observa-se que a quantidade de tráfego CBR e On-Off da classe BE é muito maior que o da classe EF, devido a baixa utilização da rede para o tráfego avançado, que nesse experimento está representando o tráfego de telefonia IP. A baixa utilização do tráfego avançado ocorre por se tratar de um dia de feriado, tendo em vista que grande parte do tráfego avançado está associado ao período comercial.

Pode-se notar a grande variação do tráfego no decorrer do dia. No período da madrugada, de 0 às 6, a rede fica praticamente ociosa, com exceção do tráfego convencional entre 0 e 2 da manhã. Das 6 às 8, existe um pequeno aumento na utilização da rede para ambas as classes de tráfego. Das 8 às 12 e das 14 às 18, existe um grande aumento no volume de tráfego EF e BE (horário comercial). Por outro lado, no período entre 12 e 14 (horário de almoço) há uma queda de tráfego, principalmente para a classe BE. Após às 18, notamos uma queda gradativa do volume de tráfego demandado, com exceção do período entre 20 e 22 onde temos um pequeno aumento de tráfego convencional (classe BE).

### 5.3.2 Modelo de fontes de tráfego para a topologia complexa

As Figuras 5.6 e 5.7 apresentam a quantidade de fontes de tráfego CBR e On-Off utilizadas durante o período de simulação de 12 minutos para a topologia complexa. A quantidade de fontes CBR variou de 0 a 130 para a classe EF e de 0 a 190 para a classe BE, enquanto a quantidade de fontes On-Off variou de 0 a 320 para a classe EF e de 0 a 480 para a classe BE, justamente porque a taxa média de tráfego On-Off considerada é aproximadamente 2,5 vezes menor. Deste modo, foi usado o perfil de tráfego extraído do Estado de São Paulo para ser otimizado.

Para ambas as classes, cada fonte CBR foi definida com 64 Kbps, ou seja, semelhante a um canal de voz PCM. De forma análoga ao tráfego On-Off da topologia simples, foi usado uma taxa de 64 Kbps com tempo de rajada de 400ms e tempo de silêncio de 600ms, representando uma taxa média de 25,6 Kbps. Foram utilizados pacotes de 576 bytes para ambos os tipos de tráfego de ambas as classes, que corresponde a 91,7% dos pacotes de um codificador G.711 (PCM) a 64 Kbps [70].

É apresentada na tabela 5.2, a relação dos 12 minutos simulados com o período do dia correspondente (24 horas)

Tabela 5.2: Relação dos minutos com horas do dia

<b>Minutos (simulados)</b>	0...1	1...2	2...3	3...4	4...5	5...6
<b>Horas (associadas)</b>	00..01:59	02..03:59	04..05:59	06..07:59	08..09:59	10..11:59

<b>Minutos (simulados)</b>	6...7	7...8	8...9	9...10	10...11	11...12
<b>Horas (associadas)</b>	12..13:59	14..15:59	16..17:59	18..19:59	20..21:59	22..23:59

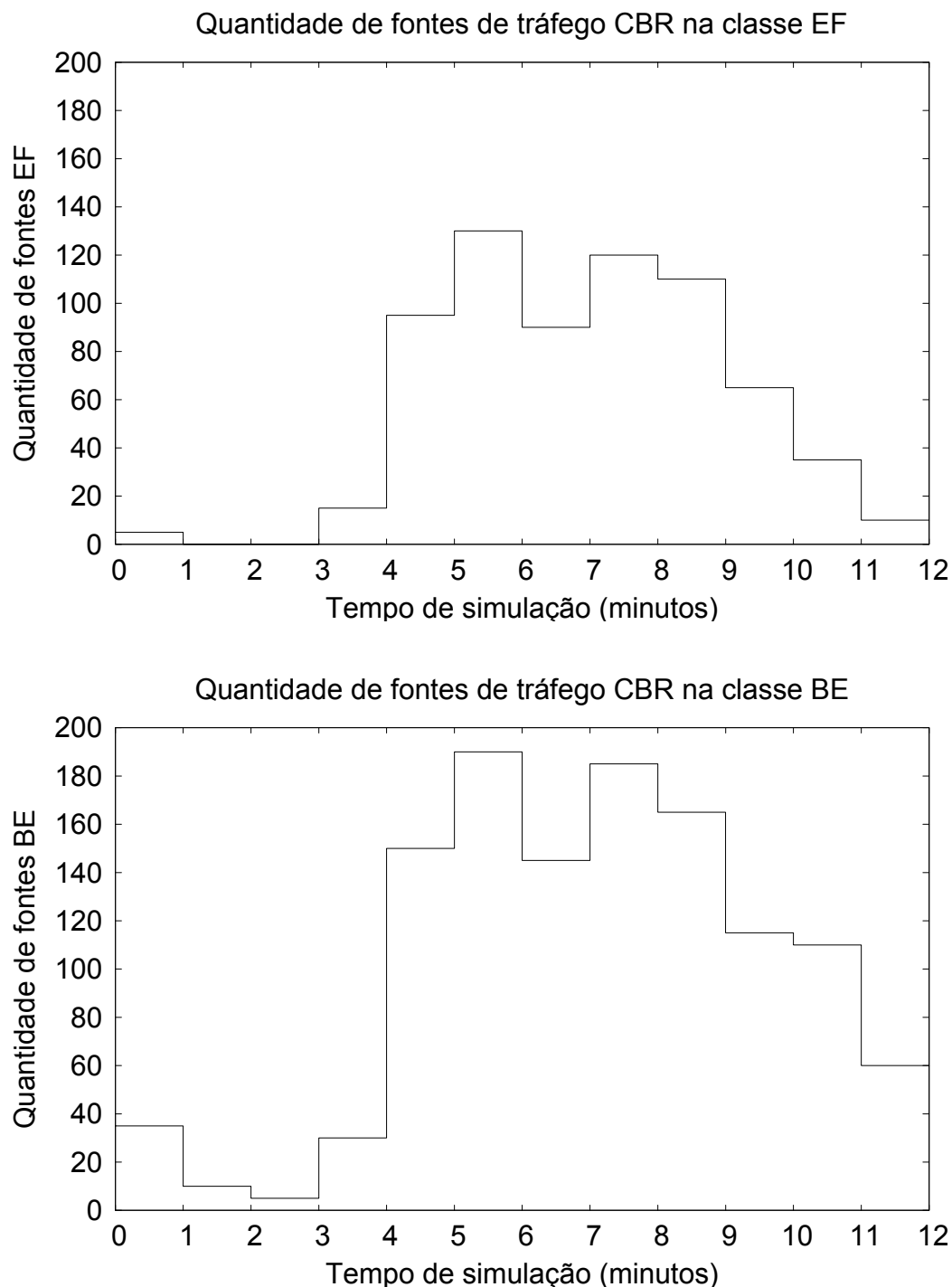


Figura 5.6: Quantidade de fontes de tráfego CBR (EF e BE) durante a simulação

O tempo de propagação de cada canal de 2 Mbps foi definido como 10ms. Todas as filas possuem tamanho de 50 pacotes. O modelo de simulação usou escalonador WRR, filas Drop Tail em ambas as classes e condicionador Token Bucket na classe EF

em todos os nós de borda (a classe BE não foi condicionada).

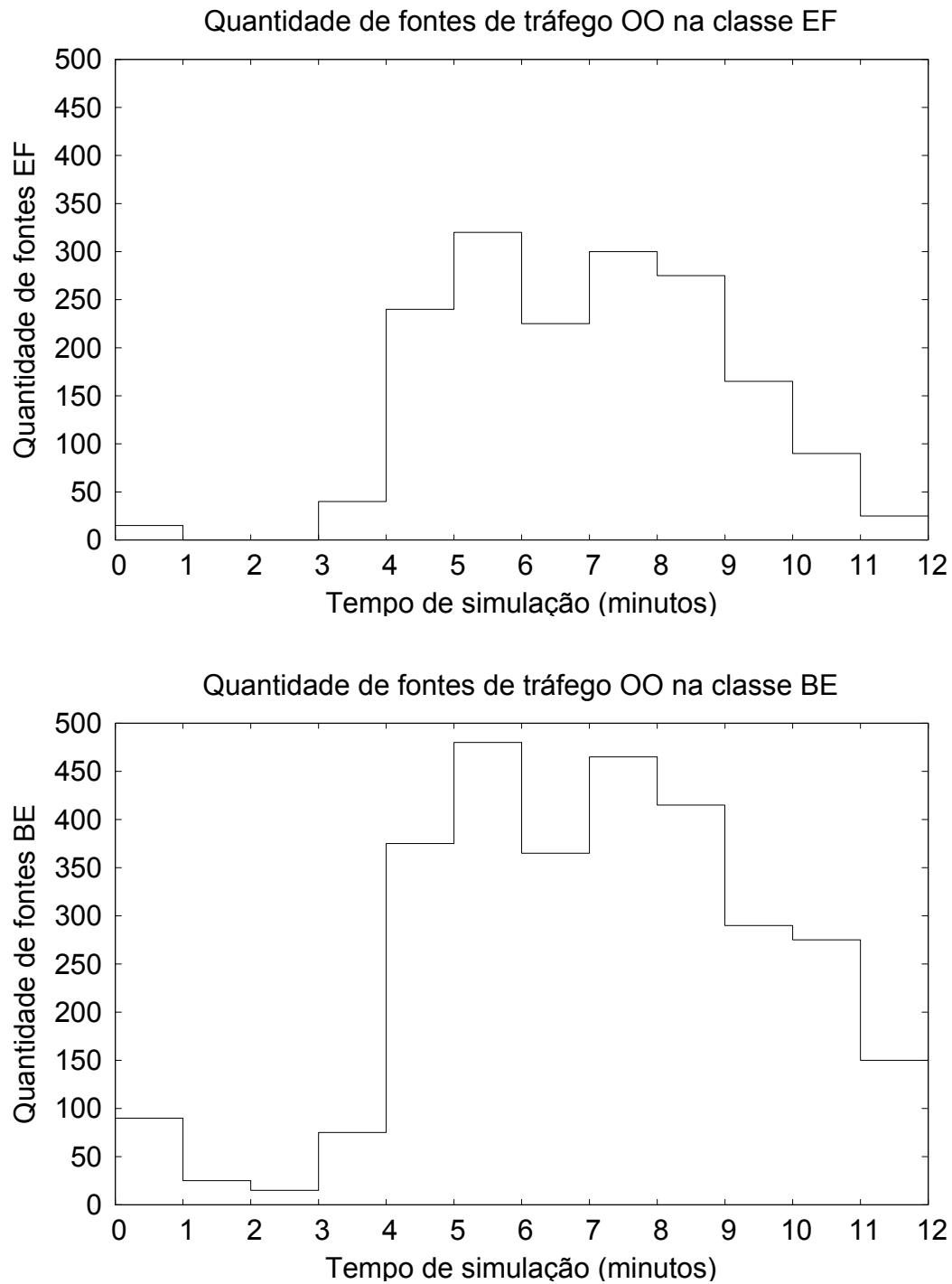


Figura 5.7: Quantidade de fontes de tráfego On-Off (EF e BE) durante a simulação

## 5.4 Medidas para avaliação

Como este trabalho se propõe a oferecer garantias de QoS, foram verificadas as seguintes medidas:

1. Retardo de um pacote da classe EF.
2. Perdas de pacotes da classe EF no domínio DiffServ.
3. Retardo da classe EF para os percentis 50, 90 e 95.
4. Variação de retardo (*jitter*) da classe EF para os percentis 50, 90 e 95.
5. Taxa de perda de pacotes da classe EF.
6. Taxa de perda de pacotes da classe BE.

Foram realizadas medidas para as fontes de tráfego CBR e On-Off exponencial. O percentil 50, também conhecido como mediana, significa que 50% dos pacotes têm retardo e variação de retardo menor que o valor apresentado. Da mesma forma, os percentis 90 e 95 significam retardo ou variação do retardo máximo para 90% e 95% dos pacotes, respectivamente.

A taxa de perda de pacotes das classes EF e BE corresponde à quantidade de pacotes descartados em relação à quantidade de pacotes transmitidos.

## 5.5 Resultados

Nesta seção, são mostrados os resultados obtidos após a simulação nas diferentes topologias. Com o intuito de verificar a viabilidade do processo de configuração das redes utilizando AG para provisão de QoS a partir de dados históricos, foram realizadas medições na rede cuja topologia adotada foi a simples, nas situações: i) com a base de regras estabelecida pelo projetista e otimizada pelo algoritmo de Wang-Mendel, e ii) com os mesmos atributos da situação anterior, e com a rede reconfigurada a partir do mecanismo de otimização do AG para suportar dados históricos.



Foram estabelecidos comparativos para as medições apresentadas na seção 5.4 nas situações com e sem a utilização do controlador fuzzy para as topologias simples e complexa. Foram usados os modelos de tráfegos determinísticos apresentados nas seções 5.3.1 e 5.3.2, para as topologias simples e complexa, respectivamente.

Ressalta-se que no processo de otimização, deve-se definir com critério as variáveis a serem otimizadas, pois o otimizador por AG não verifica o atendimento a uma determinada política, podendo assim produzir funções de pertinência errôneas, ou seja, funções que extrapolem os limites contratados (SLA) do domínio. Para evitar este problema, foram estabelecidos limites, definindo parâmetros não otimizáveis na descrição do controlador fuzzy.

São apresentados nas seções 5.5.1 e 5.5.2, os resultados das medições realizadas nas topologias simples e complexa, respectivamente.

### **5.5.1 Resultados da topologia simples**

Para avaliar o desempenho do controlador fuzzy e o ganho do processo de reconfiguração da rede nesta topologia simples, apresentada na seção 5.2.1, são comparados os resultados dos parâmetros de QoS obtidos, sem controlador, com o controlador fuzzy e regras verificadas pelo algoritmo de Wang-Mendel, e com o controlador fuzzy com base de regras verificadas pelo algoritmo de Wang-Mendel e com a rede reconfigurada, a partir do processo de otimização por AG com dados históricos extraídos do DW.

O intuito de apresentar, nesta topologia, os resultados com o controlador fuzzy apenas com a base de regras verificadas com Wang-Mendel, está em mostrar a importância do processo de reconfiguração da rede para um perfil de tráfego específico, para tratar perfis de tráfego típicos de telefonia de uma determinada região. Como mencionado na seção 5.2, a região escolhida para esta topologia foi o Estado de Pernambuco.

Todas as simulações iniciaram com uma configuração no escalonador de 50% da banda de saída para cada uma das classes EF e BE. Com o intuito de eliminar medidas

de desempenho com a rede vazia, a seqüência de medidas foi iniciada sempre 5 segundos após o início do tráfego. O tempo exibido nos gráficos não considera os tempos de propagação das conexões que, por serem constantes, foram subtraídos para melhorar a visualização.

O intervalo de atuação do controlador fuzzy utilizado nas simulações foi de 1 segundo, conseqüentemente foi usado este tempo para colher as amostras dos resultados.

### **5.5.1.1 Retardo fim-a-fim da classe EF**

Esse parâmetro indica o retardo dos pacotes pertencentes à classe EF desde a fonte até o destino do tráfego. Os resultados de retardo do tráfego CBR sem controlador, do tráfego CBR com controlador otimizado com Wang-Mendel e do tráfego CBR a partir da rede reconfigurada com AG e com suas funções de pertinência validadas por Wang-Mendel são apresentadas respectivamente nos gráficos das Figuras 5.8, 5.9 e 5.10.

Nota-se um pico de retardo no intervalo de 40 a 60 segundos de simulação, que corresponde exatamente ao período de 8 às 12 horas do dia, ou seja, parte da manhã do horário comercial. Pode-se constatar que, mesmo numa região com pouca intensidade de movimentos comerciais e num dia de feriado, este período do dia apresentou maior volume de tráfego.

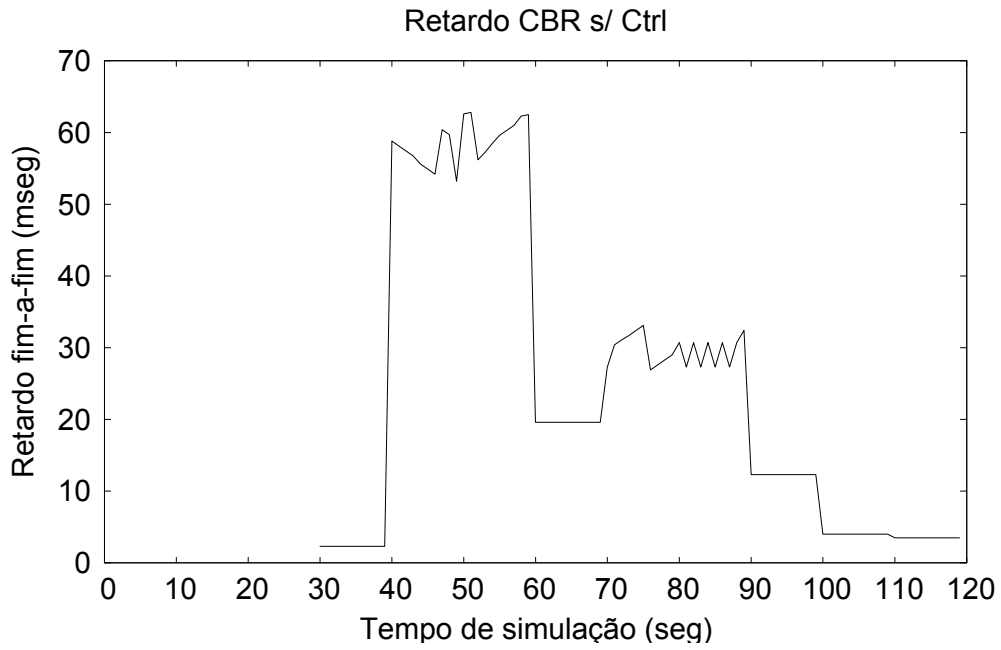


Figura 5.8: Retardo fim-a-fim da classe EF s/ o controlador

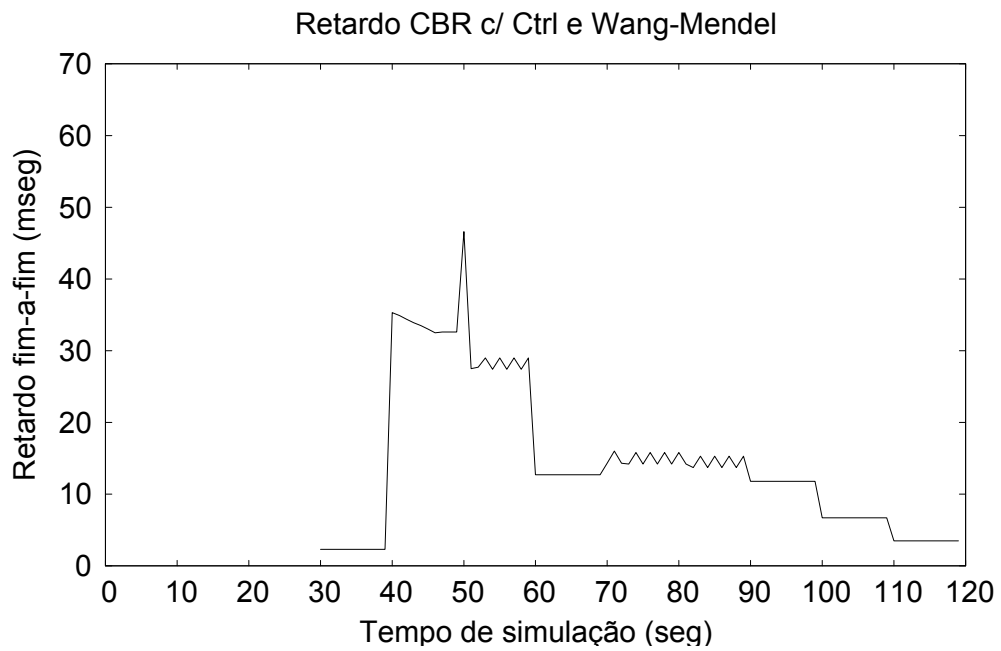


Figura 5.9: Retardo fim-a-fim da classe EF c/ controlador e Wang-Mendel

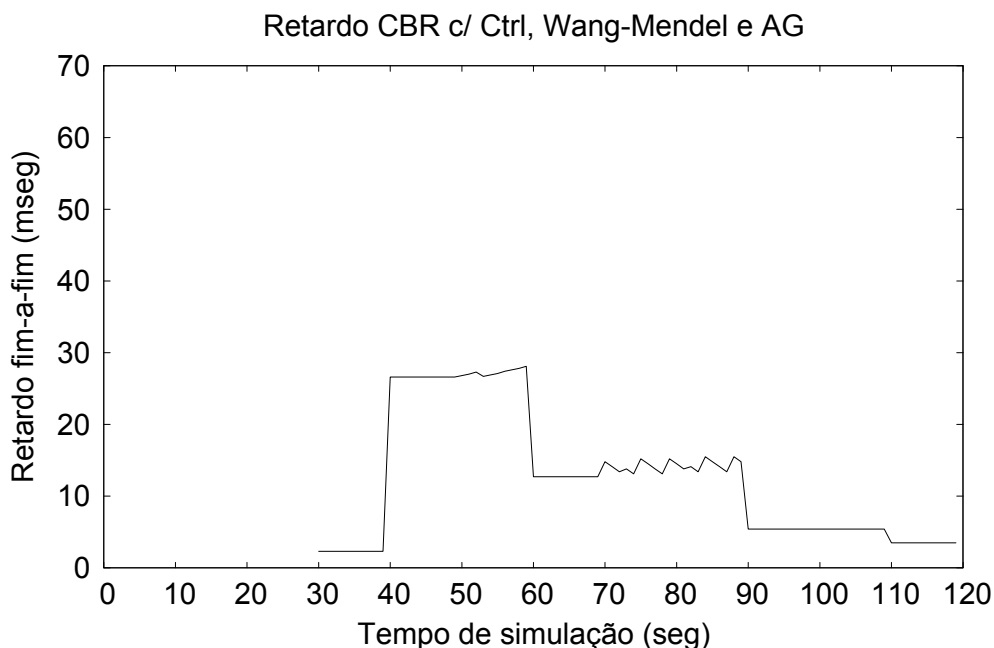


Figura 5.10: Retardo fim-a-fim da classe EF c/ controlador, Wang-Mendel e AG

Para o tráfego On-Off exponencial, os resultados obtidos de retardo fim-a-fim de um domínio DiffServ sem controlador, com controlador fuzzy e base de regras verificadas por Wang-Mendel e com o uso do controlador fuzzy otimizado com Wang-Mendel e AG são apresentados respectivamente nos gráficos da Figuras 5.11, 5.12, 5.13. Pode-se observar que neste tipo de tráfego, o retardo varia bastante ao longo da simulação, justamente em função da distribuição exponencial das fontes de tráfego que entram no domínio.

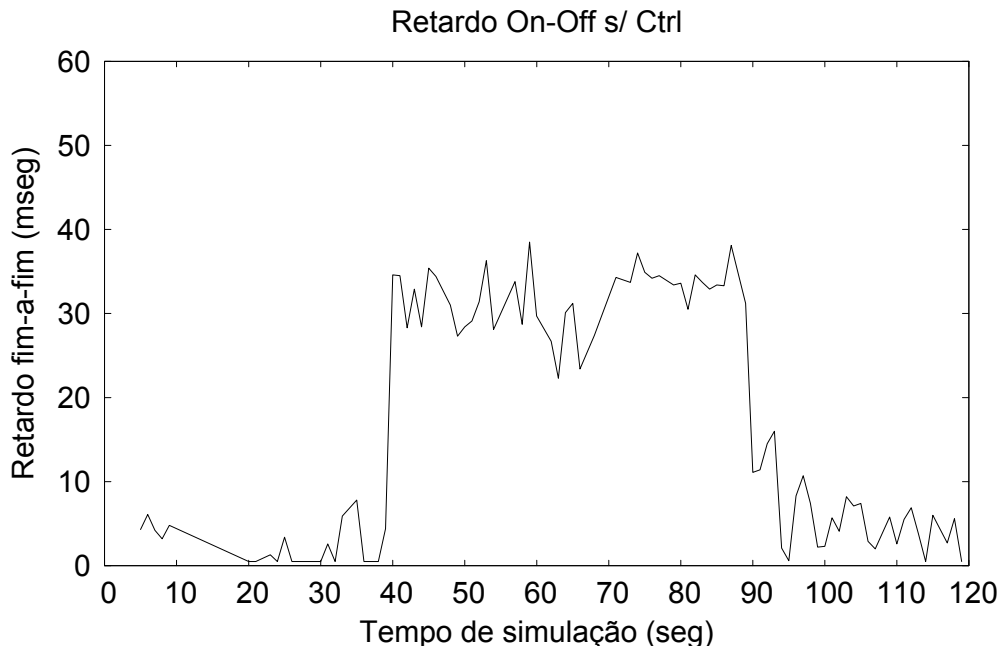


Figura 5.11: Retardo fim-a-fim da classe EF s/ controlador

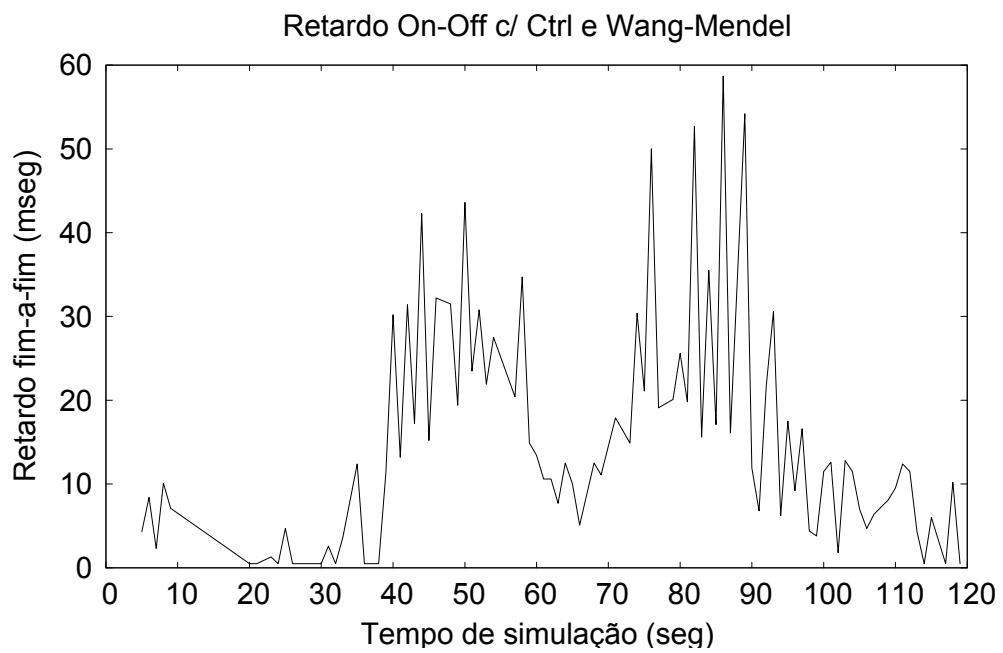


Figura 5.12: Retardo fim-a-fim da classe EF c/ controlador e Wang-Mendel

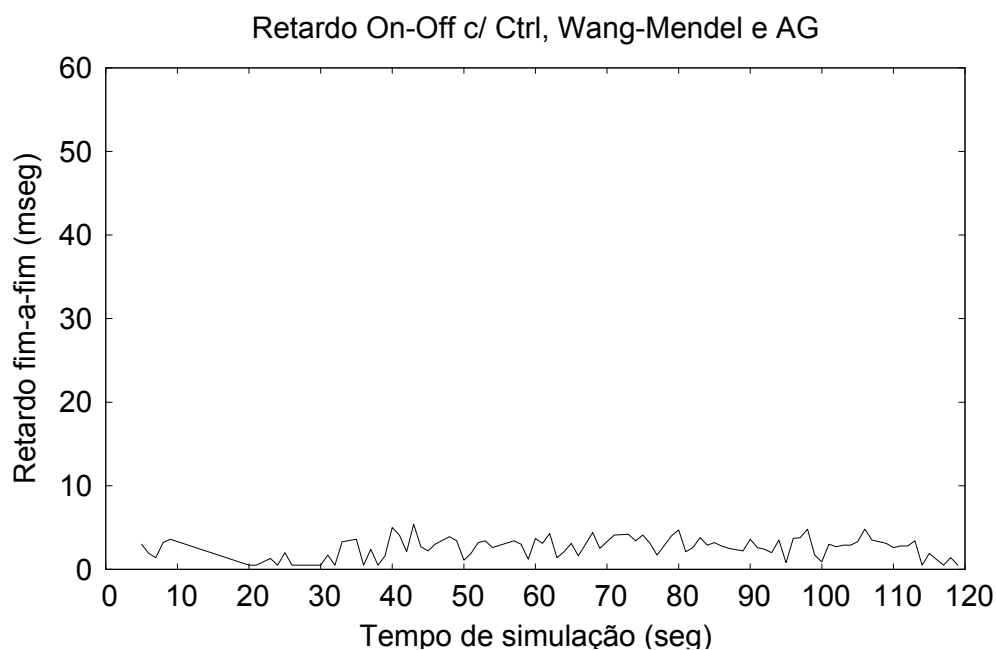


Figura 5.13: Retardo fim-a-fim da classe EF c/ controlador, Wang-Mendel e AG

Pode-se notar uma melhora significativa no retardo dos pacotes da classe EF quando é usado o controlador fuzzy em uma rede reconfigurada por AG com parâmetros de dados históricos. Principalmente no tráfego On-Off exponencial, que representa um tráfego mais próximo a uma conversação.

São apresentados, na tabela 5.3, os valores de medidas de retardo e variação do retardo médio (*jitter*) para os percentis 50, 90 e 95. São mostramos os resultados de simulação de tráfego CBR e On-Off sem o controlador, com controlador fuzzy e Wang-Mendel, e com controlador fuzzy, Wang-Mendel e AG.

Tabela 5.3: Retardo e variação de retardo da classe EF na topologia simples (ms)

Tráfego	Média		Percentil 50		Percentil 90		Percentil 95	
	Retardo	<i>Jitter</i>	Retardo	<i>Jitter</i>	Retardo	<i>Jitter</i>	Retardo	<i>Jitter</i>
<b>CBR s/ Ctrl</b>	24.4	1.6	20.4	0.4	55.4	5.4	61.1	7.3
<b>CBR c/ Ctrl e WM</b>	14.1	1	11.9	0.4	29.2	0.4	33.8	4.2
<b>CBR c/ Ctrl, WM e AG</b>	13.2	0.9	11.4	0.4	27.7	0.4	30.2	3.3
<b>OO s/ Ctrl</b>	24.9	1.7	30	0.7	37.7	5	39.8	6.5
<b>OO c/ Ctrl e WM</b>	17.8	2.1	15.1	0.8	35.9	6	43.3	8.1
<b>OO c/ Ctrl, WM e AG</b>	3.2	0.7	3.1	0.3	4.8	1.6	5.3	2.1

### 5.5.1.2 Descarte na classe EF

São apresentados a seguir, os gráficos da taxa de descarte de pacotes na classe EF, ou seja, a relação de pacotes descartados por pacotes transmitidos a cada segundo. É mostrada na Figura 5.14, a taxa de descarte do tráfego CBR sem o controlador fuzzy. As Figuras 5.15 e 5.16 foram geradas com o intuito de apresentar a taxa de descarte do tráfego CBR para as situações com o controlador fuzzy otimizado com Wang-Mendel e com o controlador otimizado com Wang-Mendel e AG. Mas como pode-se observar nas Figuras, para ambos os casos não houve descarte de pacotes para no tráfego apresentado.

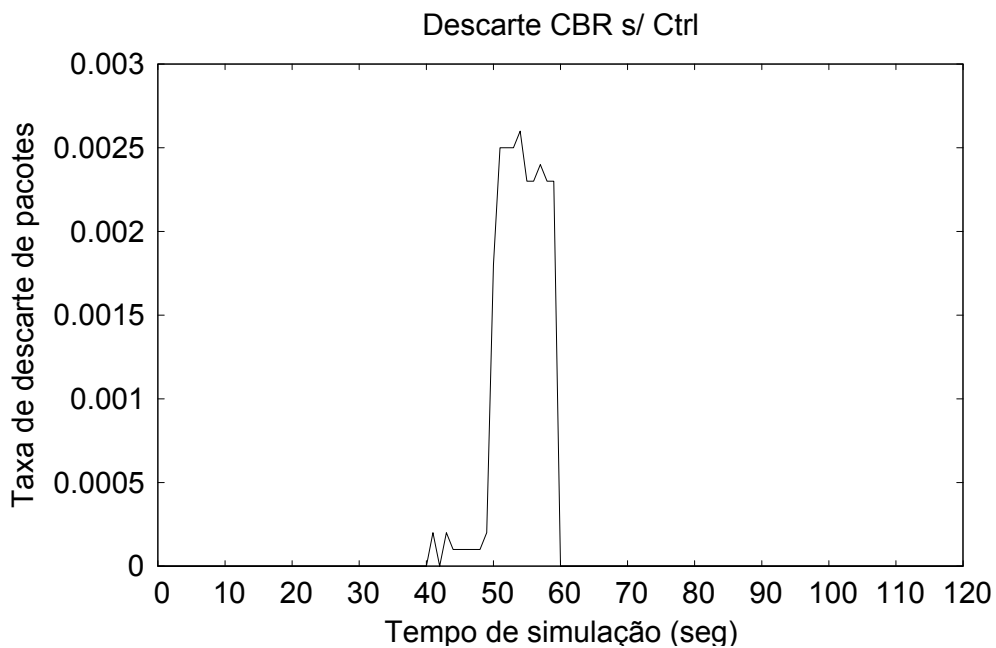


Figura 5.14: Taxa de descarte da classe EF s/ controlador

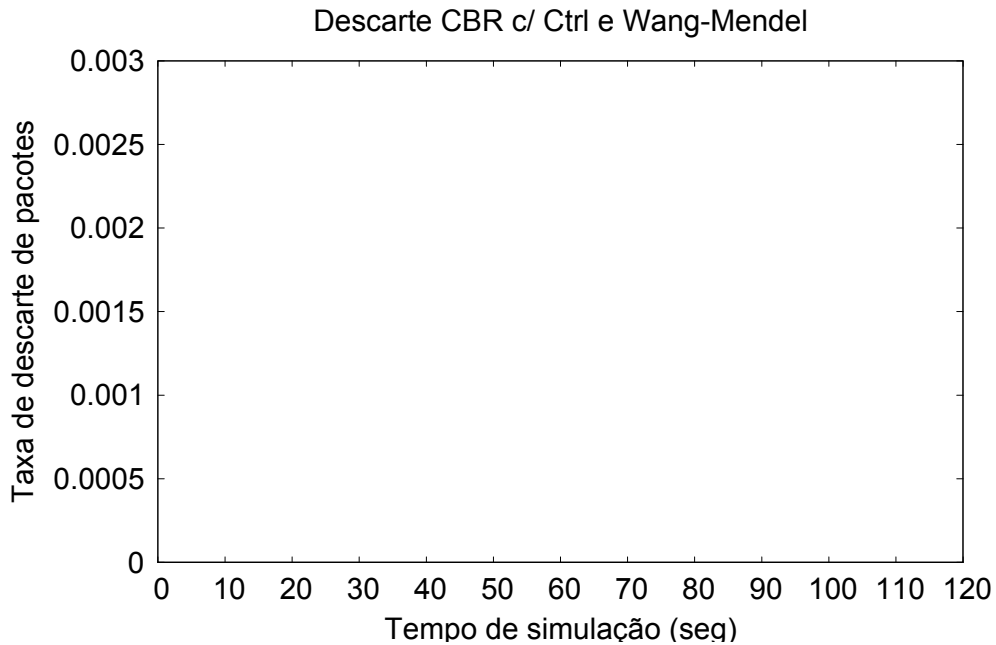


Figura 5.15: Taxa de descarte da classe EF c/ controlador e Wang-Mendel

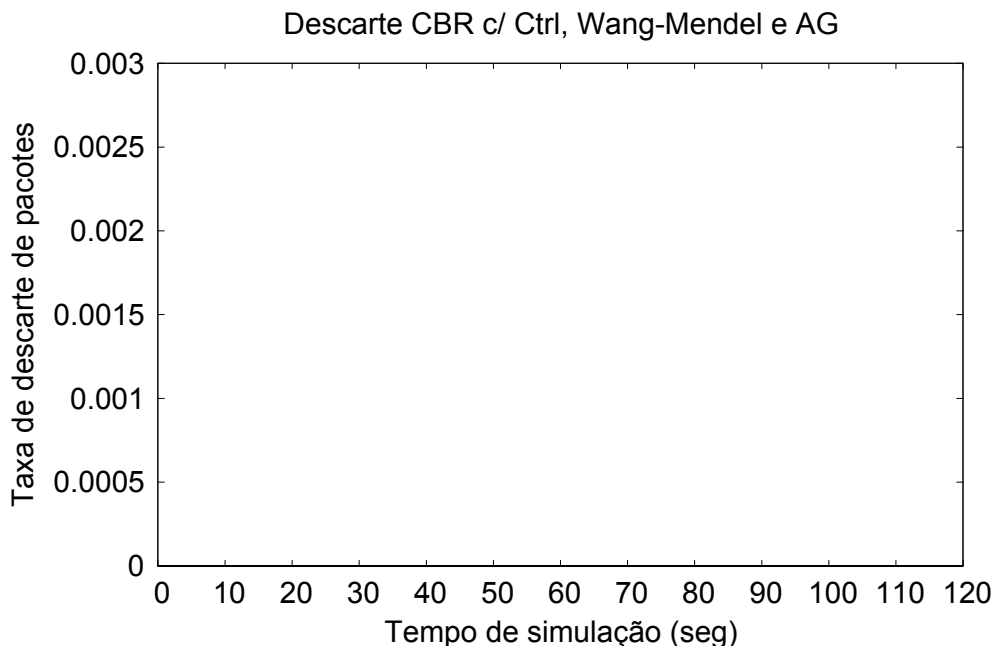


Figura 5.16: Taxa de descarte da classe EF c/ controlador, Wang-Mendel e AG

São mostrados nos gráficos a seguir, os resultados relativos à taxa de descarte de pacotes para o tráfego On-Off exponencial na classe EF. Os resultados obtidos sem controlador, utilizando controlador fuzzy e otimização por Wang-Mendel e os com a



rede reconfigurada, otimizada por Wang-Mendel e AG são apresentados respectivamente nos gráficos das figuras 5.17, 5.18 e 5.19.

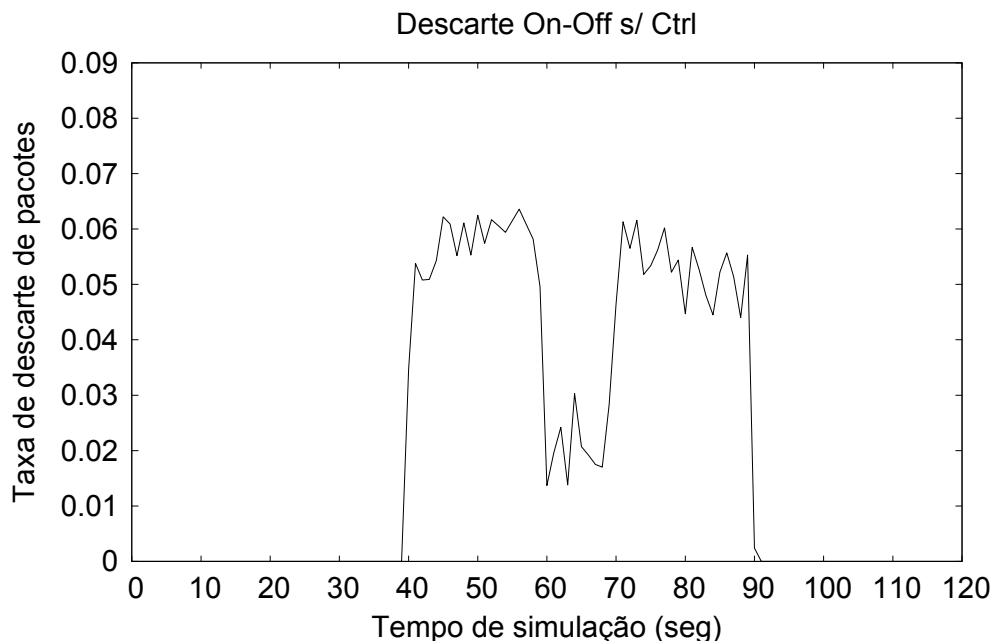


Figura 5.17: Taxa de descarte da classe EF s/ controlador

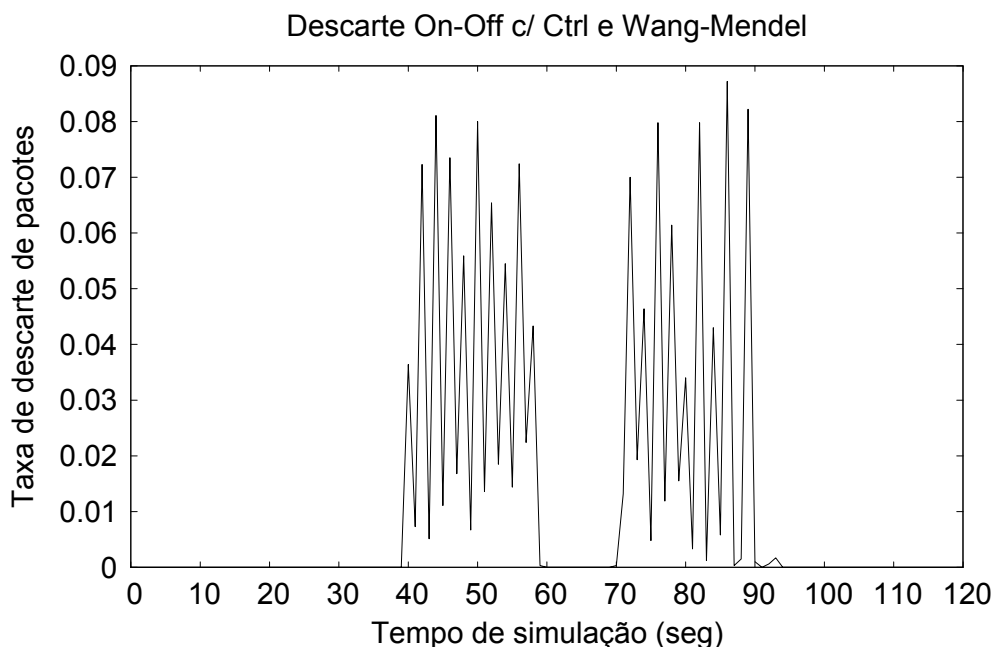


Figura 5.18: Taxa de descarte da classe EF c/ controlador e Wang-Mendel

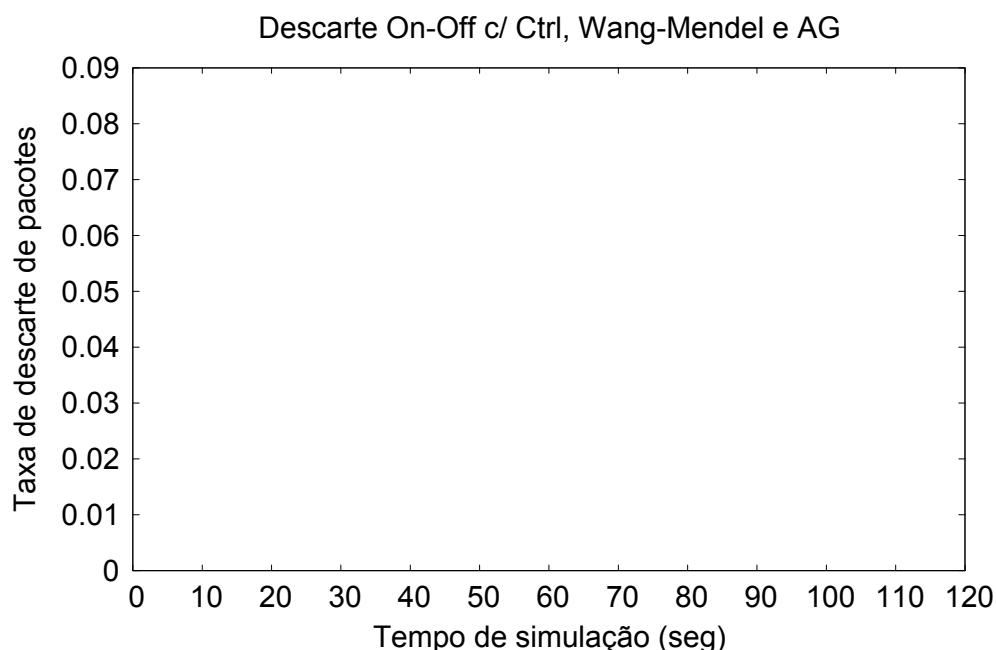


Figura 5.19: Taxa de descarte da classe EF c/ controlador, Wang-Mendel e AG

Constatou-se que nesse exemplo de tráfego On-Off, a verificação da base de regras pelo algoritmo de Wang-Mendel, apesar de diminuir a taxa de descarte como um todo para a classe EF, apresenta situações em que o resultado é pior do que na situação sem controlador. Já no caso da rede reconfigurada por AG a partir de dados históricos o descarte da classe EF foi a zero.

Pode-se verificar na tabela 5.4 que, apesar do excelente resultado para taxa de descarte, zero, para o tráfego CBR da classe EF com o controlador fuzzy otimizado com Wang-Mendel e AG, o resultado global para essa medida não foi bom, tendo em vista que o somatório do descarte (EF + BE) foi maior que no caso sem controlador. Isto ocorreu basicamente pela baixa demanda de tráfego de uma forma geral, o que acarretou num pequeno “*overhead*” de processamento do controlador. Já para o tráfego On-Off, que melhor representa uma situação de conversação, os resultados foram muito bons. Pois, além do descarte zero da classe EF na situação com controlador fuzzy e rede a otimizada com Wang-Mendel e AG, o descarte global (EF + BE) foi menor que nos casos sem controlador e com controlador otimizado com Wang-Mendel.

A tabela 5.4 apresenta o descarte de pacotes nas classes EF e BE. São mostrados os resultados de simulação para os tráfegos CBR e On-Off sem o controlador, com

controlador fuzzy otimizado por Wang-Mendel, e com controlador otimizado por Wang-Mendel e AG.

Tabela 5.4: Taxa de descarte de pacotes na topologia simples

Situações	EF (CBR)	BE (CBR)	EF+BE(CBR)	EF (On-Off)	BE (On-Off)	EF+BE(On-Off)
Sem Ctrl	0.0004	0.0489	0.0493	0.0385	0.0940	0.1325
Com Ctrl e WM	0.0000	0.0517	0.0517	0.0216	0.0979	0.1195
Com Ctrl, WM e AG	0.0000	0.0526	0.0526	0.0000	0.1051	0.1051

### 5.5.2 Resultados da topologia complexa

São apresentados nesta seção, os resultados das simulações na topologia complexa mostrada na seção 5.2.2 com o modelo de tráfego complexo mostrado na seção 5.3.2, referente ao tráfego de telefonia de um dia de semana comum de um Estado com grande demanda de recursos da rede, como o Estado de São Paulo. As medições realizadas foram: retardo e percentil de retardo fim-a-fim, variação do retardo (*jitter*) e taxa de descarte de pacotes da classe EF durante o período de simulação de 12 minutos, onde cada minuto simulado corresponde a 2 horas de um dia.

Como foi mostrado, nos resultados da topologia simples, o ganho obtido através da utilização do controlador fuzzy otimizado com Wang-Mendel e AG para um determinado perfil de tráfego se comparado com a utilização do controlador fuzzy apenas com a base de regras verificadas por Wang-Mendel. Na topologia complexa foram estabelecidos comparativos, para os tráfegos CBR e On-Off, apenas nas situações sem controlador e com controlador fuzzy a partir de uma rede reconfigurada por AG para suportar um padrão de tráfego pré-determinado e com sua base de regras verificadas por Wang-Mendel. Desta forma, este último estado será referenciado apenas com a expressão “com controlador”.

As simulações foram iniciadas com alocação de 50% da banda de saída para ambas as classes EF e BE. Para evitar medições com a rede sem tráfego, a seqüência de medidas foi iniciada somente após 30 segundos de tráfego.

### 5.5.2.1 Retardo fim-a-fim da classe EF

Os gráficos apresentados para verificar medidas de retardo fim-a-fim mostram apenas o tempo de espera nas filas, já que o tempo de transmissão dos links, por ser constante, foi subtraído. Os gráficos das Figuras 5.20 e 5.21 mostram respectivamente, o retardo no domínio DiffServ de um tráfego CBR da classe EF sem controlador e o retardo do tráfego CBR com controlador para a classe EF.

Os gráficos apresentados nas Figuras 5.22 e 5.23, mostram respectivamente, o retardo obtido para os pacotes da classe EF, sem a utilização de controlador e com controlador, ambos utilizados na situação de tráfego On-Off exponencial.

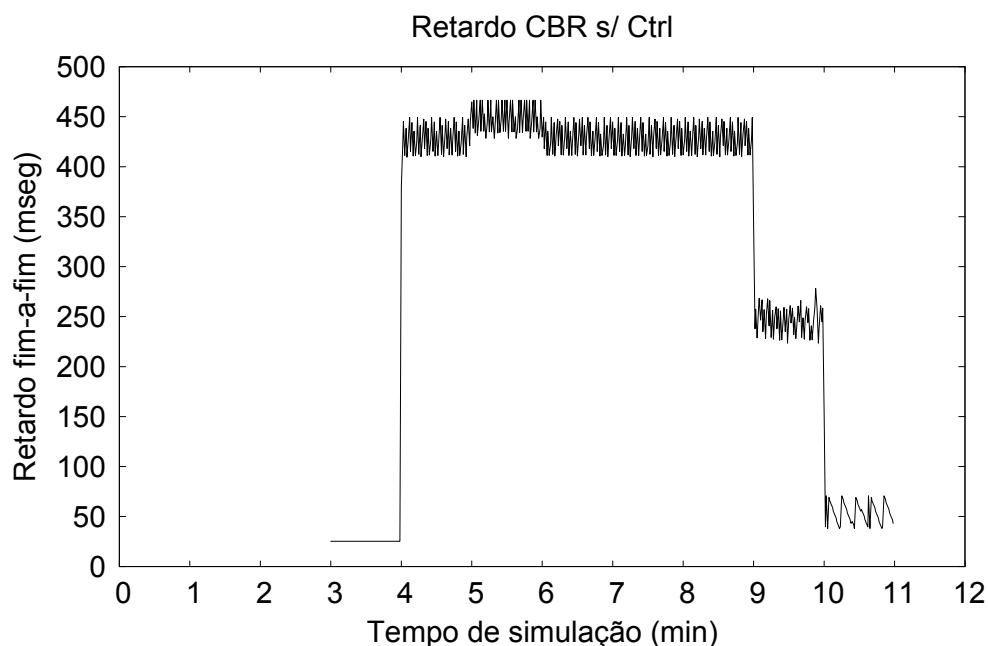


Figura 5.20: Retardo fim-a-fim da classe EF s/ controlador

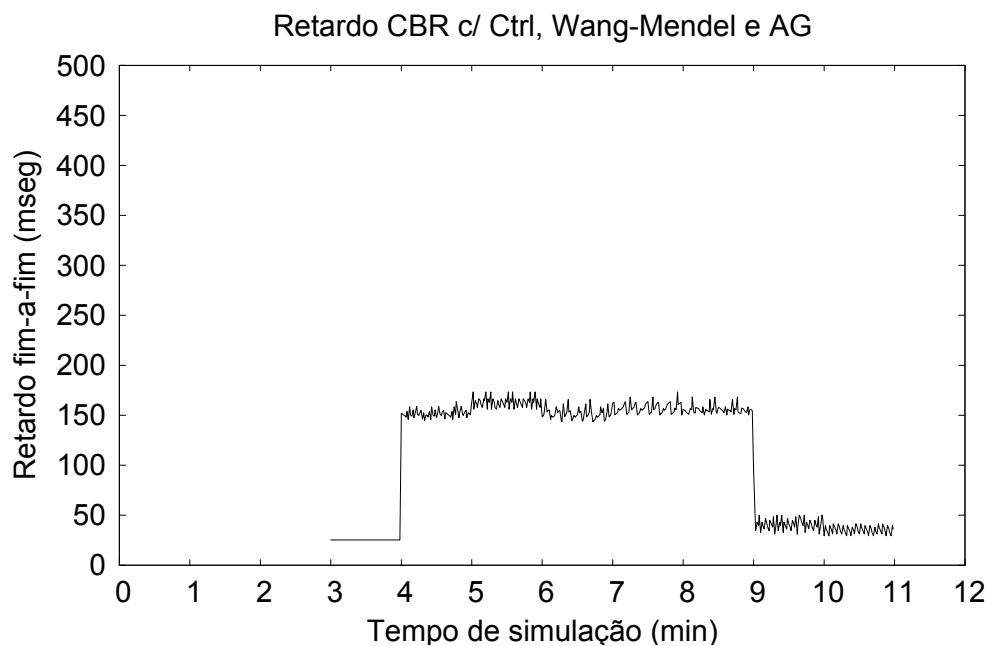


Figura 5.21: Retardo fim-a-fim da classe EF c/ controlador

Pode-se notar uma melhora significativa quanto ao retardo avaliado quando aplicamos o controlador fuzzy, principalmente no período de grande intensidade de tráfego, entre 4 e 9 minutos de simulação, que corresponde exatamente ao período de 8 às 18 horas (horário comercial). Observa-se também que, no período entre 0 a 3 e 11 a 12 minutos de simulação, que corresponde ao período entre 0 às 6 e 22 às 24 horas do dia, o tráfego CBR apresenta retardo zero, pois é um período de baixa intensidade de tráfego.

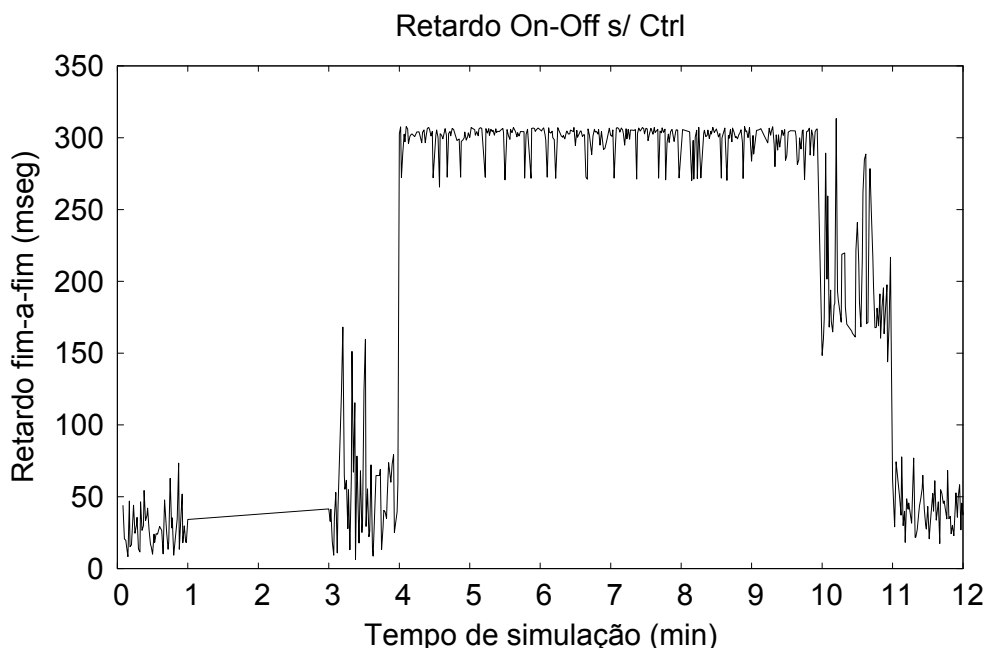


Figura 5.22: Retardo fim-a-fim da classe EF s/ controlador

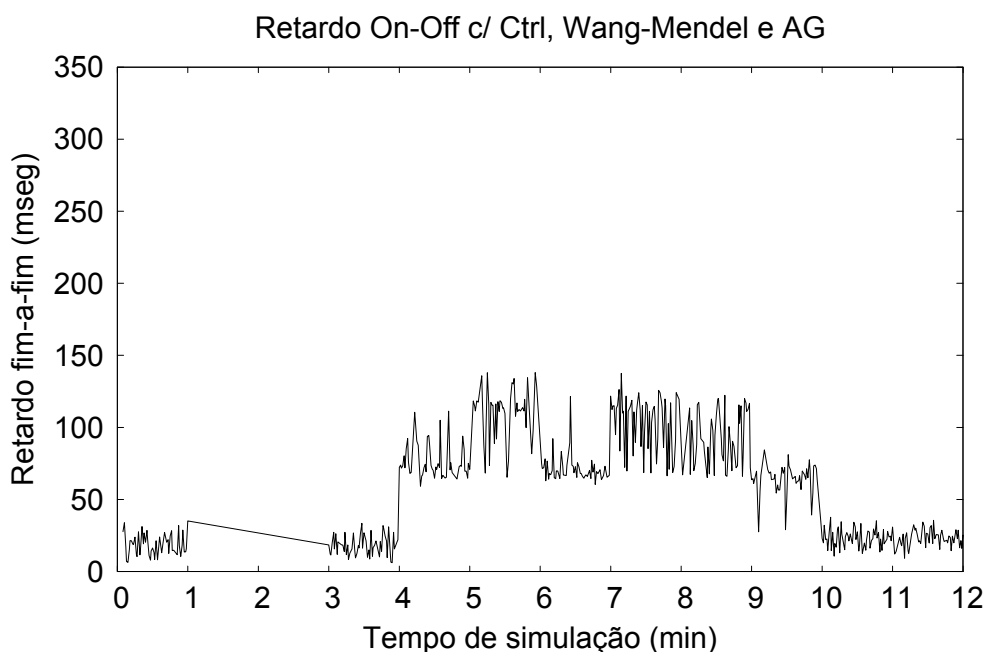


Figura 5.23: Retardo fim-a-fim da classe EF c/ controlador

Também pode-se observar uma melhora representativa no retardo da classe EF quando aplicado o controlador fuzzy para o tráfego On-Off, principalmente no período de grande intensidade de tráfego.

A tabela 5.5 mostra o retardo e a variação do retardo do tráfego CBR e On-Off para a topologia complexa.

Tabela 5.5: Retardo e variação de retardo da classe EF na topologia complexa (ms)

Tráfego	Média		Percentil 50		Percentil 90		Percentil 95	
	Retardo	Jitter	Retardo	Jitter	Retardo	Jitter	Retardo	Jitter
<b>CBR s/ Ctrl</b>	274,6	87,3	255,9	11,3	449,2	241,3	466,1	265,3
<b>CBR c/ Ctrl</b>	99,3	53,7	131,9	2,1	175,2	143,1	182,9	145,3
<b>OO s/ Ctrl</b>	209,3	17,9	254,7	1,5	312,6	42,7	316,1	104,7
<b>OO c/ Ctrl</b>	60,7	28,9	63,7	18,6	112,1	65	116,4	79,6

Analisando a tabela 5.5, pode-se observar a eficiência da utilização do controlador fuzzy em uma rede previamente configurada para atender demandas de QoS para um perfil de tráfego específico, oferecendo menor retardo que na situação de provisionamento estático (sem controlador) em todas as situações. Porém, quando analisamos a métrica variação de retardo (*jitter*), apesar da sua melhora em todas as situações para o tráfego CBR e no percentil 95 para o tráfego On-Off com o uso do controlador, esta apresentou piores resultados para a média, percentil 50 e percentil 90 quando foi usado o controlador fuzzy no tráfego On-Off. Isso ocorreu porque a ferramenta utilizada para otimização do controlador fuzzy, JFS [53], apresentou limitações para seu emprego na topologia complexa em questão. Em função disso, utilizamos a configuração do controlador fuzzy mais adequada para a topologia simples apresentada na seção 5.2.1.

### 5.5.2.2 Descarte da classe EF

Nessa seção, são apresentados os gráficos relativos à taxa de descarte de pacotes na classe EF. Os gráficos das Figuras 5.24 e 5.25 mostram respectivamente, a taxa de descarte da classe EF sem o controlador e com o controlador fuzzy para o tráfego CBR. São mostrados nas Figuras 5.26 e 5.27, os resultados da taxa de descarte da classe EF para o tráfego On-Off nas situações sem o controlador e com o controlador,

respectivamente.

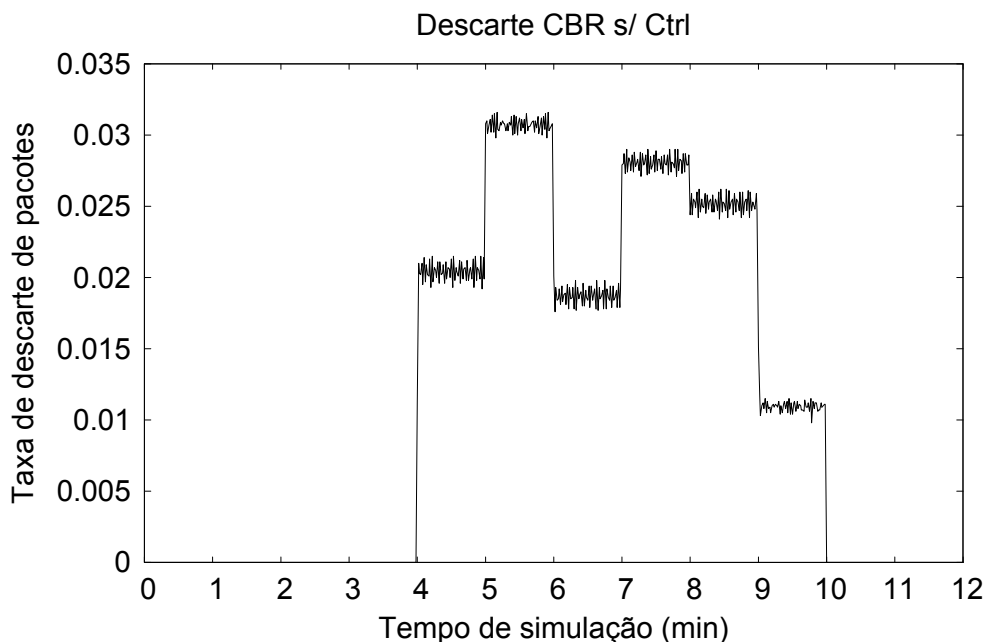


Figura 5.24: Taxa de descarte da classe EF s/ controlador

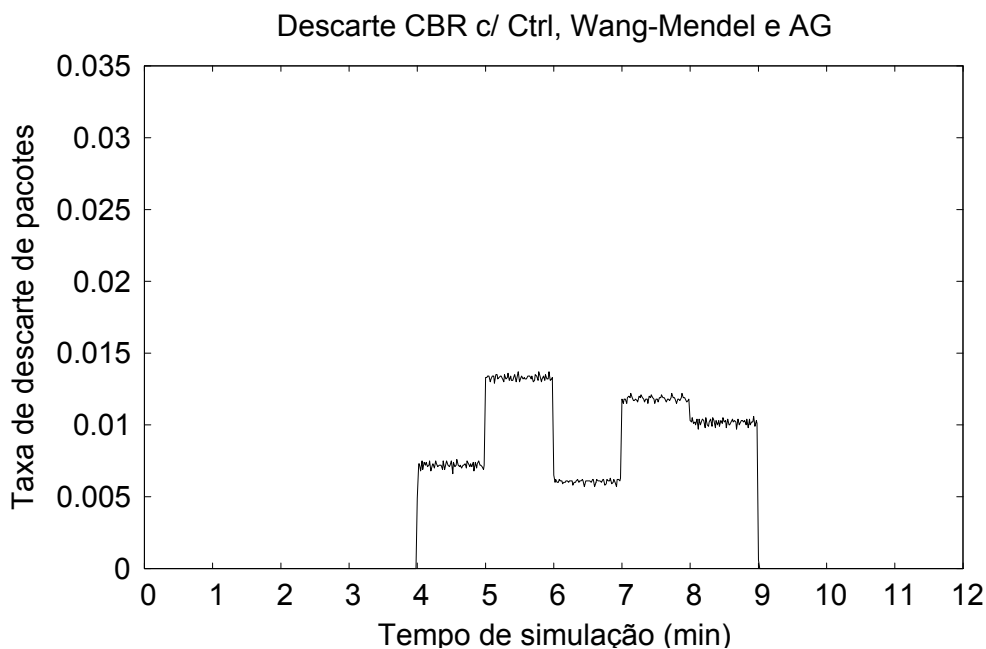


Figura 5.25: Taxa de descarte da classe EF c/ controlador, Wang-Mendel e AG

Observa-se uma melhora significativa na taxa de descarte de pacotes, quando é



usado o controlador fuzzy numa rede pré-configurada para suportar um padrão de tráfego CBR pré-determinado.

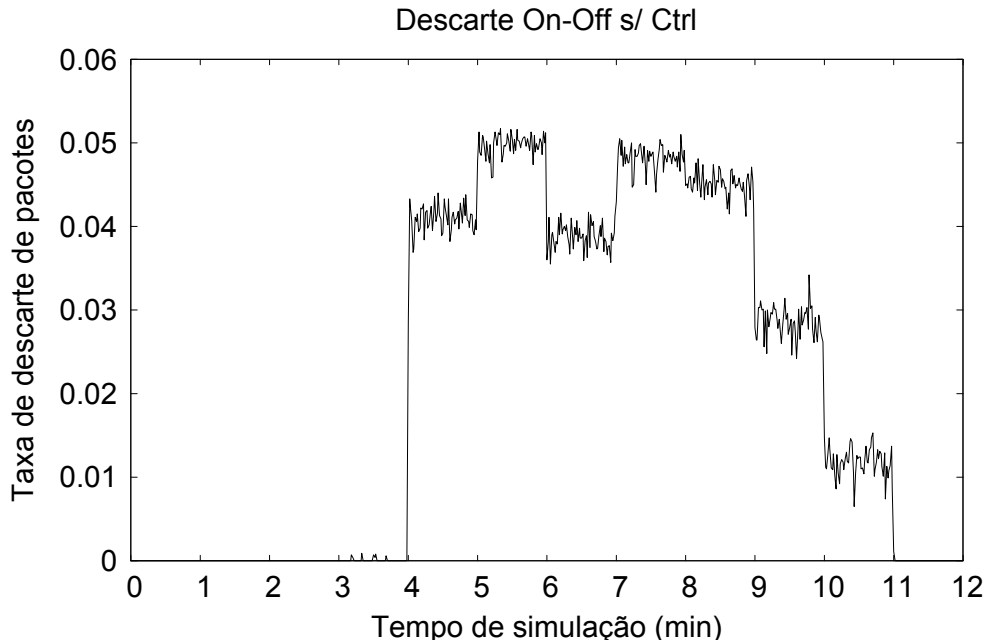


Figura 5.26: Taxa de descarte da classe EF s/ controlador

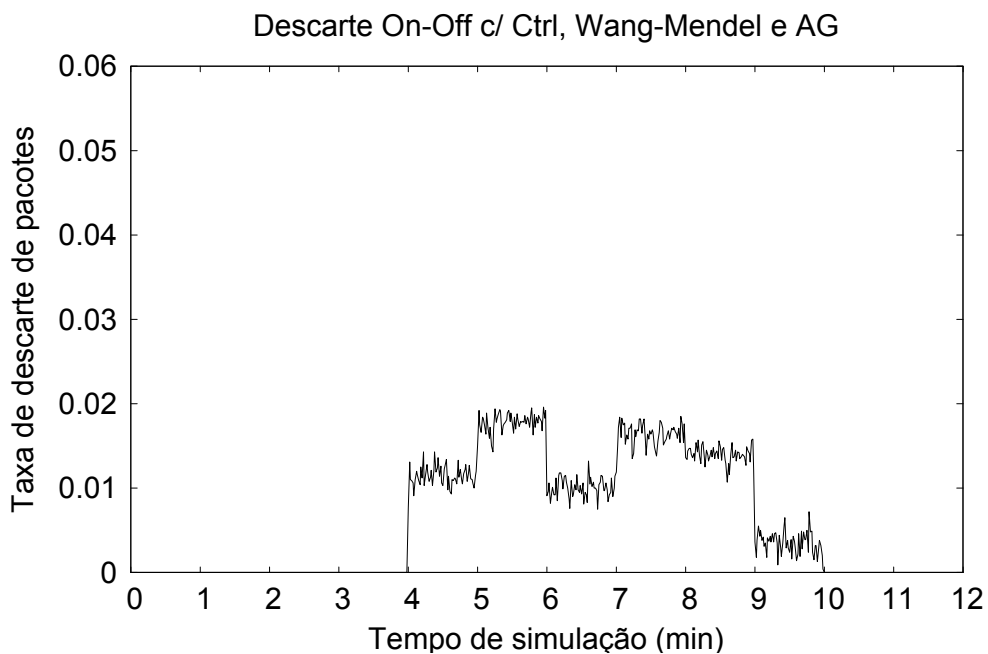


Figura 5.27: Taxa de descarte da classe EF c/ controlador, Wang-Mendel e AG

Da mesma forma que no caso de tráfego CBR, a utilização do controlador fuzzy

apresentou grande melhora no descarte de pacotes da classe EF para o tráfego On-Off, melhorando significativamente a QoS da rede.

A tabela 5.6 mostra a taxa de descarte de pacotes das classes EF e BE na topologia complexa. Pode-se notar uma redução na taxa de descarte da classe EF com o controlador fuzzy para ambos os tipos de tráfego se comparado com o caso sem controlador.

Tabela 5.6: Taxa de descarte de pacotes na topologia complexa

Situações	EF (CBR)	BE (CBR)	EF+BE(CBR)	EF (On-Off)	BE (On-Off)	EF+BE(On-Off)
Sem Ctrl	0.0200	0.0280	0.0480	0.0367	0.0439	0.0806
Com Ctrl, WM e AG	0.0078	0.0400	0.0478	0.0114	0.0562	0.0676

Como o tráfego entrante no domínio DiffServ extrapola a capacidade da rede, a redução na taxa de descarte da classe EF provoca um aumento na taxa da classe BE. Entretanto, notamos que o uso do controlador fuzzy na rede previamente otimizada com AG para suportar um certo padrão de tráfego diminui a taxa de descarte agregada de forma considerável para o tráfego On-Off, ou seja, a soma das taxas de descarte das classes EF e BE é menor que no caso sem controlador, obtendo-se assim um melhor desempenho global. Já para o tráfego CBR, os resultados de taxa de descarte agregada para os casos sem controlador e com controlador não apresentaram grandes diferenças.

## 5.6 Comentários

Neste capítulo foram apresentados o ambiente de simulação, uma síntese das ferramentas utilizadas, as topologias de rede, os modelos de tráfego definidos a partir de consultas ao DM de gerenciamento de redes desenvolvido e as medidas avaliadas para verificar a viabilidade de metodologia proposta.

Foram mostrados os resultados da simulação em uma topologia simples, que nesse trabalho está representando a rede do Estado de Pernambuco e em uma topologia complexa, que está tipificando a rede do Estado de São Paulo. Foram apresentados os

resultados de retardo fim-a-fim, variação do retardo e taxa de descarte de pacotes, além dos valores de percentil de retardo fim-a-fim e variação de retardo para as simulações que representam o tráfego na rede durante o período de todo um dia, isto é, 24 horas.

Em cada situação, foi demonstrada a eficiência da utilização do controlador fuzzy em uma rede previamente otimizada por AG a partir de padrões de tráfegos anteriores, consultados através do DM de gerenciamento de redes para estabelecer perfis específicos de tráfegos. Assim, foi possível verificar a funcionalidade do protótipo e então validar a metodologia apresentada no capítulo 4.

## Capítulo 6

### Conclusões e Trabalhos Futuros

Neste trabalho, foi apresentada uma metodologia de utilização de um mecanismo de provisionamento dinâmico de recursos de redes baseado em lógica fuzzy, seguido de um modelo de utilização de DW para prestar suporte a decisão quanto a um modelo de tráfego esperado, afim de que a rede seja previamente acomodada de modo a garantir QoS para um padrão de tráfego específico.

A utilização do DM de gerenciamento de rede possibilitou o estudo do comportamento do tráfego em determinados dias e horas em regiões distintas do território nacional. O uso do algoritmo de Wang-Mendel possibilitou a otimização da base de regras estabelecida pelo projetista, eliminando possíveis conflitos.

A aplicação do AG, tomando como base de conhecimentos medições feitas a partir de dados históricos de tráfego, propiciou a otimização dos parâmetros do controlador fuzzy e a obtenção de bons resultados, sem depender dos critérios de escolha de QoS estabelecidos pelo projetista.

O uso da lógica fuzzy no controlador previamente otimizado para suportar padrões específicos de tráfego, possibilitou um tratamento eficaz das imprecisões e incertezas do tráfego de entrada no domínio DiffServ, acarretando em bons resultados nos parâmetros de QoS avaliados em simulação.

Deve-se ressaltar que o uso do AG exige muito recurso computacional, mas ele não influencia os equipamentos de rede, pois a otimização é realizada no equipamento

gerenciador. Além disso, seu funcionamento é esporádico, na ordem de dias, ou quando ocorrer mudanças na topologia da rede ou na ocorrência de previsões de mudanças de perfil de tráfego. Enfim, seu emprego é bastante útil para situações de mudanças programadas.

Um grande benefício desta metodologia está no gerenciamento baseado em políticas, este permite que o controlador seja genérico, podendo definir o comportamento do sistema de acordo com as decisões administrativas do gerenciador do domínio. Além disso, o comportamento do controlador pode ser alterado em todo o domínio durante o funcionamento normal do sistema.

A utilização das classes EF e BE nas simulações com o uso do controlador fuzzy para validação do modelo proposto, proporcionou respostas sensivelmente melhores nas medidas de QoS para a classe EF, se comparada àquelas obtidas na situação sem controlador. A melhoria nas medidas de QoS foi aparente tanto na topologia simples, que possui baixa demanda de tráfego, como na topologia complexa que apresenta alta demanda e variação de tráfego.

As simulações realizadas para validar o modelo utilizaram dois tipos de tráfego: CBR e On-Off exponencial. Comparando os resultados obtidos de tráfego CBR e On-Off, observa-se que a metodologia aplicada apresentou-se mais eficiente para tráfegos On-Off, ou seja, tráfegos encontrados numa rede real, exceto nas medidas de variação de retardo na topologia complexa. Neste caso, a ferramenta JFS usada na otimização do controlador fuzzy por AG apresentou limitações, não possibilitando a otimização da rede nesta situação. A busca de uma ferramenta capaz de otimizar o controlador fuzzy para esta situação específica foi deixada para trabalhos futuros.

Por fim, conclui-se que a aplicação de mecanismos de provisionamento dinâmico numa rede otimizada para atender determinados padrões de tráfego, apresenta vantagens na provisão de QoS. A atitude usual de superprovisionamento possibilita a manutenção da QoS, porém o custo para manter essa infra-estrutura é consideravelmente alto. Foi mostrado que a lógica fuzzy se apresentou adequada para tratar tráfegos com alta variação, comuns em situações reais. A metodologia proposta permitiu a construção de um sistema escalável, eficiente e que admite atitudes pró-ativas quanto ao planejamento de redes, afim de adequá-las a novos perfis de tráfego.

Uma proposta de trabalho futuro seria a construção de um DM de gerenciamento de redes baseado em telefonia IP, de modo que parâmetros da rede como, tamanhos médios de pacote e quantidade de pacotes pertencentes a cada classe possam ser extraídos do DW sem precisar da utilização de fator de ajuste. Assim, seria possível fazer simulações baseadas em dados reais de telefonia.

Outra proposta de trabalho deve ser a inclusão de fatores climáticos no DW e, conseqüentemente, no DM de gerenciamento de redes, tendo em vista que estes influenciam significativamente no perfil do tráfego das redes. Deste modo, o modelo de suporte a decisão será enriquecido, ajudando a melhor dimensionar as tendências de necessidade de recursos da rede.

# Apêndice A – Tabelas do Modelo Estrela

O modelo ou esquema estrela, é constituído basicamente por dois tipos de tabelas que se relacionam formando uma estrutura simples. As tabelas que compõem este modelo são: tabela de fatos, que fica no centro do modelo; e as tabelas de dimensão, que ficam distribuídas na extremidade do modelo. A tabela de fatos é ligada às demais tabelas por várias junções, uma junção para cada tabela de dimensão. São apresentadas nas seções A.1 e A.2, as tabelas de fato e de dimensão respectivamente. Na seção A.3, é mostrada a hierarquia de dimensões.

## A.1 Tabela de Fato

Chama-se de fato tudo o que pode ser representado por valores numéricos [40, 42]. Denomina-se este conjunto de valores numéricos por métricas ou medidas. Os fatos são utilizados para avaliar a evolução de características que estão sendo medidas. Deste modo, as tabelas de fato representam as informações que estão sendo avaliadas, sendo normalmente constituídas de valores numéricos que representam objetos de análise. As tabelas de fato contém as informações básicas do nível de transação do negócio de interesse particular de uma transação, isto faz com que normalmente sejam tabelas grandes com muitos registros. Kimball [19] descreve que os projetistas devem priorizar os valores aditivos na construção das tabelas de fato.

Os fatos podem ser caracterizados por transações individuais [19], “snapshots” e linhas de itens. Onde:

1. Transações individuais: Apresentam geralmente uma estrutura simples, com um campo acumulado que contém o valor da transação;
2. “Snapshots”: Estes fatos representam medidas de atividades extraídas em tempo determinado, como por exemplo: fim do dia, fim da semana;
3. Linhas de itens: Representam exatamente uma linha de item, como por exemplo: itens de pedido ou itens de entrega.

Segundo Machado [40], os valores que caracterizam a avaliação dos fatos podem ser classificados em dois tipos: Valores aditivos e valores não aditivos.

1. Valores aditivos: São valores da tabela de fatos que permitem a aplicação das operações de soma, subtração e média. Valores como: total de vendas e media dos itens pedidos são exemplos de valores aditivos;
2. Valores não aditivos: São valores da tabela de fatos que não podem ser manipulados livremente, como percentuais ou relativos. Para esses valores, os cálculos devem ser feitos em cima de dados absolutos nos quais se baseiam. Todos os valores que medem um nível de intensidade são valores estáticos, não aditivos.

## **A.2 Tabela de Dimensão**

Por conceito, dimensões são os elementos que participam de um fato, assunto de negócios [40]. As tabelas de dimensão armazenam os dados necessários para análise ao longo das dimensões, normalmente são menores que a tabela de fatos, visto que em geral não apresentam atributos numéricos, isto é, seus atributos geralmente possuem a função de descrever e classificar os elementos que participaram de um fato.

As tabelas de dimensão apresentam chave simples e seus campos, normalmente descritivos, são empregados como fontes de restrições e linhas de cabeçalhos para



relatórios. A qualidade do banco de dados é proporcionalmente dependente dos atributos de dimensões, desta forma, deve haver muita atenção ao se construir esta tabela.

Um membro de dimensão é utilizado para determinar a posição de um item de dado. Por exemplo, a tabela de dimensão tempo é formada por todas as ocorrências de horas, dias, meses, trimestres e anos. Da mesma forma, a tabela região é formada por todas as ocorrências de cidade, município, estado e país.

## A.2.1 Hierarquia de Dimensões

Como foi visto na seção A.2, uma dimensão é formada por membros ou atributos [43]. Esses definem “agrupamentos” ou níveis de informações segundo os quais um fato pode ser analisado. As hierarquias são a base para a agregação dos dados e para a navegação, através de “drill down” e “drill up” descritos na seção 2.5.2, entre os diferentes níveis de detalhe em um sistema multidimensional [43, 44].

Algumas estruturas hierárquicas são identificadas com facilidade na construção de um modelo multidimensional, como exemplo, tem-se a dimensão tempo representada por horas, dias, semanas, meses, trimestres e anos. A Figura 4.5 mostra um exemplo de diagrama da hierarquia de tempo.

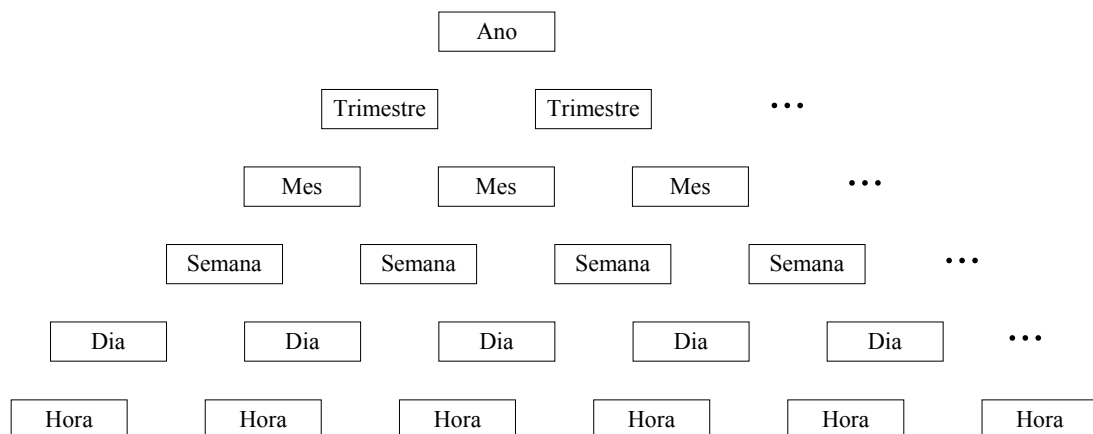


Figura A.1: Hierarquia de Tempo.

Uma importante questão, diz respeito à influência da hierarquia das dimensões sobre as tabelas de fato, tendo em vista que a tabela de fatos deve refletir a menor granularidade das dimensões. Deste modo, pode-se garantir que não estão sendo armazenados registros na tabela de fato que representem totais referentes a um nível mais alto na hierarquia de uma tabela de dimensão.

A definição da granularidade dos dados no DW possui um papel muito importante, por conceito, granularidade de dados refere-se ao nível de sumarização dos elementos de dados e ao nível de detalhe disponível nos dados [40]. Um dos aspectos essenciais na modelagem de um DW está em definir o nível da granularidade dos dados. Não existem restrições quanto ao armazenamento de dados primitivos no DW, embora este tipo de armazenamento não seja muito comum. A granularidade está relacionada ao volume de dados armazenados que estão disponíveis para consultas. Desta forma, aumentar o nível de granularidade implica em restringir o nível das consultas. Entretanto, ter um baixo nível de granularidade significa o atendimento a um maior número de consultas, mas aumenta a necessidade de espaço de armazenamento.

## Apêndice B – Metadados

Os metadados são definidos como dados sobre dados, consiste no detalhamento das informações. Tendo em vista a complexidade de informações de um DW, a documentação dos sistemas e dos bancos de dados é de vital importância num ADW. Os metadados devem definir um conjunto de elementos que possuem uma semântica padronizada, que possibilita representar as informações eletrônicas e a descrição de recursos eletrônicos de maneira bibliográfica. Pode ser também entendido como o meio de descobrir quais os recursos existentes, e como podem ser obtidos e acessados, evitando a ambigüidade dos dados [47].

Os metadados possuem a função de facilitar a localização e a recuperação das informações de uma empresa, são importantes para a identificação, organização e recuperação da informação digital. Neste sentido, são utilizados procedimentos técnicos de indexação e classificação dos conteúdos informacionais, possibilitando a integração de fontes diversificadas e heterogêneas de informação.

Num projeto de DW, deve-se gerar documentações sobre: os levantamentos de dados, o banco de dados, os relatórios a serem gerados, as origens dos dados que alimentaram o DW, os processos de extração, os tratamentos e rotinas de carga dos dados, além de, possivelmente, as regras de negócio da empresa e todas as suas mudanças.

Em Silva [48], o autor destaca a importância da utilização do metadado num ADW, segundo o autor, reunir e manter atualizada as informações de uma empresa e converter em vantagem competitiva é um dos grandes desafios dos tempos modernos. Embora seja de suma importância que os metadados estejam disponíveis para todos os

usuários, Silva [48] faz distinção do metadado entre dois tipos de usuários. Desta forma temos: o Metadado Técnico, que deve ser utilizado basicamente pelo corpo técnico da equipe de DW; e o Metadado de Negócio, que consiste nos metadados que serão gerados para atender os usuários finais. Normalmente, as informações providas pelos metadados de negócio são disponibilizadas para os usuários finais através de bases departamentais (DMs), que prestam um detalhe específico do assunto abordado.

## Apêndice C – Procedimento de ETL

O procedimento de ETL (*Extract, Transform and Load*) é responsável pelo tratamento que os dados advindos dos Sistemas Operativos sofrerão antes de chegar ao DW. Esta etapa é uma das mais críticas de um projeto de DW, pois uma informação carregada erroneamente trará sérias conseqüências nas fases posteriores. Assim, seu objetivo consiste em fazer a integração de informações de fontes múltiplas e complexas, e levá-las ao DW ou DM.

Esta fase é dividida em três passos: extração, transformação e carga dos dados.

Em Silva [48], o autor relata alguns dos tratamentos que o processo de ETL deve, necessariamente, realizar nos dados, ou seja, descreve as principais metas que devem ser atingidos pelo procedimento de ETL, que são:

- Limpar os dados;
- Transformar dados, aplicando regras de negócio;
- Combinar diferentes tipos de dados;
- Validar domínios;
- Gerar agregações;
- Realizar as cargas dos modelos, etc.

É importante ressaltar que o processo de ETL deve ser feito de forma muito cuidadosa, pois as transformações realizadas nesta fase são inclusões de critérios que permitem visualizar um determinado dado sobre uma ótica específica, o que não cria informações que só existem no DW. Desta forma, o objetivo do DW de refletir apenas as informações existentes nas bases operacionais se mantém.

## Referências Bibliográficas

- [1] BLAKE, S., BLACK, D., AND CARLSON, M. "An architecture for differentiated services". RFC 2475, dezembro de 1998.
- [2] BLIGHT, D., AND HAMADA, T. "Policy-based networking architecture for QoS interworking in IP Management". In: *Integrated Network Management, Distributed Management for the Networked Millenium 1999*, pp.811-826, 1999.
- [3] SMITH, A. M. "Data Warehousing: A Short Overview" for ITtoolbox Data Warehouse, Janeiro de 2002.
- [4] LEWIS, J., WILLIAM, *Data Warehousing and Ecommerce - Review by Anne Marie Smith*, LaSalle University, Prentice-Hall, Inc., 2001.
- [5] FERNANDEZ, M. P., *Provisionamento de Recursos em Arquitetura DiffServ para Melhoria de Qualidade de Serviço (QoS)*, Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, outubro de 2002.
- [6] LEE, C. C. "Fuzzy Logic in control systems: Fuzzy logic controller, Par II", *IEEE Transactions on Systems, Man, and Cybernetics v. 20*, n. 2, pp. 419-435, março de 1990.
- [7] LIAO, R. R., AND CAMPBELL, A. T. "Dynamic edge provisioning for core networks". In: *IFIP/IEEE Eighth International Workshop on Quality of Service (IWQoS 2000)*, Pittsburgh, USA, junho de 2000.
- [8] LIAO, R. R., AND CAMPBELL, A. T. "Dynamic core provisioning for quantitative differentiated service". In: *IFIP/IEEE Ninth International Workshop on Quality of Service (IWQoS 2001)*, Karlsruhe, Germany, junho de 2001.

- [9] FERREIRA, A. C. F., *Um Modelo para Suporte à Integração de Análises Multidimensionais e Espaciais*, Tese de M.Sc., Instituto de Matemática – NCE/UFRJ, Rio de Janeiro, RJ, Brasil, abril de 2002.
- [10] MOURA, F. A., *Suporte a Decisão em Interpretação de Fenômenos*, Tese de M.Sc., Instituto de Matemática – NCE/UFRJ, Rio de Janeiro, RJ, Brasil, março de 2001.
- [11] FERNANDEZ, M. P., DE CASTRO P. PEDROZA, A., AND DE REZENDE, J. F. “Qualidade de serviço em um domínio diffserv através de gerenciamento baseado em políticas”. In: *XIX Simpósio Brasileiro de Redes de Computadores (SBRC 2001)*, Florianópolis, Brasil, maio de 2001.
- [12] FERNANDEZ, M. P., DE CASTRO P. PEDROZA, A., AND DE REZENDE, J. F. “Quality of service in a diffserv domain using policy-based management”. In: *Teletraffic Engineering in the Internet Era*, 1 ed., v. 4. Elsevier Science B. V., Amsterdam, The Netherlands, ch. QoS IP – Diff Service, ISBN 0-444-50911-9, pp. 919-930, setembro de 2001.
- [13] FERNANDEZ, M. P., DE CASTRO P. PEDROZA, A., AND DE REZENDE, J. F. “Policy-based management providing qos in a diffserv domain”. In: *Latin American Network Operation and Management Symposium (LANOMS'2001)*, Belo Horizonte, Brazil, agosto de 2001.
- [14] FERNANDEZ, M. P., DE CASTRO P. PEDROZA, A., AND DE REZENDE, J. F. “QoS provisioning across a diffserv domain using policy-based management”. In: *(Globecom 2001)*, San Antonio, USA, novembro de 2001.
- [15] FERNANDEZ, M. P., DE CASTRO P. PEDROZA, A., AND DE REZENDE, J. F. “Otimização de controlador fuzzy para provisionamento de recursos em ambiente diffserv através de algoritmo genético”. In: *XX Simpósio Brasileiro de Redes de Computadores (SBRC'2002)*, Búzios, Brasil, maio de 2002.
- [16] FERNANDEZ, M. P., DE CASTRO P. PEDROZA, A., AND DE REZENDE, J. F. “Optimizing fuzzy controllers with genetic algorithms for qos improvement”. In: *International Telecommunication Symposium (ITS'2002)*, Natal, Brazil, setembro de 2002.



- [17] FERNANDEZ, M. P., DE CASTRO P. PEDROZA, A., AND DE REZENDE, J. F. “Dynamic qos provisioning in diffserv domains using fuzzy logic controllers”, *Telecommunication Systems Journal*, julho de 2002.
- [18] KIMBALL, R. “Bringing Up Supermarts – A Step-by-Step Approach to Building a Data Warehouse from Granular Data”, *DBMS*, v. 11, n. 1, Janeiro de 1998.
- [19] KIMBALL, R., REEVES, L., ROSS, M. AND THORNTHWAITE, W., *The Data Warehouse Lifecycle Toolkit – Expert Methods for Designing, Developing and Deploying Data Warehouses*, John Willey & Sons, Inc., New York, 1998.
- [20] INMOM, W. H., *Como construir o Data Warehouse*, Editora Campus, Rio de Janeiro, 1997.
- [21] WU, M. AND BUCHMANN, A. P. “Research Issues in Data Warehousing”, DVS1, Fachbereich Informatik, Technische Hochschule Darmstadt, 1997.
- [22] POE, V., KLAUER, P., BROBST, S., *Building a Data Warehousing for Decision Support*, New Jersey, Prentice-Hall, Inc, 1998.
- [23] INMOM, W. H. AND HACKATHORN, R. D., *Como Usar o Data Warehouse*, Editora Infobook, Rio de Janeiro, 1997.
- [24] GOLFARELLI, M., MAIO, D. AND RIZZI, S. “Conceptual Design of Data Warehouses from E/R Schemes”, *Proceedings of the Hawaii International Conference on System Sciences*, Januar, Kona, Hawaii, 1998.
- [25] COREY, M. J. AND ABBEY, M., *Oracle Data Warehousing*, Oracle Press Edition, Califórnia, 1996.
- [26] NICHOLS, K., AND CARPENTER, B. “Definition of differentiated services per domain behavior and rules for their specification”. RFC 3086, abril de 2001.
- [27] BLACK, D., BRIM, S., CARPENTER, B., AND FAUCHEUR, F. L. “Per hop behavior identification codes”. RFC 3140, junho de 2001.
- [28] BERNET, Y., BLAKE, S., GROSSMAN, D., AND SMITH, A. “An Informal Management Model for Diffserv Routers”. RFC 3290, maio de 2002.
- [29] SLOMAN, M. “Policy Driven Management for Distributed Systems”, *Journal of*

*Network and Systems Management*, v. 2, n. 4, Plenum Press, pp. 333-360, 1994.

[30] FERNANDEZ, M. P., DE CASTRO P. PEDROZA, A., AND DE REZENDE, J. F. “Implementação de Políticas de Gerenciamento através de Lógica Fuzzy visando melhoria da Qualidade de Serviço (QoS)”. In: *XXI Simpósio Brasileiro de Redes de Computadores (SBRC'2003)*, Natal, Brasil, maio de 2003.

[31] LORENZ, D. AND ORDA, A. “QoS Routing in Networks with Uncertain Parameters”, In: *IEEE Infocom 98*, 1998.

[32] VASILAKOS, A. AND ANAGNOSTAKIS, K. “Evolutionary-Fuzzy Prediction for Strategic Inter-Domain Routing: Architecture and Mechanisms”, In: *WCCI 1998*, Anchorage, USA, May 1998.

[33] AGUIAR, H. AND JUNIOR, O., *Lógica Difusa – Aspectos práticos e aplicações*, Ed. Interciência, Rio de Janeiro, 1999.

[34] ZADEH, L. A., *Information and Control*, v. 8, pp. 338 – 353, 1965.

[35] COX, E., O'HAGAN, M. AND ZADEH, L. *The fuzzy system handbook*. Academic Press Inc, outubro de 1998.

[36] CORDÓN, O., HERRERA, F. AND PEREGRIN, A. “A practical study on the implementation of fuzzy logic controllers”, *The International Journal of Intelligent Control and Systems* v. 3, n. 3, pp. 49-91, junho de 1999.

[37] LI, B. AND NAHRSTEDT, K. “Dynamic reconfiguration for complex multimedia application”. In: *IEEE International Conference on Multimedia Computing and Systems 99*, v. 1, pp. 165-170, julho de 1999.

[38] HOLLAND, J. H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[39] BATINI, C., CERI, S. AND NAVATHE, S. B., *Conceptual Database Design: An Entity-Relationship Approach*, The Benjamin/Cummings Publishing Company, Califórnia, Inc. 1992.

[40] MACHADO, F. N. R., *Projeto de DataWarehouse – Uma Visão Multidimensional*, Ed. Érica, São Paulo, 2000.

- [41] KIMBALL, R. “A Dimensional Modeling Manifesto – Drawing the Line Between Dimensional Modeling and E-R Modeling Techniques”, *DBMS Magazine*, 1997.
- [42] KIMBALL, R. AND MERZ, R., *The Data Webhouse Toolkit – Building the Web-Enabled Data Webhouse*, John Wiley & Sons, Inc. 2000.
- [43] THOMSEN, ERIK., *Olap Solutions – Building Multidimensional Information Systems*, New York: John Wiley & Sons, Inc, 1997.
- [44] MEYER, D. AND CANNON, C., *Building a Better Data Warehouse*, New Jersey, Prentice Hall PTR, 1998.
- [45] INMOM, W. H. “Data Mart Does not Equal Data Warehouse”. Disponível na Internet, site: [http://www.dmreview.com/issues/1998/may/articles/may98\\_38.htm](http://www.dmreview.com/issues/1998/may/articles/may98_38.htm).
- [46] DYCHÉ, J. “Scoping Your Data Mart Implementation”. Disponível na Internet, site: <http://www.dbmsmag.com/9808d13.html>.
- [47] MODESTO, F. “Metadados: Uma abordagem”. Representação Descritiva II – CBD/ECA/USP, Abril de 2003.
- [48] MONTEIRO SILVA, M., *Aspectos Críticos na Implementação de um DW*, Escola Politécnica - UFRJ, Gerencia Estratégica da Informação, Rio de Janeiro, RJ, Brasil, 2002.
- [49] DAMIANOU, N., DULAY, N., LUPU, E. AND SLOMAN, M. “Ponder Policy Specification Language”. <http://www-dse.doc.ic.ac.uk/policies/ponder.shtml>.
- [50] DAMIANOU, N., DULAY, N., LUPU, E. AND SLOMAN, M. “The Ponder policy specification language”, In: *Policy 2001: Workshop on Policies for Distributed Systems and Networks*, Bristol, UK, pp. 18-39, Janeiro de 2001.
- [51] LUPU, E. AND SLOMAN, M. “Conflicts in policy-based distributed systems management”, *IEEE Transactions on Software Engineering, Special Issue on Inconsistency Management*, v. 25, n. 6, IEEE, pp. 852-869, 1999.
- [52] LUPU, E. AND SLOMAN, M. “Towards a role based framework for distributed systems management”, *Journal of Network and Systems Management*, v. 5, n. 1, Plenum Press Publishing, pp. 5-30, Janeiro de 1997.

- [53] MORTENSEN, J. E. “JFS Fuzzy System”. <http://www.inet.uni2.dk/~jemor/jfs.htm>, 1998.
- [54] JACOBSON, V., NICHOLS, K. AND PODURI, K. “An expedited forwarding PHB”. RFC 2598, Junho de 1999.
- [55] INMON, W. H., *Building the Data Warehouse*, Third Edition, John Wiley & Sons, Inc. 2002.
- [56] Pilot Works, “Aligning Execution with Strategy”. <http://www.pilotsoftware.com>, Pilot Software, Maio de 2004.
- [57] WANG, L. X. AND MENDEL, J. “Generating fuzzy rules by learning from examples”, *IEEE Transactions on Systems, Man, and Cybernetics* v. 22, n. 2, julho de 1992.
- [58] COX, E., *Fuzzy logic for business and industry*, Charles River Media, outubro de 1955.
- [59] HERRERA, F., LOZANO, M., AND VERDEGAY, J. “Tuning fuzzy logic controllers by genetic algorithms”, *International Journal of Approximate Reasoning* v. 12, n. 3, pp. 299-315, junho de 1995.
- [60] SANDRI, S. AND CORREA, C. “Lógica Nebulosa”, *Promoção: Conselho Nacional de Redes Neurais pp. c073-c090*, V Escola de Redes Neurais, ITA, São José dos Campos – SP, 19 de julho de 1999.
- [61] MICHAELEWICZ, Z., *Genetic algorithms + data structure = evolution programs*, Springer-Verlag, março de 1996.
- [62] MITCHELL, M., *An Introduction to Genetic Algorithms*, MIT Press, março de 1996.
- [63] DE JONG, K., *An Analysis of the Behavior of a class of Genetic Adaptive System*, Tese de doutorado, University of Michigan, 1975.
- [64] MCCANNE, S. AND FLOYD, S. “ns Simulator – Version 2”. <http://www.isi.edu/nsnam/ns/>, 1998.
- [65] PIEDA, P., ETHRIDGE, J. AND BAINES, M. “A network simulator differentiated

services implementation – Open IP”. <http://www7.nortel.com:8080/CTL/>, outubro de 2000.

[66] ZEGURA, E. W., CALVERT, K. AND BHATTACHARJEE, S. “How to model an internetwork”. In: *IEEE Infocom 96*, San Francisco, USA, março de 1996.

[67] ZEGURA, E. W., CALVERT, K. L. AND DONAHOO, M. J. “A quantitative comparison of graph-based models for internet topology”, *IEEE/ACM Transactions on Networking* v. 5, n. 6, pp. 770-783, dezembro de 1997.

[68] CALVERT, K., DOAR, M. AND ZEGURA, E. W. “Modeling internet topology”, *IEEE Communications Magazine*, junho de 1997.

[69] WAXMAN, B. M. “Routing of multipoint connections”, *IEEE Journal on Selected Areas in Communications* v. 6, n. 9, pp. 1617-1622, 1988.

[70] HSIUNG, H., FISCHER, M. J., MASI, D. M., CUFFIE, D. AND SCHEURICH, S. “An approach to IP telephony performance measurement and modeling in government environments”. In: *9th Annual Conference of the Internet Society, INET'99*, San Jose, USA, julho de 1999.