

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**UMA PLATAFORMA DE CONTROLE, MONITORAMENTO
E AVALIAÇÃO DE DESEMPENHO EM SISTEMAS DE
COMUNICAÇÕES MÓVEIS
UTILIZANDO ALOCAÇÃO DINÂMICA E INTELIGENTE
DE CANAIS**

ERNESTO VASCONCELOS CARVALHO

ORIENTADOR: FRANCISCO ASSIS DE OLIVEIRA NASCIMENTO

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.DM – 010/06

BRASÍLIA/DF: JULHO - 2006

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**UMA PLATAFORMA DE CONTROLE,
MONITORAMENTO E AVALIAÇÃO DE DESEMPE-
NHO EM SISTEMAS DE COMUNICAÇÕES MÓVEIS
UTILIZANDO ALOCAÇÃO DINÂMICA
E INTELIGENTE DE CANAIS**

ERNESTO VASCONCELOS CARVALHO

**TESE SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉ-
TRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE
DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS
PARA A OBTENÇÃO DO GRAU DE DOUTOR**

APROVADA POR:

**Prof. Francisco de Assis Nascimento, Dr. (ENE-UnB)
(ORIENTADOR)**

**Prof. Paulo Henrique Portela de Carvalho, Dr. (ENE-UnB)
(Examinador Interno)**

**Prof. Leonardo R. A. X. Menezes, PhD. (ENE-UnB)
(Examinador Interno)**

**Lúcio Martins da Silva, Dr. (ECT)
(Examinador Interno)**

**Prof. José Neuman de Souza, Dr. (DCC-UFC)
(Examinador Externo)**

BRASÍLIA/DF, 24 DE JULHO DE 2006

FICHA CATALOGRÁFICA

CARVALHO, ERNESTO VASCONCELOS

Uma Plataforma de Controle, Monitoramento e Avaliação de Desempenho em Sistemas de Comunicações Móveis Utilizando Alocação Dinâmica e Inteligente de Canais [Distrito Federal] 2006.

iv, 232 p., 297 mm (ENE/FT/UnB, Doutor, Telecomunicações, 2006).

Tese de Doutorado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Telefonia Celular

3. Sistema Distribuído

2. Sistemas Inteligentes

4. Engenharia de Software

I. ENC/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

CARVALHO, E. V. (2006). Uma Plataforma de Controle, Monitoramento e Desempenho em Sistemas de Comunicações Móveis Utilizando Alocação Dinâmica e Inteligente de Canais. Tese de Doutorado em Engenharia Elétrica, Publicação PPGENE.DM-10/06, Departamento de Engenharia Elétrica, Universidade de Brasília, DF, 232p.

CESSÃO DE DIREITOS

AUTOR: Ernesto Vasconcelos Carvalho

TÍTULO: Uma Plataforma de Controle, Monitoramento e Avaliação de Desempenho em Sistemas de Comunicações Móveis utilizando Alocação Dinâmica e Inteligente de Canais.

GRAU: Doutor

ANO: 2006

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta tese de doutorado pode ser reproduzida sem a autorização por escrito do autor.

Ernesto Vasconcelos Carvalho
SQN 308 – Bl. E – Apto 602 – Asa Norte
70.747-050 BRASÍLIA – DF – Brasil.

AGRADECIMENTOS

Foram vários os momentos pelos quais passei na concretização desse sonho. Agradeço a todos os que, direta e indiretamente, me ajudaram a realizar esse projeto de vida.

Agradeço aos professores das disciplinas pelas quais transitei, a base recebida foi fundamental para o aprimoramento de meus conhecimentos.

Agradeço ao meu orientador Prof. Assis Nascimento, meu maior incentivador na concretização desse trabalho.

Agradeço aos amigos Rogério e Raquel Araújo, os vários momentos de descontração e de incentivo nos momentos mais difíceis.

Agradeço também à minha chefe e amiga, Ana Angélica Carvalhosa, pela flexibilização de horário de trabalho concedido permitindo assim que eu cursasse as disciplinas obrigatórias deste curso.

Por fim, agradeço a Deus pelo Dom da Vida...

**Dedico esta dissertação a meus pais,
José Carlos e Lenira
pelo princípio da existência;
e a Deus, princípio da criação de tudo o que é bom e vivo.**

RESUMO
UMA PLATAFORMA DE CONTROLE, MONITORAMENTO E AVALIAÇÃO DE DESEMPENHO EM SISTEMAS DE COMUNICAÇÕES MÓVEIS UTILIZANDO ALOCAÇÃO DINÂMICA E INTELIGENTE DE CANAIS

Autor: Ernesto Vasconcelos Carvalho

Orientador: Francisco Assis de Oliveira Nascimento

Programa de Pós-graduação em Engenharia Elétrica

Brasília, Julho de 2006.

A telefonia celular atual está convergindo para uma integração dos serviços de voz e dados que deverão ser fornecidos com diferentes perfis de qualidade de serviço. Alguns desses serviços são mais sensíveis ao atraso que outros; o que torna necessário a priorização de alguns sem, contudo, degradar a distribuição dos demais.

Devido à complexidade das redes telefônicas atuais, torna-se imperativo que o processo de avaliação do tráfego e as tomadas de decisão sejam distribuídos ao longo da rede. Permitindo, com isso, a distribuição de informações gerenciais entre máquinas heterogêneas e tratadas por meio de diferentes linguagens.

Baseados nas diferentes Qualidades de Serviços dessas mídias este trabalho propõe um ambiente distribuído que se preocupe com questões como controle, monitoramento e avaliação de desempenho dos diferentes tipos de serviços de uma rede de comunicações. Para tal, este trabalho tem por objetivo uma maior eficiência na alocação dos recursos de rádio das redes telefônicas atuais, para os diferentes perfis de qualidade de serviço, voltados aos sistemas de tráfego FDD (do inglês, *Frequency Division Duplex*) e TDD (do inglês, *Time Division Duplex*).

Evidenciando um melhor dimensionamento desses recursos, é apresentada uma plataforma de monitoramento, avaliação e controle, composta da automação de duas diferentes técnicas de inteligência artificial. O modelo proposto faz uso das técnicas de inteligência computacionais: Redes Neurais Artificiais e Lógica *Fuzzy*, de forma independente, e os compara em relação à alocação fixa convencional (FCA, do inglês, *fixed channel assignment*) largamente empregada nas operadoras atuais. Estas duas técnicas de inteligência computacio-

nal adotadas neste trabalho, são utilizadas para otimização do compartilhamento dos recursos de rádio dessas redes.

Outro objetivo do modelo é abordar a distribuição de suas interfaces e sua portabilidade, uma vez que o modelo se propõe a ser executado em plataformas heterogêneas e implementado sem restrição de uma linguagem específica. O estudo de caso desenvolvido contempla a plataforma de distribuição das interfaces dos serviços (ou funcionalidades) e sua portabilidade; tudo isso, por meio de protocolos comuns à *Internet*.

Além do mais, foram idealizados dois cenários telefônicos para realização do processo de simulação do tráfego: uma micro-célula e outra macro-célula, composta por sete células adjacentes e uma sobreposta, sobre uma plataforma simulada. Sobre eles, são feitas simulações com alocação fixa de canais e com alocação dinâmica por meio de técnicas *fuzzy* e neural, isoladamente.

ABSTRACT

A CONTROLLING, MONITORING AND PERFORMANCE EVALUATION PLATFORM FOR COMMUNICATION MOBILE SYSTEM USING DYNAMIC AND INTELLIGENT ALLOCATION CHANNELS.

Author: Ernesto Vasconcelos Carvalho

Supervisor: Francisco Assis de Oliveira Nascimento

Programa de Pós-graduação em Engenharia Elétrica

Brasília, July of 2006

Cellular telecommunications are converging for voice and data integration, which must be provided by different quality of services. Some of them are more sensible for delay variation, needing use of prioritization schemes without debasing others.

As the telecommunication complexity grows up, it's important that the traffic process evaluation and decisions must be delivered through the network. It enables the spreading of management information through heterogeneous machines and multi-language implementations.

For maintaining Qualities of Services use, it's proposed a distributed environment for controlling, monitoring and evaluating the performance of radio resources. This work leads for a radio research allocation most efficiently than the actual telecommunication networks, treating Frequency Division Duplex and Time Division Duplex traffic systems.

To improve the network dimension, it's provided a controlling platform composed by two distinct artificial intelligent techniques. The model proposed uses: artificial neural network and fuzzy logic in independent ways, comparing then with Fixed Channel Assignment, the most widely spread operation. These two intelligent computing were developed here for radio sharing optimization.

Another model's objective is the studying of distribution and portability, allowing heterogeneous platform and multi-language implementations. The case study treats the services (or functionality) distribution and portability; all through Internet common protocols.

Two different telecommunication scenes were designed for traffic simulation: a micro-cell and a macro-cell, which is formed by seven other adjacent micro-cells overlapped. Simulations with fixed channel allocation and dynamic ones, by neural network and fuzzy logic, are done, separated.

SUMÁRIO

1 - INTRODUÇÃO.....	01
1.1 - PROBLEMA ABORDADO.....	03
1.2 - OBJETIVOS DO TRABALHO.....	04
1.3 - METODOLOGIA ADOTADA.....	05
1.4 - VALIDAÇÃO DA PROPOSTA.....	06
1.5 - ORGANIZAÇÃO DO TRABALHO.....	06
2 - EVOLUÇÃO DOS SISTEMAS DE COMUNICAÇÃO MÓ- VEIS.....	09
2.1 - AS GERAÇÕES TELEFÔNICAS.....	10
2.2 - AMPS E DAMPS (IS-136)	12
2.3 - IS-95.....	13
2.4 - GSM.....	15
2.4.1 - HSCSD.....	17
2.4.2 - GPRS.....	18
2.4.3 - Infra-Estrutura Telefônica.....	20
2.4.3.1 - Link Aéreo.....	22
2.4.3.2 - MSU.....	22
2.4.3.3 - BTS.....	22
2.4.4 - Infra-Estrutura de Voz.....	23
2.4.4.1 - BSC.....	23
2.4.4.2 - MSC.....	23
2.4.4.3 - TSC.....	24
2.4.4.4 - GMSC.....	25
2.4.5 - Infra-Estrutura de Dados.....	25
2.4.5.1 - PCU.....	25
2.4.5.2 - SGSN.....	26
2.4.5.3 - Backbone do GPRS.....	26
2.4.5.4 - GGSN.....	27
2.5 - CARACTERÍSTICAS DOS SERVIÇOS GPRS.....	27

2.5.1 - Aplicações para GPRS.....	27
2.5.2 - Qualidade de Serviço (QoS)	
2.6 - GERENCIAMENTO DE MOBILIDADE, DE SESSÃO E DE ROTEAMENTO.....	29
2.6.1 - Gerenciamento de Mobilidade.....	30
2.7 - ANALISANDO O GPRS PARA ACESSO À INTERNET.....	
3 - METODOLOGIA PARA ALOCAÇÃO DE RECURSOS DE RÁDIO EM REDES TELEFÔNICAS.....	33
3.1 - ALOCAÇÃO DE RECURSOS DE RÁDIO EM REDES TELEFÔNICAS MÓVEIS.....	33
3.2 - ALOCAÇÃO DINÂMICA DE CANAIS.....	36
3.3 - DIFERENÇAS ENTRE NOVAS CHAMADAS E <i>HANDOVERS</i>	37
3.4 - ESTRATÉGIAS PARA DETECÇÃO DE <i>HANDOVER</i>	38
3.4.1 - <i>Handover</i> Controlado pelo Terminal Móvel (MCHO)	40
3.4.2 - <i>Handover</i> Controlado pela Rede (NCHO)	41
3.4.3 - <i>Handover</i> Assistido pelo Terminal Móvel (MAHO)	42
3.5 - TÉCNICAS PARA ATRIBUIÇÃO DO CANAL.....	43
3.6 - ESQUEMAS DE PRIORIZAÇÃO DE <i>HANDOVERS</i>	45
3.6.1 - Reserva de Canais para <i>Handover</i>	46
3.6.2 - Esquema de Empilhamento de Chamadas.....	47
3.6.3 - Empilhamento com Priorização FIFO.....	47
3.6.4 - Empilhamento com Priorização Baseada na Mensuração do Sinal.....	48
3.6.5 - Esquema de Sub-taxação.....	49
3.6.6 - Esquema Híbrido.....	49
3.7 - TRANSMISSÃO PELO ENLACE TELEFÔNICO.....	50
3.7.1 - Analisando o Throughput e o Buffer para o GSM/GPRS.....	50
3.7.2 - Re-roteamento de Tráfego em Redes Telefônicas Móveis.....	
3.7.3 - Análise de Desempenho de Redes GSM com Inteligência de Sobreposição e Subposição.....	51
3.8 - CONSIDERAÇÕES.....	52
4 - ARQUITETURA ORIENTADA A SERVIÇOS VOLTADO À INTERNET.....	54
4.1 - ARQUITETURA DE REDES MÓVEIS.....	55

4.1 - INTEROPERABILIDADE NOS SERVIÇOS WEB.....	56
4.2 - COMPONENTES DO SOA.....	56
4.3 - CARACTERÍSTICAS DO SOA.....	57
4.4 - A TECNOLOGIA DE SERVIÇOS WEB.....	58
4.4.1 - Protocolo SOAP.....	59
4.4.2 - Modelos de Mensagem.....	61
4.4.3 - Elementos de uma WSDL.....	62
4.4.3.1 - Types.....	64
4.4.3.2 - Messages.....	64
4.4.3.3 - Port type.....	65
4.4.3.4 - Binding.....	66
4.4.3.5 - Service e Port.....	67
4.4.3.6 - Chamadas RPC.....	67
4.4.3.7 - Mapeamentos.....	69
4.4.3.8 - Códigos e tratamentos de erros.....	69
4.5 - SERVIÇOS WEB CORPORATIVOS.....	70
4.5.1 - Componentes EJBs.....	
5 - MODELAGEM ADAPTATIVA PARA REDES DE COMUNICA- ÇÕES MÓVEIS.....	72
5.1 - ESPECIFICAÇÃO DO MODELO TELEFÔNICO.....	73
5.2 - META MODELO PARA SERVIÇOS DE TELECOMUNICAÇÕES A- DAPTÁVEIS.....	74
5.3 - A COMUNICAÇÃO EM AMBIENTES DISTRIBUÍDOS.....	79
5.3.1 - A Gerência de Recursos por meio da WEB.....	80
5.3.2 - Os Serviços WEB utilizados.....	82
5.3.3 - Modelo de uso dos Serviços WEB Orientados a RPC.....	
5.4 - SISTEMA DE DISTRIBUIÇÃO DO GERENCIAMENTO DAS INFOR- MAÇÕES.....	84
5.5 - AMBIENTE DE SOBREPOSIÇÃO DO MODELO DE INTELIGÊNCIA..	87
5.6 - MODELAGEM PROPOSTA PARA OTIMIZAÇÃO INTELIGENTE DE CANAIS.....	89
5.7 - BASE DE INFORMAÇÕES DE GERENCIAMENTO (MIB) DO MO-	95

DELO TELEFÔNICO.....	98
5.8 - CONSIDERAÇÕES.....	
	100
6 - SIMULAÇÕES E IMPLEMENTAÇÃO DO MODELO.....	103
6.1 - PROJETO DO SISTEMA <i>FUZZY</i>	106
6.1.1 - Motor de Inferência.....	107
6.2 - PROJETO DO SISTEMA NEURAL.....	108
6.3 - MODELAGEM DO TRÁFEGO TELEFÔNICO.....	111
6.4 - CENÁRIOS DE SIMULAÇÃO.....	113
6.5 - CRITÉRIOS DE MODELAGEM DO TRÁFEGO.....	115
6.6 - O PROCESSO DE EXECUÇÃO.....	115
6.6.1 - A simulação do Tráfego Telefônico.....	121
6.6.2 - Inferência nos Sistemas Inteligentes.....	121
6.6.2.1 - Inferência <i>Fuzzy</i>	121
6.6.2.2 - Inferência Neural.....	122
6.7 - RESULTADOS DA SIMULAÇÃO.....	123
6.7.1 - Primeiro Cenário – Micro-célula GSM/GPRS.....	
6.7.2 - Segundo Cenário – Micro-célula sobreposta a uma Macro-célula GSM/GPRS.....	133 138
6.7.3 - Análise dos Resultados.....	141
6.8 - CONSIDERAÇÕES.....	
	142
7 - CONCLUSÃO.....	144
7.1 - TRABALHOS FUTUROS.....	
	146
REFERÊNCIAS BIBLIOGRÁFICAS.....	
	158
APÊNDICES.....	159
A – LINGUAGEM DE DESCRIÇÃO DE SERVIÇOS WEB – WSDL.....	196
A.1 – PARÂMETROS DO SERVIDOR DE SERVIÇOS WEB.....	171
B - DIAGRAMATIZAÇÃO DA ARQUITETURA VIA UML.....	179

C - CONTEÚDO DO ARQUIVO EJB-JAR.XML.....	179
C.1 - CONTEÚDO DO ARQUIVO EJB-JAR.XML.....	181
C.2 - CONTEÚDO DO ARQUIVO JBOSS.XML.....	181
C.3 - REFERÊNCIAS A EJBS.....	183
D - CONTROLADOR FUZZY.....	186
D.1 - REGRAS DO CONTROLADOR FUZZY.....	
D.2 - RESULTADOS DE PONTOS DISCRETOS DA OTIMIZAÇÃO FUZZY PARA O CENÁRIO DE MICRO-CÉLULA.....	189
D.3 - RESULTADOS DE PONTOS DISCRETOS DA OTIMIZAÇÃO FUZZY PARA O CENÁRIO DE MACRO-CÉLULA.....	190
E - REDE NEURAL ARTIFICIAL.....	191
E.1 - TREINAMENTO DA REDE NEURAL.....	193
E.2 - ROTEIRO DE EXECUÇÃO.....	
E.3 - RESULTADOS DE PONTOS DISCRETOS DA OTIMIZAÇÃO NEURAL PARA O CENÁRIO DE MICRO-CÉLULA.....	194
E.4 - ROTEIRO DE EXECUÇÃO DO CENÁRIO DE MACRO-CÉLULA.....	195
E.5 - RESULTADOS DE PONTOS DISCRETOS DA OTIMIZAÇÃO NEURAL PARA O CENÁRIO DE MACRO-CÉLULA.....	197
F - MIB INTELIGENTE.....	201
F.1 - MIB NEURAL-FUZZY.....	225
G - AGENTE DE DADOS INTELIGENTE.....	

LISTA DE TABELAS

Tabela 6.1 – Configurações Básicas da Micro-Célula.....	123
Tabela 6.2 – Avaliação dos Resultados do Gráfico 6.14 (Micro-Célula)	138
Tabela 6.3 – Avaliação de Resultados do Gráfico 6.27 (Macro-Célula)	138
Tabela 6.4 – Parâmetros de Desempenho FCA para os cenários, com $A_{off} = 21$ Erlangs.....	139
Tabela 6.5 – Parâmetros de Desempenho usando Inteligência Artificial para ambos os cenários, com $A_{off} = 21$ Erlangs.....	141

LISTA DE FIGURAS

Figura 2.1 - Percurso de Atualização da rede telefônica para a Terceira Geração.....	11
Figura 2.2 - Estrutura dos <i>Slots</i> , dos frames e dos multiframe no GSM.....	16
Figura 2.3 - Percurso da Voz em uma Rede GSM.....	21
Figura 2.4 - Percurso dos dados WAP em uma rede GSM/GPRS.....	21
Figura 2.5 - Subsistemas da base de dados da MSC.....	24
Figura 2.6 - Diagrama de Estados para uma Estação Móvel.....	30
Figura 3.1 - Técnicas de Alocação de Rádio em Redes 2½ G.....	35
Figura 3.2 – Processo de Manutenção da Qualidade do Terminal Móvel.....	41
Figura 3.3 – Estratégias de Atribuição de Frequência.....	43
Figura 3.4 – Fluxograma para Esquema sem Priorização.....	44
Figura 3.5 – Fluxograma para Esquema de Canais Reservados para <i>Handover</i>	45
Figura 3.6 – Fluxograma para o Esquema de Empilhamento de <i>Handovers</i>	47
Figura 3.7 – Fluxograma para Esquema de Sub-taxação.....	49
Figura 4.1 – Interação entre os componentes SOA e Sistemas Legados.....	56
Figura 4.2 – Estrutura de empacotamento do projeto.....	70
Figura 5.1 – Ambiente de oferecimento de serviços.....	75
Figura 5.2 – Representação gráfica de uma associação explícita entre os filtros.....	77
Figura 5.3 – Infra-estrutura e acesso a provedores de serviços.....	77
Figura 5.4 – Interação entre serviços WEB, registro e consumo do serviço.....	81
Figura 5.5 – Um roteador SOAP com base em Serviços WEB.....	83
Figura 5.6 - Modelo de Negócio para o desenvolvimento de padrões abertos.....	84
Figura 5.7 – Modelo de Distribuição de Processo.....	87
Figura 5.8 – Arquitetura das Redes PCS.....	87
Figura 5.9 – Protocolo de Telefonía Proposto.....	91
Figura 5.10 – Modelo Proposto para Alocação Dinâmica e Inteligente de Canais.....	92
Figura 5.11 - Rede Neural adotada.....	94
Figura 5.12 – Representação da MIB em árvore.....	96
Figura 6.1 – Modelo do Motor de Inferência <i>Fuzzy</i> utilizado.....	106
Figura 6.2 – Topologia da Rede Neural Artificial de 3 Camadas Utilizada.....	108

Figura 6.3 – Elementos Estruturais do Modelo de Simulação.....	110
Figura 6.4 – Cenário de classificação de Canais em uma Estação Rádio Base.....	112
Figura 6.5 – Disposição de Células Telefônicas.....	112
Figura 6.6 – Classificação de Canais em um Estação Rádio Base com priorização de canais.....	113
Figura 6.7 – Diagrama de Atividades.....	117
Figura 6.8 – Escalonador de Eventos.....	117
Figura 6.9 – Gerador de Tráfego.....	118
Figura 6.10 – Servidor da BTS (Priorização de <i>Handover</i>)	118
Figura 6.11 – Servidor da BTS (Empilhamento de Novas Chamadas)	119
Figura 6.12 - Servidor da BTS (Empilhamento de <i>Handover</i>)	119
Figura 6.13 - Servidor da BTS (Sub-taxação de Chamadas)	120
Figura 6.14 – Comparação dos desempenhos dos tráfegos em uma Micro-célula.....	125
Figura 6.15 – Taxa de Perda quando alocados canais de dados.....	125
Figura 6.16 – Carga da Rede.....	126
Figura 6.17 – Taxa de Perda quando alocados canais para priorização de <i>handover</i>	127
Figura 6.18 – Carga da Rede.....	128
Figura 6.19 – Taxa de Perda de chamadas com tipo de reserva de canais: pré/pós-reserva.....	128
Figura 6.20 – Carga da Perda.....	129
Figura 6.21 – Taxa de Perda quando utilizada uma Técnica de Empilhamento de Chamadas.....	130
Figura 6.22 – Carga da Rede.....	131
Figura 6.23 – Taxa de Perda quando utilizada uma Técnica de Sub-taxação de Canais.....	131
Figura 6.24 – Carga da Rede.....	132
Figura 6.25 – Comparação entre várias das taxas de perdas.....	132
Figura 6.26 – Cenário de Sobre-posição Macro-células/Micro-célula.....	134
Figura 6.27 – Comparação dos desempenhos dos tráfegos em Micro-célula/Macro-célula.....	135
Figura 6.28 – Taxa de perda quando alocados canais de dados.....	135
Figura 6.29 – Carga da Rede.....	136
Figura 6.30 – Taxa de Perda quando alocados canais para priorização de <i>handover</i>	136
Figura 6.31 – Carga da Rede.....	137

Figura 6.32 – Comparação entre as várias taxas de perdas.....	137
Figura B.1 – Diagrama de Caso de Uso para acesso direto ao servidor.....	171
Figura B.2 – Diagrama de Caso de Uso para acesso a partir da aplicação Cliente de interface com o serviço WEB.....	171
Figura B.3 – Classe br.unb.fat.dee.doutorado.sti.FachadaTelefonica.java.....	172
Figura B.4 – Classe br.unb.fat.dee.doutorado.sti.TELCOM.gerente.Amostra.....	174
Figura B.5 – Classe br.unb.fat.dee.doutorado.sti.TELCOM.gerente.macrocell. MacrocellBean e br.unb.fat.dee.doutorado.sti.TELCOM.gerente. microcell.MicrocellBean.....	175
Figura B.6 – Estrutura da aplicação cliente para acesso ao serviço.....	176
Figura B.7 – Estrutura da aplicação servidora que implementa o serviço.....	177
Figura D.1 – Tráfego Ofertado (Erlangs)	183
Figura D.2 – Canais GPRS (quantidade de canais)	184
Figura D.3 – Canais de Handover (quantidade de canais)	184
Figura D.4 – Canais Sub-taxados (quantidade de canais)	184
Figura D.5 – Percentual de tráfego simulado de Novas Chamadas.....	185
Figura D.6 – Percentual de tráfego simulado de tráfego de Handover.....	185
Figura D.7 – Formas de Esquema de Reservas de Canais: Pré-Reserva ou Pós-Reserva.....	185
Figura D.8 – Formas de Esquema de Empilhamento de Canais: FIFO, MBP e NONE	186
Figura E.1 – Topologia da Rede Neural Artificial.....	191
Figura F.1 – Estrutura da .iso.org.dod.internet.private.enterprises.ERB-FUZZY-NEURAL-MIB (parte 1)	199
Figura F.2 – Estrutura da .iso.org.dod.internet.private.enterprises.ERB-FUZZY-NEURAL-MIB (parte 2).....	200

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES

AMPS	– <i>Advanced/American Mobile Phone System</i>
API	– <i>Application Programming Interface</i>
ASN.1	– <i>Abstract Syntax Notation 1</i>
ATM	– <i>Asynchronous Transfer Mode</i>
AuC	– <i>Authentication Center</i>
BEC	– <i>Backward Error Correction</i>
BER	– <i>Bit Error Rate</i>
BS	– <i>Base Station ou ERB</i>
BSC	– <i>Base Station Controller</i>
BTS	– <i>Base Transceiver Station</i>
CAC	– <i>Controle de Admissão de Chamadas</i>
CDMA	– <i>Code Division Multiple Access</i>
CDPD	– <i>Cellular Digital Packet Data</i>
C/I	– <i>Taxa de Ruído</i>
CORBA	– <i>Common Object Request Broker Architecture</i>
CP	– <i>Complete Partitioning</i>
CS	– <i>Complete Sharing</i>
DAMPS	– <i>Digital Advanced/American Mobile Phone System</i>
DCOM	– <i>Microsoft Distributed Component Object Model</i>
DeMUX	– <i>Demultiplexador</i>
DMTF	– <i>Desktop Management Task Force</i>
EDGE	– <i>Enhanced Data Rates for GSM Evolution</i>
EIR	– <i>Equipment Identity Register</i>
EJB	– <i>Enterprise Java Bean</i>
ERB	– <i>Estação Rádio Base (do inglês, Base Station ou BS)</i>
FCA	– <i>Fixed Channel Assignment</i>
FCC	– <i>Federal Communications Commission</i>
FDD	– <i>Frequency Division Duplex</i>
FDMA	– <i>Frequency Division Multiple Access</i>
FIFO	– <i>First-In First-Out</i>
FTP	– <i>File Transfer Protocol</i>

GGSN	– <i>Gateway GPRS Support Node</i>
GMDO	– <i>Guidelines for Definitions of Managed Options</i>
GMSC	– <i>Gateway Mobile Switching Center</i>
GMSK	– <i>Gaussian Minimum Shift Keying</i>
GPRS	– <i>General Packet Radio Service</i>
GR	– <i>GPRS Register</i>
GSM	– <i>Global System for Mobile Communication</i>
GSN	– <i>GPRS Service Node</i>
GTP	– <i>GPRS Tunnel Protocol</i>
HCS	– <i>Hierarchical Cell Structure</i>
HLR	– <i>Home Location Register</i>
HSCSD	– <i>High Speed Circuit Switched Data</i>
HTTP	– <i>Protocolo de Transferência de Hipertexto</i>
IDL	– <i>Interface Description Language</i>
IETF	– <i>Internet Engineering Task Force</i>
IIP	– <i>Internet Inter-ORB Protocol</i>
IN	– <i>Intelligent Network</i>
IQ	– <i>Indicador de Qualidade</i>
ISDN	– <i>Integrated Service Digital Network.</i>
ISO	– <i>International Organization for Standardization</i>
IUO	– <i>Múltiplas Camadas de Célula</i>
J2EE	– <i>Java 2 Enterprise Edition</i>
J2SE	– <i>Java 2 Standard Edition</i>
JRMP	– <i>Java Remote Method Protocol</i>
JSR	– <i>Java Specification Request</i>
LLC	– <i>Logical Link Control</i>
MAHO	– <i>Mobile-Assisted Handoff</i>
MBPS	– <i>Measurable Band Priority Schema</i>
MCHO	– <i>Mobile-Controlled Handoff</i>
MIB	– <i>Base de Informações de Gerenciamento</i>
MIME	– <i>Multipurpose Internet Mail Extensions</i>
MPDCH	– <i>Master Packet Data Channel</i>
MS	– <i>Mobile Station</i>
MSC	– <i>Mobile Switching Center</i>

MSU	– <i>Mobile Station Unit</i>
NCHO	– <i>Network-Controlled Handoff</i>
NMF	– <i>Network Management Forum</i>
NMT	– <i>Network Mobile Telephone</i>
NPS	– <i>Non-Priority Schema</i>
ODP	– <i>Open Distributed Processing</i>
OID	– <i>Identificador do Objeto</i>
ORB	– <i>Object Request Broker</i>
OSF	– <i>Open Software Foundation</i>
OSR	– <i>Oversampling ratio</i>
PBCCH	– <i>Packet Broadcast Control Channel</i>
PCCCH	– <i>Packet Common Control Channel</i>
PCS	– <i>Personal Communication Services</i>
PCU	– <i>Packet Control Unit</i>
PDC	– <i>Personal Digital Cellular</i>
PDCH	– <i>Packet Data Channel</i>
PDN	– <i>Packet Data Network</i>
PDA	– <i>Personnel Digital Assistant</i>
PDTCH	– <i>Packet Data Traffic Channel</i>
PDP	– <i>Packet Data Protocol</i>
PRACH	– <i>Packet Random Access Channel</i>
PS	– <i>Partial Sharing</i>
PSTN	– <i>Public Switched Telephony Network</i>
P-TMSI	– <i>Packet Temporary Mobile Subscriber Identity</i>
QoS	– <i>Quality of Service</i>
QPS	– <i>Queue Priority Schema</i>
RCS	– <i>Reserved Channel Schema</i>
RLC	– <i>Radio Link Control</i>
RMI	– <i>Remote Method Invocation</i>
RMON	– <i>Remote Monitoring</i>
RPC	– <i>Remote Procedure Call</i>
RSSI	– <i>Tamanho do Sinal Recebido</i>
SCM	– <i>Service Control Manager</i>
SEI	– <i>Service Endpoint Interface</i>

SGSN	– <i>Serving GPRS Support Node</i>
SMI	– <i>Structure of Management Information</i>
SMS	– <i>Short Message Service</i>
SOA	– <i>Service Oriented Architecture</i>
SOAP	– <i>Simple Object Access Protocol</i>
SPDCH	– <i>Slave Packet Data Channel</i>
SMTP	– <i>Simple Mail Transfer Protocol</i>
SNDCP	– <i>Subnetwork Dependent Convergence Protocol</i>
SRS	– <i>Sub-Rated Schema</i>
TACS	– <i>Total Access Communications Systems</i>
TCP/IP	– <i>Transport Control Protocol/Internet Protocol</i>
TDD	– <i>Time Division Duplex</i>
TDMA	– <i>Time Division Multiple Access</i>
TMN	– <i>Telecommunications Management Network</i>
TRX	– <i>Transceiver</i>
TSC	– <i>Trunking Switching Center</i>
UDDI	– <i>Universal Description Discovery and Integration</i>
UML	– <i>Linguagem de Modelagem Unificada</i>
UMTS	– <i>Universal Mobile Telecommunications System</i>
URI	– <i>Uniform Resource Identifier</i>
VLR	– <i>Visitors Location Register</i>
WAP	– <i>Wireless Application Protocol</i>
WEI	– <i>Indicador de Erro</i>
WSDL	– <i>Linguagem de Descrição de Serviços WEB</i>
WS-I	– <i>WEB Service Interoperability Organization</i>
WWW	– <i>World Wide Web</i>
XML	– <i>Extensible Markup Language</i>

1 - INTRODUÇÃO

Atualmente, a Internet é um importante meio de distribuição e de acesso à informação. O crescimento do número de novos serviços, como protocolo WWW para navegação em sítios da Internet, transferência de arquivos em larga escala e a altas taxas de transmissão, distribuição de *streams* de áudio e vídeo sobre demanda, ou mesmo a realização de vídeo conferências ou tele-conferências, tem provocado um crescimento exponencial da carga de pacotes de dados a que as redes são submetidas. Tal crescimento demonstra claramente que a Internet está se fundamentando mais e mais na realização da maioria das atividades diárias.

De forma similar, as redes móveis estão se submetendo a um crescimento exponencial do número de usuários. Este número já ultrapassou a marca dos dois bilhões. O crescimento se deve principalmente aos mercados onde o setor é menos desenvolvido como China, Índia, Leste Europeu, América Latina e África [1]. Na Finlândia e em outros países, o número de assinantes móveis já ultrapassa o número de linhas fixas [2].

Com base nessas duas realidades de mercado, o crescimento da Internet e o sucesso das redes móveis, dá-se, inevitavelmente, uma convergência de transmissão das aplicações de dados (FTP – Protocolo de Transmissão de Arquivos, WWW – do inglês, *World Wide Web*, e-mail) [3] sobre a malha telefônica, tradicionalmente de voz.

Uma vez que a malha telefônica foi projetada, originalmente, para a transferência de voz por meio de transmissão analógica, a incorporação de diferentes tipos de tráfego traz uma série de complicações. Dentre elas estão a necessidade de se tratar a interferência ou ruído entre as diferentes transmissões e a necessidade de se tratar os diferentes perfis de tráfego por meio de diferentes níveis de qualidade de serviços, priorizando alguns em detrimento de outros menos exigentes.

Este trabalho objetiva a investigação da alocação eficiente e dinâmica de canais de tráfego nos recursos de rádio transmissão da malha telefônica, otimizando o desempenho de tráfego efetivo por meio do uso de técnicas de inteligência computacionais. Em geral, as tecnologias atuais alocam os recursos dos canais estaticamente, ou seja, de forma fixa sem qual-

quer variação em seus perfis de funcionamento. Isso se dá devido a um bom planejamento feito preliminarmente, buscando idealizar o volume e os perfis de tráfego que venham a ser conduzidos sobre o conjunto da rede em questão. Sucintamente, este trabalho tem como objetivo a abordagem científica de cinco pontos distintos:

- 1 – Desenvolvimento e adaptação para microcomputador de uma plataforma para a simulação de redes móveis de comunicações e, desenvolvimento de interfaces para a administração dos recursos disponíveis: canais de comunicação.
- 2 - Construção de um banco de dados com parâmetros que avaliem o desempenho de uma rede com alocação estática de canais de voz e dados.
- 3 – Desenvolvimento de aplicativo para alocação dinâmica para canais de dados e de voz com base em técnicas inteligentes: redes neurais artificiais e lógica *fuzzy*.
- 4 – Avaliação de desempenho do uso de técnicas inteligentes comparando com a tecnologia atual onde a alocação de canais é fixa.
- 5 – Simulação de cenários experimentais.

Nesse contexto, a Engenharia de Serviços de Telecomunicações trata aspectos de especificação, projeto, implementação, gerenciamento e validação dos serviços de telefonia, além de sua implantação sobre arquiteturas de rede existentes e futuras [3]. Seu desafio é orientar na construção de uma infra-estrutura de comunicação baseada em duas características básicas: (i) a integração dos vários serviços em um único sistema de comunicação, e (ii) a criação de um ambiente, no qual esses serviços sejam facilmente criados, alterados e adaptados às novas necessidades e condições. É cada vez mais difícil encontrarmos arquiteturas únicas, genéricas o suficiente para acomodar os diferentes serviços e adaptações através de parâmetros simples ou opções pré-definidas.

Buscando adequar este trabalho à disciplina de Engenharia de Serviços de Telecomunicações, é apresentado o modelo de uma arquitetura que viabiliza a aglomeração de diferentes protocolos e serviços, os quais são facilmente modularizados, facilitando sua re-engenharia para novos cenários e infra-estruturas telefônicas. Este modelo está voltado ao conceito de SOA (Arquitetura Orientada a Serviços) [4], preocupando-se com questões como: portabilidade, escalabilidade e distribuição de informações para outros sistemas legados.

Atualmente, são usados, cada vez mais, diferentes arquiteturas e modelos de objetos integrados pela engenharia de serviços de telecomunicações. Os principais trabalhos baseiam-

se no suporte oferecido por plataformas CORBA (do inglês, *Common Object Request Broker Architecture*) [5] ou ODP (do inglês, *Open Distributed Processing*) [6]. O modelo aqui proposto faz uso de protocolos padrões da própria Internet viabilizando a comunicação entre serviços que utilizam o protocolo HTTP (Protocolo de Transferência de Hipertexto). Isso facilita a integração da arquitetura proposta com plataformas dos sistemas legados existentes.

Nesse sentido, este trabalho é motivado pelo interesse em propor um modelo distribuído com o uso de técnicas de inteligência, para o controle de canais telefônicos, por um sistema de gerenciamento. Tal modelo possibilita a distribuição de seus módulos e dos objetos gerenciados da rede, permitindo transparência na administração dos recursos. Essas informações podem ser disponibilizadas tanto para clientes internos quanto externos à corporação, permitindo assim, que as atividades de administração da rede sejam realizadas em qualquer lugar e a qualquer tempo.

1.1 - PROBLEMA ABORDADO

Ideologicamente, uma operadora de transmissão de voz, ao acrescentar o serviço de dados à sua rede, deveria garantir a mesma disponibilidade do serviço anteriormente fornecido, com a mesma qualidade que antes. Isso seria facilmente alcançado, caso novos canais fossem adicionados exclusivamente para a transmissão destes dados. Infelizmente, isso não acontece na prática, porque a aquisição de novos canais custa excessivamente caro para as operadoras, além de nem sempre serem disponibilizados pelas autoridades governamentais.

Por conta da escassez destes recursos de rádio é fundamental a utilização de técnicas ou ferramentas que otimizem a alocação e desalocação dos canais de transmissão nas Estações Rádio Base. Algo que não acontece com a maioria das operadoras, uma vez que elas determinam a quantidade de canais alocados para cada serviço, anteriormente, quando do planejamento da malha telefônica.

Uma das grandes dificuldades encontradas em arquiteturas de QoS (Qualidade de Serviços) surge da diversidade com que os aspectos de qualidade são especificados e tratados nos diferentes sistemas que compõem o ambiente de comunicação e processamento. A modelagem de serviços de acesso por meio da abstração de uma rede telefônica propõe

uma forma de tornar mais homogêneos o tratamento da QoS em ambientes de processamento e de comunicação [3]. Esse tipo de abordagem ganha importância no mundo atual, onde a mudança tecnológica nos meios de comunicação é crescente e veloz.

1.2 - OBJETIVOS DO TRABALHO

Este trabalho possui dois grandes objetivos. Primeiramente, propor um ambiente genérico e adaptável ao monitoramento, controle e otimização de recursos de rádio das diferentes redes telefônicas. Em seguida, fazer uso de técnicas de inteligência artificial como forma de garantir a otimização destes recursos em relação à forma de alocação fixa de canais.

Tal ambiente deve se preocupar com questões como: portabilidade, distribuição das informações, escalabilidade, além do controle e diagnóstico pró-ativo e reativo das incidências de desempenho.

Este ambiente é composto por uma arquitetura modular e em camadas, permitindo sua instanciação sobre os vários componentes de uma rede de telefonia móvel, como: ERBs (Estação Rádio-Base), SGSN (do inglês, *Serving GPRS Support Node*), GGSN (do inglês, *Gateway GPRS Support Node*) e os dispositivos móveis. A distribuição das informações, que trafegam sobre os diversos módulos, é feita por meio de um ambiente de distribuição de serviços baseado nos conceitos de SOA e implementado na forma de um Serviço WEB. Esta abordagem proporciona a distribuição de diferentes ações (atividades) ao longo de toda a malha telefônica e fora dela, auxiliando no gerenciamento do tráfego e balanceamento de carga entre múltiplas instâncias de um mesmo serviço.

O processo de avaliação das informações monitoradas nesta tese faz uso de duas técnicas de inteligência artificial: redes neurais multicamadas e controladores *fuzzy*. Ambos são usados como forma de otimizar o tráfego, permitindo a automatização das tomadas de decisão, controlando a malha telefônica e permitindo uma otimização automática dos recursos de rádio; bem como, informando ao administrador a situação apresentada de tais recursos por meio dos serviços de gerenciamento.

Os principais atributos monitorados para permitir a avaliação de desempenho sobre o caso proposto são:

- as possíveis taxas de erro, como a probabilidade de falha de *handover* e a probabilidade de bloqueio de novas chamadas; e,
- as taxas de vazão, como a carga versus o tráfego ofertado, a taxa de utilização por canal e, em particular, o desempenho dos canais, tais como: vazão, retardo, etc.

1.3 - METODOLOGIA ADOTADA

Esta tese implementa um ambiente de simulação que obedece aos critérios acima expostos de: portabilidade, escalabilidade, distribuição das informações, controle e diagnóstico preventivo de incidentes.

O critério de portabilidade é tratado por meio do desenvolvimento em linguagem de programação JAVA e C++, obedecendo aos princípios da análise e projeto orientado a objetos, com as suas classes modeladas e apresentadas em documentação padrão, no apêndice B deste trabalho.

O critério de escalabilidade é aqui tratado pela ótica da disponibilização, bem definida, dos serviços executados. Cada conjunto de funcionalidades (ou serviços de aplicação) é disponibilizado segundo o paradigma de Serviços WEB. Esse paradigma vem sendo considerado a próxima onda da revolução da internet.

A distribuição das informações deve obedecer a critérios de comunicação entre módulos de serviços. Com atribuições muito bem definidas, cada módulo é implementado por meio de um EJB (do inglês, *Enterprise Java Bean*) do tipo *Session Bean*. Essa abordagem garante uma série de serviços a nível de sistema, dentre eles, temos: serviços de ciclo de vida do componente, de persistência, de nomes, de concorrência, de transação, de relacionamento, de localização, dentre outros.

O critério de controle da rede telefônica é implementado por meio do modelo proposto para o seu gerenciamento. O modelo está concebido de forma a permitir um contínuo monitoramento dos dispositivos de canais que visam a alocação de dados e de voz. Este monitoramento, conforme concebido, permite que o administrador da rede possa readequar os parâmetros de configuração desses canais.

Por fim, o critério de diagnóstico preventivo de incidentes é tratado através de estratégias de inteligência artificial para alocação de recursos de rádio. As técnicas de lógica *fuzzy* e redes neurais artificiais são analisadas separadamente e comparadas à alocação fixa dos mesmos.

1.4 - VALIDAÇÃO DA PROPOSTA

Na abordagem prática, a simulação é realizada sobre dois cenários separadamente. O primeiro, e mais simples cenário, é formado por uma micro-célula do sistema móvel celular e, o segundo, por uma macro-célula composta por sete células adjacentes, sobrepostas.

Em ambos os cenários foram estudados a alocação de tráfego originário de novas chamadas e a realização de procedimentos de *handover* sobre os diferentes níveis de carga a que as células são submetidas. Analisou-se preferencialmente sua taxa de perda em relação à taxa efetivamente aproveitada.

1.5 - ORGANIZAÇÃO DO TRABALHO

Este trabalho é constituído de sete capítulos, acrescidos os apêndices, onde são apresentados alguns dos resultados obtidos. Neste primeiro capítulo, deu-se uma visão ao leitor dos tópicos que virão a seguir, juntamente com a metodologia adotada e os focos de contribuição a serem abordados. Esses pontos de contribuição foram explanados de forma mais geral neste capítulo.

O segundo capítulo apresenta uma explanação sobre os fundamentos da Comunicação Móvel, os pontos fundamentais em cada geração tecnológica e o estado da arte em que eles se encontram.

O terceiro capítulo foca o gerenciamento de mobilidade em dispositivos móveis, tratando de questões como o gerenciamento de *handovers* além de particularidades dos canais de rádio transmissão.

O capítulo quatro dá uma visão geral dos fundamentos e funcionalidades de uma arquitetura orientada a serviços. Essa arquitetura é analisada sobre o enfoque de um serviço WEB, vinculando a implementação dessa funcionalidade ao caso proposto nesta tese.

O quinto capítulo apresenta o ambiente proposto neste trabalho, juntamente com a evolução do modelo definido no plano conceitual e atendendo aos critérios anteriormente apresentados. Neste capítulo, enfatizamos o ambiente de distribuição também com os seus vários serviços disponíveis; em especial, aqueles que fazem parte do módulo de inteligência artificial.

Sua arquitetura funcional é seguida das técnicas de geração de tráfego e critérios de avaliação de mobilidade adotados e apresentados no capítulo sexto. Assim, o capítulo sexto apresenta os diferentes cenários desenvolvidos, juntamente com as características do tráfego simulado, os resultados obtidos por meio do processo destas simulações em ambiente de alocação fixa de canais e pelo uso de ferramentas inteligentes.

O capítulo sete se refere às conclusões encontradas, abordando as várias contribuições que fazem parte deste trabalho.

Por fim, os apêndices trazem detalhes adicionais sobre a implementação deste trabalho.

O apêndice A traz o script WSDL (Linguagem de Descrição de Serviços WEB) usado para garantia do critério de heterogeneidade do modelo.

O apêndice B mostra uma diagramação UML (Linguagem de Modelagem Unificada) para o modelo proposto.

O apêndice C apresenta os scripts EJBs definidos.

O apêndice D mostra a estrutura do controlador fuzzy usado na implementação do módulo de inteligência do modelo.

O apêndice E mostra a estrutura da rede neural artificial usado na implementação do módulo de inteligência do modelo.

O apêndice F apresenta a MIB (Base de Informações de Gerenciamento) usada para armazenamento de dados das informações de inferência do modelo.

Por fim, o apêndice G apresenta alguns dos valores armazenados na estrutura de dados explanada no apêndice anterior.

2 - EVOLUÇÃO DOS SISTEMAS DE COMUNICAÇÃO MÓVEIS

Conforme a Internet se expandia, as redes sem fio seguiam um crescimento semelhante. Isso tem forçado um acoplamento das duas tecnologias, tornando a conectividade móvel mais veloz para suportar os novos requisitos. A evolução dessas tecnologias móveis e de rádio nas últimas duas décadas possibilitou o desenvolvimento de novas tecnologias móveis que são coletivamente denominadas PCS (Serviços de Comunicação Pessoais). Elas viabilizam o acesso a voz de alta qualidade, dados e informações de multimídia, a qualquer hora ou lugar, e interação entre diferentes padrões e protocolos que compõe a plataforma celular.

No panorama geral, o PCS representa a segunda geração da tecnologia sem fio, intermediária entre os telefones celulares analógicos (primeira geração) e a multimídia móvel de alta largura de banda (gerações seguintes). O PCS inclui todas as formas de comunicações telefônicas via rádio que são interconectadas a uma rede pública de transmissão PSTN (do inglês, *Public Switched Telephone Network*), incluindo os seguintes atributos:

- serviço difuso, para prover a mobilidade dos usuários;
- terminais e tarifas de baixo custo;
- compatibilidade com padrões de mercado;
- compactação dos aparelhos móveis.

A evolução recente nos aspectos de rádio-transmissão, acompanhado do crescimento da largura de banda com conexões mais velozes a custos reduzidos são ingredientes chave para os aspectos da mobilidade dos PCSs.

O PCS foi originalmente descrito pela FCC (do inglês, *Federal Communications Commission*) como um grupo especial de licenças para a banda de 1900 MHz, mas foi logo estendido para incluir qualquer sistema digital que forneça voz de alta qualidade e dados de banda estreita. Dessa forma, a estrutura básica das redes celulares continua a mesma, ainda que utilizada para a multimídia de terceira geração.

O PCS é classificado nas 3 categorias seguintes, todas elas fornecem recursos diferentes ao usuário final [7]:

- Os celulares digitais, oferecem voz de alta qualidade e dados limitados; usualmente apenas texto. Ex. GSM, HSCSD, GPRS, D-AMPS, PDC e cdmaOne.
- Outros serviços utilizam a rede celular, mas somente para a transmissão de dados. Ex. CDPD, Mobitex, MCDN.
- por fim, as tecnologias não-celulares que funcionam sem uma estação base. Embora pouco conhecidas, são importantes para serviços de emergência e outros negócios. Ex. TETRA, iDEN.

Este trabalho está inserido no contexto dos sistemas móveis PCS, oferecendo uma alta qualidade de voz sobre uma quantidade de dados limitados nos celulares digitais.

2.1 - AS GERAÇÕES TELEFÔNICAS

Os diversos padrões telefônicos foram classificados pela comunidade em três grupos, caracterizando as três gerações da telefonia móvel.

A Primeira Geração (1G) é representada pelos celulares de transmissão totalmente analógica, ou seja, as ondas transmitidas são contínuas no tempo. São utilizados apenas para voz e são bastante sensíveis às interferências. A falta de garantias na privacidade das ligações é um grande fator de insegurança.

A Segunda Geração (2G) é representada pelos PCSs, anteriormente comentado. A sua transmissão é totalmente digital permitindo uma maior segurança por meio de técnicas de criptografia. Em geral, incluem serviços de mensagens, como correio de voz e identificador de chamadas. Os dados são transmitidos a taxas inferiores a 10 Kbps.

A largura de banda foi ampliada para permitir a transmissão de dados para velocidades semelhantes às de um modem rápido. Denominamos esta geração é denominada de 2½ G (ou 2,5G). O WAP (do inglês, *Wireless Application Protocol*), que utiliza uma versão compacta da Web, também está embutido nesta geração.

A Terceira Geração (3G) será essencial para os serviços de Internet sem Fio. Proporcionará acesso permanente a WEB, vídeo conferências e alta qualidade de voz. No lugar de telefones, muitos terminais serão pequenos computadores ou PDAs (do inglês, *Personal Digital Assistants*).

Muitas das futuras aplicações ainda são desconhecidas, mas os especialistas do setor afirmam que serão descobertas com o passar do tempo. Entende-se que as redes 3G serão capazes de fornecer aos usuários móveis o desempenho das redes ISDN (do inglês, *Integrated Service Digital Network*) ou superior – pelo menos 144 kbps. O objetivo será oferecer multimídia móvel para todos, com o mesmo preço da telefonia fixa atual.

Alguns dos padrões 2 ½ G, como o GPRS e o IS-95b, são capazes de oferecer esse desempenho, mas apenas em condições ideais. Os sistemas de terceira geração devem fornecer velocidades ISDN a todos e não apenas a quem esteja equipado com os terminais mais caros e em localização próxima à estação base.

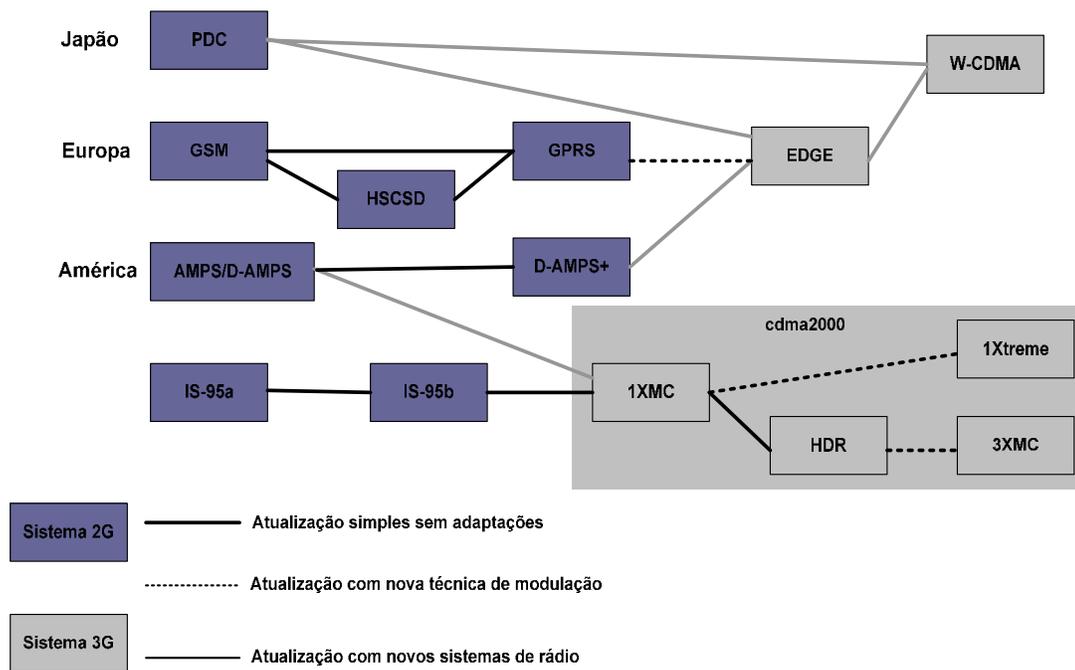


Figura 2.1 Percurso de Atualização da telefonia móvel para Terceira Geração (extraído de [7])

Segundo a categorização dada por [7], a quarta geração é definida (ou 4G) como uma rede telefônica que proporcionará a transferência de dados de até 100 Mbps. Essa velocidade

será suficiente para proporcionar a tele-presença, ou seja, a ilusão de se estar diante da pessoa de verdade, o objeto real.

A figura 2.1 fornece uma visão geral da árvore evolutiva dos padrões mais populares.

Este trabalho estuda a telefonia móvel de segunda geração, possibilitando sua expansão para a terceira. O estudo de caso é sobre um sistema de transmissão totalmente digital; nele, serviços de dados são inseridos evitando causar forte impacto na qualidade do serviço de voz.

As seções seguintes apresentam uma visão geral de alguns destes padrões, a saber: AMPS, DAMPS, IS-95, GSM, dentre outros.

2.2 - AMPS E DAMPS (IS-136)

Desenvolvido na década de 70 pela *Bell Laboratories*, o AMPS (do inglês, *Advanced/American Mobile Phone System*) pertencia à primeira geração de celulares. Ele é considerado por [8] como o primeiro sistema celular implantado nos Estados Unidos.

Baseado na tecnologia de Acesso Múltiplo por Divisão de Frequência (FDMA), ele se baseia em um esquema de reuso de frequências. Os canais de voz são atribuídos às frequências de rádio usando o FDMA. Um total de 50 MHz é alocado na banda de 824-849 MHz e 869-894 MHz. Esse espectro é dividido em 832 canais *full-duplex*, totalizando 1664 frequências discretas. Ou seja, 832 para *uplinks* e outros 832 para *downlinks*.

No AMPS, as células de transmissão são agrupadas em *clusters*. Cada célula interage umas com as outras dentro de um mesmo *cluster*, podendo usar diferentes frequências. Essas frequências são reusadas em diferentes *clusters*. Com base no planejamento de reuso de frequências, existem aproximadamente 50 canais por células.

Em [8], o AMPS é comparado com outros serviços digitais disponíveis nos Estados Unidos. Segundo ele, o AMPS oferece maior cobertura geográfica por meio de uma carga menor de serviço.

A substituição do AMPS pelas redes digitais está permitindo atender uma densidade maior de usuários a um custo menor. Isso permite a liberação de uma largura de banda para utilização de pacotes de dados, tal como, o serviço de CDPD (Comutação de Pacotes Digitais Celulares, do inglês – *Cellular Digital Packet Data*), capaz de efetuar a transmissão de pacotes de dados em um único canal AMPS.

Sucessor do IS-54 [9] após a revisão C, o DAMPS (IS-136) (do inglês, *Digital Advanced/American Mobile Phone System*) [10] suporta interfaces aéreas TDMA (do inglês, *Time Division Multiple Access*) similares ao GSM. Ele opera no mesmo espectro de frequência (30 kHz) usado pelos sistemas AMPS existentes, o que garante sua compatibilidade com as redes em uso de toda a América do Norte. Como o AMPS, o IS-136 usa o padrão IS-41 para gerenciar a mobilidade.

Pelo TDMA, cada portadora IS-136 suporta em sua frequência 3 canais de voz, com taxa de velocidade de 7,95 kbps. Assim, o DAMPS tem aproximadamente 3 vezes a capacidade dos sistemas AMPS existentes.

O IS-95 também está definido para o espectro de 1850 a 1990 MHz. Como o GSM, que será visto adiante, o IS-136 inclui a transmissão de mensagens curtas, mensagens de *broadcast*, endereçamento de grupo, estrutura de célula hierárquica e canais de paginação de *slots* para suportar o modo de descanso no aparelho móvel, conservando a potência da bateria.

Enquanto o IS-136 surgia, o IS-95 ganhava mercado por ser baseado na tecnologia CDMA (Acesso Múltiplo por Divisão de Código).

2.3 - IS-95

Este padrão IS-95, popularmente chamado PDC (do inglês, *Personal Digital Cellular*). Também chamado CDMA de Banda Estreita (ou N-CDMA), foi desenvolvido no Japão, pela *Qualcomm*, e é baseado na tecnologia CDMA.

O CDMA permite que muitos usuários compartilhem o mesmo canal de transmissão de informações. Sinais de diferentes usuários são propagados com códigos que os diferenciam entre si.

O IS-95 usa canais com larguras de banda de 1,25 MHz. Esta largura de banda é relativamente estreita para um sistema CDMA. Assim, a migração do serviço analógico para digital fica mais difícil que em sistemas AMPS para DAMPS. A terceira geração dos sistemas CDMA resolve esse problema, a largura de banda é estendida para 5 MHz.

Segundo [8], o sistema coreano IS-95 começou a operar comercialmente em abril de 1996, com capacidade máxima de 512 transceptores por estação base, ou seja, 320 canais de tráfego por estação base, conectados a 12 centrais de comutação base. Estas centrais são conectadas a uma central de comutação móvel (chamada MX) usando 768 linhas de sinal básico E1.

O IS-95, assim como o IS-54 e o IS-136 vistos anteriormente, também usa o padrão IS-41 para o gerenciamento da mobilidade.

O cdmaOne é a marca comercial do CDMA *Development Group*, uma associação de fornecedores, para os padrões CDMA IS-95a e IS-95b atuais. Alguns cientistas consideram o cdmaOne como o sistema mais avançado da geração 2 e 2½ G.

Todos os sistemas anteriormente citados foram desenvolvidos para suportar serviços a taxa de 5 a 15 kbits/s. Com a incorporação do tráfego Internet e o sucesso de serviços de dados como SMS (Serviço de Mensagem Curta) na Europa e o i-Mode no Japão, o crescimento dos serviços de dados móveis ficou previsível. Nessa conjuntura, as Redes de Dados de Terceira Geração (3G), já padronizadas nos anos 90, estão sendo introduzidas brevemente. Essas redes 3G focam o uso de diversos recursos multimídia sobre uma alta largura de banda [7].

Embora haja uma variedade enorme de sistemas diferentes, as discussões têm concentrado mais na eficiência do espectro por representar a capacidade que o sistema pode extrair a partir de suas frequências alocadas. Infelizmente, esta eficiência depende de um grande número de variáveis cujo cálculo é tremendamente difícil de realizar com precisão [7].

Cada operadora possui uma fatia reduzida do espectro de frequência. Assim, é fundamental que o espectro seja utilizado da forma mais eficiente possível. Até certo limite, pode-se aumentar a capacidade do sistema por meio da instalação de mais estações rádio-base de menor potencial, o que já está implícito no super faturamento dos leilões de frequência.

Embora o IS-95 seja baseado na tecnologia CDMA, o GSM e a maioria dos sistemas PCS são baseados no padrão TDMA.

2.4 GSM

O *Groupe Spéciale Mobile* foi o comitê europeu responsável por definir o *Global System for Mobile Communications* (GSM). Uma das metas desse comitê era formular uma tecnologia que compatibilizasse serviços celulares entre os vários países europeus.

Tecnologia usada por mais da metade dos telefones móveis do mundo, o GSM é o sistema que oferece maior cobertura utilizando voz de qualidade relativamente alta, tendo sido padronizado de forma a facilitar sua atualização para velocidades superiores.

O padrão GSM europeu é um sistema celular digital que combina o FDMA e o TDMA, utilizando 8 *time-slots* sobre uma faixa de frequências de 200 kHz. Dessa forma, na estação base GSM, cada par de radio transmissão (transmissor-receptor) suporta 8 canais de voz, transmitindo e recebendo em cerca de 1/8 do tempo do canal, ou seja, 33,9 kbps. Parte dessa capacidade é deixada vazia para compensar o retardo da propagação.

Na Europa e em alguns outros países, o GSM opera à frequência de 900 MHz (GSM900) e 1800 MHz (GSM1800). Na América, por questões de disponibilidade do espectro, o mesmo opera a 1900 MHz (GSM1900).

Atualmente, o sistema opera em quatro bandas de frequência diferentes:

- 900 MHz: projeto inicial para reutilização do espectro destinado às redes TACS (do inglês, *Total Access Communications Systems*) analógicas da Europa;
- 1800 MHz: licenciada na Europa especialmente para o GSM;

- 1900 MHz: usado na América para várias redes digitais diferentes;
- 450 MHz: variante mais recente que substitui redes analógicas baseadas no sistema escandinavo NMT (do inglês, *Network Mobile Telephony*).

Como as transmissões são unidirecionais, o GSM utiliza canais emparelhados distintos para enviar e receber. Isso possibilita velocidades de dados da ordem de 270,8 kbps em cada canal de 200 kHz. Entretanto, a velocidade obtida é uma reduzida fração dos 270 kbps máximos.

No GSM, a distância máxima entre o terminal móvel e a estação base é de até 30 km. Isso faz com que cada *time slot* demore 577 micro-segundos para ida e volta. Cada *slot* dispõe de 156,25 bits, distribuídos na estrutura de *frames* e *multiframes* conforme mostrado na figura 2.2 e composto dos seguintes campos:

- cabeçalho e rodapé: espaços vazios no início e no final do slot separam o slot de seus vizinhos, impedindo o retardo de propagação a distância de até 35 km da estação base. Acima desse limite os aparelhos GSM não são utilizados.
- Sequência de Treinamento: é usado para ajudar receptores bloqueados no slot, mas não conduz informações úteis. Também serve como cabeçalho e rodapé quando o slot é submetido a novas divisões.
- 2 bits suprimidos: é usado para conduzir dados ou controlar informações sobre o próprio GSM.

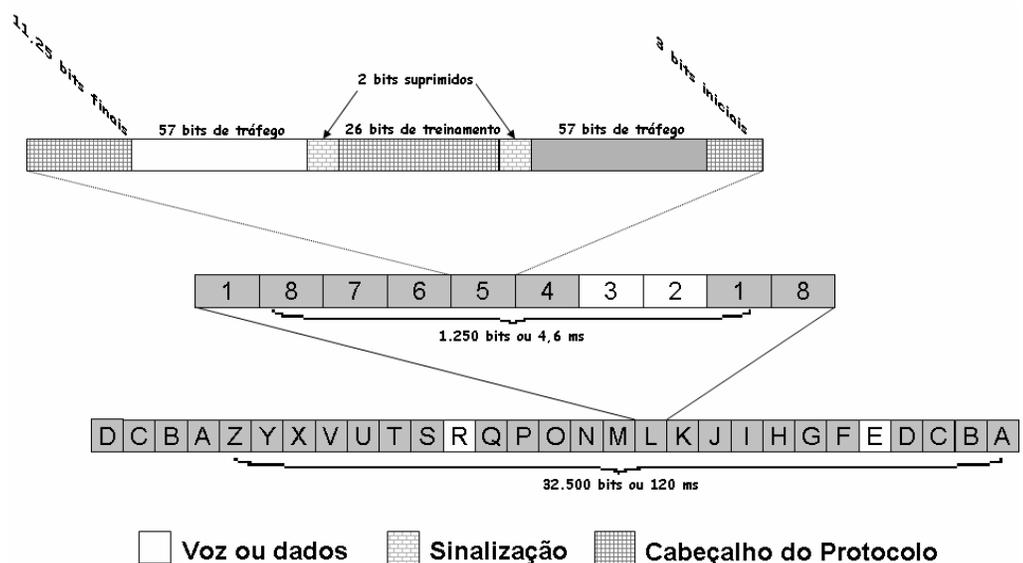


Figura 2.2 Estrutura dos *Slots*, dos *frames* e dos *multiframes* no GSM [7]

- bits de tráfego: informações de controle e correção de erros, além de fala e dados. Em um sistema GSM, essa carga é de 24,7 kbps.

O trabalho desta tese utiliza o conceito de *slots* por divisão de tempo, na simulação do tráfego telefônico móvel. Aqui, cada *slot* é visto como uma unidade atômica, não sendo necessário a manipulação de qualquer informação sobre as partes que o compõe, ou seja, não há referências a cabeçalho, rodapés ou mesmo às informações contidas nos bits de dados. O tráfego de cada fatia de tempo é gerado randomicamente em tempo de execução, de acordo com os percentuais das quantidades de cada tipo de tráfego previamente definido.

Como em outros sistemas celulares, no GSM os dados são codificados em ondas utilizando a modulação em fase conhecida como GMSK (do inglês, *Gaussian Minimum Shift Keying*).

No GSM, o terminal móvel controla a potência de saída de sua taxa de frequência, mantendo a interferência em níveis baixos.

Quanto ao protocolo de gerenciamento de *roaming* do GSM, este possui funcionalidades similares ao IS-41.

O GSM também inclui características digitais como o envio de mensagens curtas ponto a ponto, endereçamento de grupo, espera de chamadas, serviços *multicast*, e outros.

A interface aérea GSM tem evoluído em direção ao EDGE (do inglês, *Enhanced Data Rates for GSM Evolution*), com taxa de dados variável e novas adaptações nos enlaces. Nesse processo evolutivo, o protocolo HSCSD é anterior a este último.

2.4.1 - HSCSD

Em 1997, o ETSI padronizou o HSCSD (do inglês, *High Speed Circuit Switched Data*) como uma pequena atualização ao GSM para que cada usuário utilizasse mais de um *time slot* na operação de multiplexação. Ele é um protocolo de comutação por circuito para a transferência de grandes arquivos e de aplicações multimídia [10].

Todas as redes ou terminais HSCSD utilizam um codec de dados aprimorado. O padrão reúne até 4 dos canais de 14,4 kbps para um máximo de 57,6 kbps. Etapas intermediárias de 28,8 e 43,2 kbps são possíveis e também são mais comuns. As transferências também podem ser assimétricas, ou seja, 3 *slots* para o *downlink* – Transmissão BS/MS (do inglês, *Base Station/Mobile Station*) - e apenas 1 para o *uplink* - Transmissão MS/ERB. Essa assimetria decorre da necessidade de se tratar o problema de superaquecimento desses terminais móveis.

Seria possível agregar todos os 8 canais de um *time-slot* em uma única chamada, mas isso não ocorre por dificultar o projeto dos aparelhos de telefone. Outros problemas decorrem da utilização de canais adicionais que incluem a capacidade e o consumo de energia da rede.

Por conta do uso de canais adicionais em uma chamada regular, o HSCSD precisa de 4 até 8 vezes mais largura de banda que o GSM convencional. A maioria dos usuários não deseja pagar mais por chamadas regulares, especialmente com a idéia de que os dados deveriam ser gratuitos. Por isso, muitas operadoras decidiram não implementar o HSCSD e aguardar outro sistema capaz de utilizar a largura de banda de maneira mais eficiente.

Os terminais HSCSD podem transmitir por aproximadamente 4 vezes a velocidade de um telefone GSM comum, com os correspondentes aumentos de consumo de bateria e emissão de radiação [7].

Infelizmente, o espectro emparelhado não permite re-alocar a largura de banda da transmissão não utilizada da emissão (do inglês, *upload*) para a recepção (do inglês, *download*). Assim, alguns *slots* são deixados vazios.

2.4.2 - GPRS

O esquema de internet sem fio mais popular entre as operadoras é o GPRS (do inglês, *General Packet Radio Service*). Ele fornece aos usuários uma conexão, à Internet, permanente e de alta capacidade. Embora a primeira geração de terminais GPRS oferecesse transferências de dados inferiores às tecnologias HSCSD mais primitivas.

Seu principal benefício é o uso da comutação de pacotes por ser mais eficiente que o padrão de circuito comutado. No GSM e no HSCSD, cada usuário mantém aberto um circuito de 9,6 kbps ou mais enquanto estiver on-line. Isso é um desperdício tanto para o cliente, que passa a maior parte do tempo lendo páginas da WEB e tem que pagar uma tarifa por essa conexão ociosa, quanto para a operadora que compromete uma parte do espectro que poderia ser utilizado em outro local.

No GPRS, um único *time-slot* de 14,4 kbps ou mais, pode ser compartilhado por centenas de pessoas, desde que o acesso não seja simultâneo. Cada usuário tem uma conexão contínua com velocidade de dados de 0,1 kbps ou menos. Essa velocidade sobe quando o usuário recebe e-mail ou acessa um *hyperlink*.

A especificação completa utiliza terminais capazes de usar até 8 *time-slots* simultaneamente, disponibilizando a cada usuário uma taxa maior que o HSCSD. Entretanto o GPRS também enfrenta os mesmos problemas de superaquecimento do HSCSD. Enquanto isso não for solucionado, o GPRS terá o mesmo limite de velocidade que seu antecessor. Com o uso de múltiplos *time-slots* de um mesmo frame TDMA, a largura de banda seria dinamicamente variável e o retardo de transferência seria reduzido, deixando a taxa de dados em aproximadamente 170 kb/s que só pode ser alcançada teoricamente.

Como o HSCSD, muitos sistemas GPRS são assimétricos [11]. A estação base monitora o tráfego enviado aos terminais móveis (*link* direto ou *downlink*) e organiza-o de forma a otimizar o uso da largura de banda. O mesmo não acontece com o tráfego advindo dos terminais móveis, uma vez que um telefone móvel não tem acesso a outras transmissões; havendo assim, maior capacidade disponível para o *link* direto que para o *link* reverso (ou *uplink*).

Por enquanto, o GPRS precisa co-existir com o GSM e talvez com o HSCSD. Por isso, os terminais móveis admitem a comutação de pacotes e a comutação de circuitos caracterizada nos dois sistemas anteriores.

O ideal seria que as redes GPRS existissem isoladamente, conduzindo voz nos pacotes, utilizando um codec de velocidade variável para que os dados pudessem ser transmitidos durante os intervalos da conversação. O GPRS precisa de uma rede de *backbone* inteira-

mente nova substituindo o sistema de telefonia existente por outro semelhante à Internet. Ele foi projetado para o GSM, mas pode ser adaptado facilmente a qualquer rede TDMA. As redes 3G podem vir a se utilizar dessa tecnologia por serem totalmente comutadas por pacotes.

O EDGE, tecnologia de terceira geração, utiliza modulação de espectro altamente eficiente para transmitir a taxas superiores à tecnologia GSM. O EDGE requer atualização dos transceptores das estações base para o suporte a transmissão de dados de alta velocidade em células menores.

2.4.3 - Infra-Estrutura Telefônica

A maioria dos usuários entra em contato apenas com seus próprios telefones móveis e, ocasionalmente, com as estações base cada vez mais visíveis. Muito embora, a rede de telefonia possua um conjunto de outras pequenas partes que a compõe. Desde servidores WEB, armazenando páginas formatadas para o acesso sem fio, *switches* e *links* auxiliando na conexão da rede com o mundo exterior, até recursos mais avançados para rastrear a localização do usuário e gerenciar o espectro de forma eficiente.

A Infra-Estrutura Telefônica tem ficado mais complicada por causa do crescimento dos dados móveis. Quase sempre a configuração completa, com todos os recursos disponíveis, é muito mais complexa do que a das redes de telefonia tradicionais.

Hoje os novos sistemas de terceira geração foram projetados para dados e são baseados na comutação de pacotes de alta capacidade. Contudo, usuários mais antigos (com aparelhos móveis mais antigos) continuam utilizando as redes originais de voz onde se trafegavam sinais puramente analógicos.

A figura 2.3 mostra o GSM sem atualização para GPRS, ilustrando o percurso que a voz segue em uma rede entre um telefone celular e um telefone fixo, já a figura 2.4 apresenta o percurso que os dados fazem nessa mesma rede telefônica.

De acordo com a figura 2.3, o sinal de voz é capturado pela ERB e encaminhado ao telefone fixo localizado em uma outra rede telefônica. O sinal é conduzido ao controlador da es-

tação base que o encaminhe para uma central de comutação que se encarrega de rastrear a localização dos usuários, realizando o roteamento do sinal ao destino correto. Este sinal é comutado dentro da própria operadora até que uma rota seja encontrada para o telefone fixo. Uma vez identificado que o destino pertence a outra rede telefônica, o *gateway* GMSC (do inglês, *Gateway Mobile Switching Center*) se encarrega de ligar a rede móvel a outras redes telefônicas, cuidando também de questões de *roaming*. O retorno da transmissão segue a mesma estrutura global.



Figura 2.3 Percurso da Voz em uma Rede GSM (modificado - [7])

A figura 2.4 apresenta o protocolo WAP para também exemplificar a transmissão de dados. O WAP é um padrão aberto para a implementação de um serviço de informações textuais permitindo a mobilidade de navegação sobre a WEB. Esse padrão foi definido pelo *WAP Forum* – um consórcio de empresas formado originalmente pela Nokia, Ericsson e Motorola, mas agora com a participação de outros membros. Esse protocolo ganhou muita força no mercado; primeiro por ser uma solução não proprietária; segundo, por não ser apenas uma linguagem de marcação para a WEB sem fio, mas toda uma nova pilha de protocolos, projetada para superar problemas de latência e flutuação.

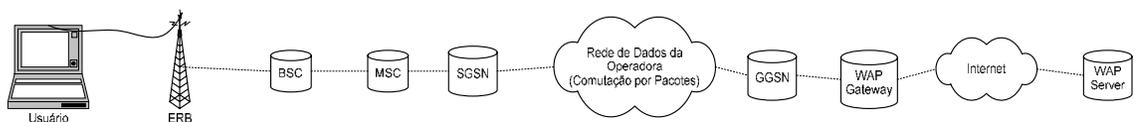


Figura 2.4 Percurso dos dados WAP em uma rede GSM/GPRS (modificado – [7])

Nesse contexto, a figura 2.4 mostra o percurso entre um terminal WAP e um servidor WAP. Nela, dados são transmitidos por meio de canais na ERB. Em seguida, o pacote (*frame* ou *multi-frame*) é enviado do controlador para a rede de comutação – MSC (do inglês, *Mobile Switching Center*) - e em seguida para o SGSN (do inglês, *Serving GPRS Support Node*) que o encaminha ao GGSN capaz de encontrar a informação solicitada junto a uma outra rede. O *WAP Gateway* trata as páginas WAP para a transmissão sobre a rede móvel. O servidor WAP armazena páginas da WEB e as envia sob demanda aos surfistas de unidades móveis.

Em síntese, uma rede telefônica GSM/GPRS completa, compõe-se das seguintes partes:

- *Link Aéreo*

- o MSU (do inglês, *Mobile Station Unit*);
- o BTS (do inglês, *Base Transceiver Station*).

- Infra-Estrutura de Voz

- o BSC (do inglês, *Base Station Controller*);
- o MSC;
- o TSC (do inglês, *Trunking Switching Center*);
- o GMSC.

- Infra-Estrutura de Dados

- o PCU (do inglês, *Packet Control Unit*);
- o SGSN;
- o Backbone do GPRS (do inglês, *General Packet Radio System*);
- o GGSN (do inglês, *Gateway GPRS Support Node*).

2.4.3.1 Link Aéreo

O *Link Aéreo* refere-se ao enlace local, a conexão final entre o cliente e a rede. São necessários dois rádios: um carregado pelo usuário (MSU) e o outro conectado permanentemente à rede da operadora (BTS).

2.4.3.2 MSU

É o terminal móvel transportado pelo usuário; seja ele um telefone móvel (como nos sistemas de 1ª e 2ª geração) ou quaisquer outros dispositivos de transmissão de dados (desde *paggers* simples até dispositivos avançados como o Palm VII).

2.4.3.3 BTS

O BTS (do inglês, *Base Transceiver Station*), também chamada de estação base ou estação rádio-base, é o principal rádio no centro de cada célula. Compostas por várias antenas; normalmente montadas dentro de caixas retangulares medindo 1x3 metros.

Os BTSs são fontes mais poderosas de microondas e não existe dúvida de que possam ser perigosas. Entretanto, elas são indispensáveis para captura e recepção dos sinais de onda de nossos celulares.

2.4.4. Infra-Estrutura de Voz

A rede celular de voz é, tradicionalmente, a PSTN fixa, acrescida de algumas bases de dados extras para cuidar da mobilidade.

Toda a comutação é digital, mesmo nos sistemas G1 com seus *links* aéreos analógicos. Todos os telefones móveis utilizam discagem de tom; embora ainda existam telefones fixos que utilizam discagem de pulso.

Quando as redes de dados estiverem consolidadas muitas chamadas de voz ainda serão enviadas pela infra-estrutura de circuito comutado existente. Quando as atualizações estiverem concluídas, o circuito comutado será a única opção para os dados. É o que acontece com o HSCSD que aumenta a velocidade de transferência do GSM vinculando circuitos em conjunto.

2.4.4.1 - BSC

O BSC simplesmente converte os sinais da estação base em um formato de linha terrestre.

As redes analógicas precisam que o BSC esteja na própria BTS para que o *link* analógico seja convertido tão logo recebido. As redes digitais, reduzem os custos com a instalação de BSCs permitindo que vários BTS utilizem o mesmo BSC.

Células interligadas a um único BSC correspondem a clusters que são agrupamentos de 3 a 21 células utilizando, em conjunto, todo o espectro de rádio disponível da operadora.

2.4.4.2 MSC

O MSC é responsável pelo rastreamento de usuários e pelo envio de chamadas quando solicitadas. Normalmente, um MSC é responsável por vários clusters de células.

Todos os clientes de uma rede móvel são registrados em uma determinada MSC, ou seja, a MSC local do cliente. Esta MSC local sempre é contatada quando ocorre tentativa de comutação de chamadas para o usuário.

Cada MSC está associada a várias bases de dados utilizadas para o armazenamento de localizações de usuários e informações de bilhetagem. Algumas utilizam base de dados local que podem estar em um computador da própria MSC ou na rede fixa, e também mantém uma cópia de segurança.

As bases de dados são mostradas na figura 2.5 e incluem:

- HLR (do inglês, *Home Location Register*) – contém a localização dos aparelhos registrados na própria MSC.
- VLR (do inglês, *Visitors Location Register*) – armazena temporariamente a localização geográfica de todos os aparelhos de telefone encontrados e pertencentes a outra MSC.
- EIR (do inglês, *Equipment Identity Register*) – é um registrador que identifica o equipamento que lista números únicos, identificando cada aparelho e também permite a bilhetagem das ligações.
- AuC (do inglês, *Authentication Center*) só existe em redes digitais. Autentica o assinante móvel, solicitando que o número de identificação seja criptografado durante a transmissão pelo ar.

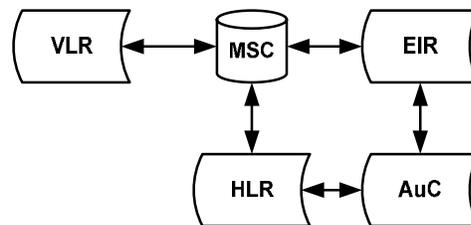


Figura 2.5 Subsistemas da base de dados da MSC [7]

2.4.4.3 TSC

Os TSC são centrais de comutação de troncos que simplesmente agregam as conexões de várias MSCs, combinando-os em cabos de alta capacidade.

Nem todas as operadoras utilizam uma TSC, algumas utilizam apenas MSC cobrindo uma área bastante extensa. Nas operadoras em que os TSCs são empregados, a rede costuma conter poucas dessas centrais ou mesmo apenas uma.

2.4.4.4 - GMSC

O GMSC conecta a rede móvel a PSTN fixa e a outras operadoras com as quais possui acordos de *roaming*.

2.4.5 - Infra-Estrutura de Dados

Os dados móveis utilizam os BTSs e os BSCs; entretanto no lugar dos *switches* utilizados para comutação de voz, eles precisam de roteadores como os da Internet, para o envio de dados em pacotes pequenos.

Nas redes GPRS os dados são transportados livremente durante intervalos na conversação de voz enquanto a rede não estiver ocupada, embora ocorram congestionamentos em períodos de pico.

2.4.5.1 PCU

A parte mais cara das redes GPRS ou D-AMPS é a PCU, necessária para adaptar a estação base aos dados em pacotes.

Em alguns casos, uma PCU é necessária em cada estação base; em outros casos, é permitido haver uma única PCU a ser colocada em um BSC ou até em uma MSC.

Muitas redes D-AMPS já dispõem de alguma capacidade de dados em pacotes, utilizada pelos serviços CDPD existentes. Infelizmente, essa capacidade não costuma ser suficiente para as redes D-AMPS+ e, portanto, é necessária uma atualização.

2.4.5.2 SGSN

O SGSN é o equivalente de dados da MSC. Também possui várias bases de dados que autenticam e localizam o tráfego.

Diferentemente da MSC, o nó SGSN não precisa estar conectado diretamente a cada uma das células as quais atende.

Numa rede ideal, o SGSN teria um *link* dedicado a cada PCU de sua área; mas o tráfego também poderia ser transportado de volta sobre a rede de voz. Isso é um processo mais lento e menos eficaz, mas permite que as operadoras configurem e coloquem em funcionamento um sistema GPRS, rapidamente.

Algumas operadoras anunciam uma cobertura GPRS completa, com apenas um SGSN. Neste caso, a maioria do tráfego é conduzida pela rede de voz. Entretanto, o que importa é a densidade da cobertura – quantos SGSNs a operadora possui para uma determinada área.

Muitas operadoras iniciam com poucos SGSNs, adicionando outros à medida que a rede fica congestionada.

Diferentemente das PCUs, existem padrões claramente bem definidos para a conexão de SGSNs as redes celulares, permitindo que as operadoras misturem e combinem fornecedores. Isso é importante, pois nem todos os SGSNs proporcionam o mesmo desempenho; existem várias opções de desempenho, como o gerenciamento de filas e a qualidade do serviço. O tipo de SGSN implementado depende do tipo de cliente que as operadoras têm como alvo.

2.4.5.3 Backbone do GPRS

O *backbone* do GPRS é formado por uma rede de SGSNs e outros dispositivos GPRS. O SGSN funciona como um roteador e não como um *switch*. Apesar da MSC estabelecer a conexão, que permanece por toda a duração da chamada, a MSC encaminha cada pacote de dados separadamente.

2.4.5.4 GGSN

O GGSN é usado para permitir o acesso a redes de dados externas. Ele converte pacotes de dados GTP (do inglês, *GPRS Tunnel Protocol*) em TCP/IP (do inglês, Transport Control Protocol/Internet Protocol), e vice-versa. Dessa forma, uma rede telefônica precisa apenas de um GGSN. Dois ou mais desses dispositivos seriam úteis no Gerenciamento de Falhas, permitindo a redundância desses dispositivos.

As máquinas baseadas em IP o vêem apenas como um roteador na Internet, sem saber que seus usuários pertencem a redes móveis.

2.4.6 - Características dos Serviços GPRS

2.4.6.1 Aplicações para GPRS

O GPRS suporta uma larga variedade de aplicações, desde transmissões com um grande volume de dados até transmissões infreqüentes. Por trás de tudo isso, a aplicação mais importante para GPRS é a Internet sem Fio. Dessa forma, os computadores móveis devem suportar quaisquer aplicações baseadas em Internet convencional, como transferência de arquivos, e-mail, chat ou navegação web, além de vídeo sob demanda, a serem transmitidos para o GPRS.

Outras aplicações igualmente importantes são as aplicações de informação de transporte como; informações de controle de rotas e gerenciamento de frotas. Temos também as aplicações financeiras, como saldo eletrônico e transferência entre fundos de investimentos em um canal GSM/GPRS.

Também é possível mandar mensagens SMS sobre o GPRS pela conexão do GSM SMS-MSC para o SGSN.

2.4.6.2 Qualidade de Serviço (QoS)

Com a introdução do GPRS nas redes móveis GSM, os padrões de qualidade dessas redes tiveram que ser repensados sobre o foco das aplicações e serviços introduzidos. O suporte

a diferentes classes, para cada sessão particular, teria que levar em conta padrões de serviços presentes em todas essas classes. Daí levar-se em conta questões como: precedência, confiabilidade, retardo e vazão.

Por definição, *precedência* é a prioridade que um serviço tem em relação a outro. São três os níveis de precedência: alta, média e baixa.

A *confiabilidade* indica as características que a transmissão de uma aplicação deve seguir. Uma espécie de contrato que reflita as necessidades de uma classe de aplicação. Em [12] são apresentados três classes de confiabilidade garantindo para cada uma delas os valores máximos de probabilidade de perda de informação (bits), duplicação, não seqüenciamento dos bits e de pacotes corrompidos.

Da mesma forma, [12] define valores para os retardos máximos de cada pacote e de todos os pacotes trafegados. O retardo é definido como o tempo de transferência fim-a-fim entre dois sistemas móveis ou entre um sistema móvel e um PDN (do inglês, *Packet Data Network*) externo; isto é, inclui os retardos para solicitação e atribuição dos recursos de rádio, tempo de transmissão das mensagens e o retardo de transição nas redes GPRS. Conforme explicado na mesma tese, o percentual de retardo de confiança em 95% é a garantia máxima para todas as transmissões.

A vazão especifica a taxa de pedido de dados do usuário. Ela define os dois parâmetros de negociação: a taxa de pico (ou máxima) e a taxa média de bits. Ela pode ser negociada, enquanto que os demais atributos são classificados por diferentes classes de qualidade de serviço.

Os perfis de QoS também podem ser negociados entre o usuário móvel e a rede, para cada sessão, dependendo da demanda de QoS e dos recursos disponíveis. A tarifação desses serviços é baseada numa taxa fixa de assinatura e uma taxa variável conforme o volume de tráfego, o tipo de serviço requerido e os perfis de QoS escolhidos.

O desempenho de uma rede telefônica PCS é significativamente afetado pela forma como a rede gerencia o movimento dos usuários móveis. A seção seguinte faz um breve comentário sobre essa mobilidade nas redes GSM/GPRS.

2.4.7 - Gerenciamento de Mobilidade, de Sessão e de Roteamento

2.4.7.1 Gerenciamento de Mobilidade

O protocolo de gerenciamento de mobilidade no GPRS é chamado GTP; ele é baseado no protocolo TCP/IP utilizado na Internet.

A comunicação móvel é feita pela implementação de funções que traçam a localização do usuário, e transferem as informações dele e da chamada do sistema de origem do assinante, para o sistema temporariamente visitado. Assim, o gerenciamento de mobilidade é composto por:

- portabilidade do serviço, assim como as funcionalidades de *handover* e de *roaming*;
- implantação de provedores de serviços que possibilitem o emprego de sistemas de sinalização *out-of-band*.

Um *handover* ocorre no sistema móvel toda vez que o aparelho move-se de uma estação base para outra, durante o curso de uma conversação. Isso traz sérias preocupações quanto ao gerenciamento, como detecção e atribuição de canais e transferência dos enlaces de rádio. São vários os tipos de interconexão de *handovers*, incluindo *handover-forward*, *handover-backward*, *handover-to-third* e *handover path minimization*. Entretanto, não existem *handovers* entre células nos canais alocados para dados, apenas voz.

O *roaming* ocorre quando um MS move-se de um sistema PCS para outro; por exemplo, do sistema na cidade de Brasília para outro em Fortaleza. Nesse caso, o sistema original deve ser informado sobre a nova localização do MS. Caso contrário, será impossível distribuir os serviços para o usuário.

Durante uma chamada, a estação móvel encontra-se em um dentre os três estados apresentados na figura 2.6: aguardando, ocioso e executando. No estado “ocioso” (do inglês, *idle*), a estação móvel não é contatada. Quando é estabelecida uma comunicação, a estação móvel vai para o estado de “executando” (do inglês, *ready*). Após o envio de um pacote a estação passa então para o estado de aguardando (do inglês, *standby state*) saindo apenas

quando sua solicitação for atendida, onde voltará para o estado executando. Caso expire o tempo de espera, a estação voltará para o estado ocioso. A figura 2.6 apresenta essas mudanças de estado.

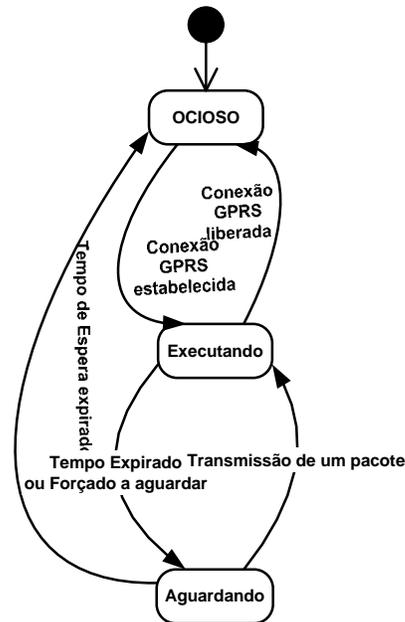


Figura 2.6 Diagrama de Estados para uma Estação Móvel [12]

Antes que um aparelho móvel possa utilizar o serviço GPRS, uma conexão GPRS (ou GPRS *attach*) deve ser estabelecida com a ERB. O aparelho deve se registrar no SGSN; assim, a rede checa sua autorização para que só a partir daí o perfil do usuário seja copiado da HLR/GR para o SGSN. Neste momento, é atribuído ao assinante um identificador temporário de usuário.

Para trocar pacotes de dados com redes externas, o aparelho móvel deve utilizar um identificador de endereço apropriado para a rede em questão. Dessa forma, um ou mais endereços devem ser utilizados temporariamente para o real sucesso da transmissão.

O HLR/GR mantém esses contextos PDP para cada estação móvel como parte dos dados do assinante.

2.4.8 - Analisando o GPRS para Acesso à Internet

Em [13] é implementado um simulador onde mostra o desempenho do GPRS para acesso às aplicações TCP/IP, tipicamente pertencentes à Internet.

Os resultados que ele obteve, demonstram ter o GPRS um bom desempenho para o acesso aos serviços da Internet. Visto que, uma das vantagens da comutação por pacotes é a utilização mais eficiente dos recursos de rádio escassos no enlace aéreo. Esta característica presente nas redes GPRS, por si só, já traz grandes benefícios às redes GSM originais e é estudada em [13] sobre o enfoque dos recursos de rádio que são disponibilizados para o tráfego de dados.

Vários parâmetros foram usados na sua simulação, a saber:

- número de usuários WWW de 1 a 40;
- tamanho máximo do segmento TCP de 536 bytes;
- média da taxa de perda de pacotes Internet de 2%;
- número de PDCH (do inglês, *Packet Data Channel*) de 1 a 8;
- apenas 1 *timeslot* para o *uplink*, por usuário;
- 1, 2 ou 4 *timeslots* de *downlink* por usuário;
- média de C/I de 12 dB;
- Variância C/I de 5 dB;
- Blocos PRACH (do inglês, *Packet Random Access Channel*) de 0 e 6;
- codificação do canal: CS 2;
- duração da simulação de 30 minutos.

O número de PDCHs especifica quantos dos 8 *timeslots* de um frame TDMA são reservados para tráfego GPRS pelo operador da rede para uma frequência GSM.

Os resultados observados por ele, geraram vários gráficos comparando o *throughput* médio de *downlink* por número de usuários WWW. Contudo nenhuma comparação é feita com os demais canais que são disponibilizados para transmissão de voz.

Seus resultados demonstram, entre outras coisas, que o GPRS suporta um grande número de conexões paralelas. Essa informação é útil pois, no estudo apresentado no capítulo 6, várias conexões paralelas de GPRS são testadas a fim de se identificar a máxima quantidade

de ideal de recursos alocados para a carga a que a rede está submetida sem degradar consideravelmente a transmissão de voz.

O trabalho apresentado nesta tese foca a tecnologia GPRS/GSM, principalmente, no dinamismo da alocação de canais; visando uma maior garantia na recepção das chamadas de voz; por essas representarem informações prioritárias nas redes 2G e 2.5G.

3 - METODOLOGIA PARA ALOCAÇÃO DE RECURSOS DE RÁDIO EM REDES TELEFÔNICAS

A análise da modelagem do tráfego multimídia e de Internet é muito mais complexa que para as redes comutadas por circuito. Embora já existam boas soluções para a multiplexação de pacotes em redes ATM (do inglês, *Asynchronous Transfer Mode*) [14, 15], soluções específicas para a modelagem das interfaces de rádio ainda são poucas e não adotadas pelas empresas telefônicas.

Modelos analíticos e algoritmos para as interfaces aéreas comutadas por pacotes ainda estão em desenvolvimento; no entanto, estudos demonstram que todos os detalhes da pilha de protocolos das redes telefônicas móveis, das redes telefônicas fixas e dos protocolos de Internet (incluindo o TCP), não podem ser tratados, de forma precisa, sobre a abordagem de um modelo analítico.

A mensuração do desempenho do tráfego nas redes móveis existentes só é possível se configurado para um cenário com uma carga de tráfego muito bem definida. A análise do comportamento dos diferentes protocolos em uma infra-estrutura de rádio existente não é possível com os métodos tradicionais vigentes [106].

Dessa forma, esse trabalho apresenta uma simulação baseada na implementação de alguns cenários, aglomerando o comportamento do GSM com os protocolos Internet e outros geradores de tráfego estocásticos que são modelados conforme apresentados nas interfaces de rádio. Como método alternativo às análises utilizadas, a simulação faz uso de técnicas de lógica *fuzzy* e redes neurais artificiais como ferramenta heurística para avaliação do tráfego.

Em particular, este capítulo relata os princípios do gerenciamento de recursos de rádio para as redes móveis, descrevendo técnicas para a alocação de recursos na sobreposição de redes de transmissão de voz e de dados.

3.1 - ALOCAÇÃO DE RECURSOS DE RÁDIO EM REDES TELEFÔNICAS MÓVEIS

Por mais de duas décadas se estudam a integração de serviços em redes, em geral. Nas redes sem fio, os serviços de voz e dados competem pelos mesmos recursos sendo importante conhecer esses serviços para se realizar estudos em dimensionamento de rede.

Os diferentes modos de transferência de informação em redes celulares são:

1. Comutação por Circuito – neste modo um canal é dedicado à transmissão de informação (voz ou dados) durante todo o estabelecimento da comunicação. Este modo de transmissão é ideal para aplicações em tempo real onde o fluxo de tráfego é contínuo (ex. GSM, AMPS);
2. Comutação por Pacotes – neste modo a largura de banda é alocada apenas durante períodos ativos de conversação, ou seja, quando o usuário tem pacotes a transmitir. Ideal para aplicações com tráfego em rajada (ex. Internet, GPRS, CDPD), e;
3. Comutação Híbrida – em que são suportados ambos os modos de comutação anteriormente citados (ex. redes GPRS/GSM, CDPD/AMPS).

Quando uma operadora decide oferecer serviços de dados, baseados na comutação por pacotes, dentro de uma rede comutada por circuitos, duas alternativas são possíveis:

- Alocação de um novo espectro para a transmissão dos pacotes de dados; ou,
- Compartilhamento do espectro entre os serviços de voz e de dados.

A primeira alternativa é a forma mais simples para aumentar a capacidade da malha telefônica. No entanto, seriam necessários novos investimentos em infra-estrutura (novas células e novos transceptores - TRXs) implicando em gastos elevados. Esses novos transceptores possibilitam o aumento da capacidade de tráfego usando o espectro corrente.

O aumento do número de usuários requer a adição de novas frequências, mas como a operadora, muito provavelmente, não disporá de nova largura de banda, o espaço de frequências existente deverá ser re-planejado.

Devido à escassez de espectro e aos novos investimentos em infra-estrutura, os recursos de rádio já existentes devem ser compartilhados implicando em um novo dimensionamento e

planejamento para a incorporação do serviço de dados. Esse novo planejamento é feito adotando-se alguma das técnicas de alocação de canais mencionadas adiante.

Nas questões de gerenciamento de recursos de rádio em redes de sobreposição de circuitos e de pacotes, a literatura relaciona três diferentes técnicas de alocação de recursos. Todas baseadas nas duas formas de sobreposição das redes de dados sobre a rede de voz. A figura 3.1 apresenta um diagrama de blocos com essas três técnicas.

Uma vez que este trabalho foca, em particular, o uso das tecnologias 2½ G, torna-se fundamental a adoção de comutação híbrida. Nas redes híbridas os recursos são concorrentes para ambos os modos de transmissão. São utilizadas as seguintes técnicas para alocação de canais:

1. **Particionamento Completo (CP)** – a largura de banda disponível é dividida em duas diferentes partes, uma parte apenas para voz e outra apenas para dados. Essa técnica reduz a garantia do serviço de voz se nenhum novo espectro for alocado.
2. **Compartilhamento Completo (CS)** – neste método a largura de banda é dinamicamente compartilhada entre ambos os tipos de usuários, uma vez que a comutação por circuito tem prioridade em relação à comutação por pacote. Portanto, os canais temporariamente não utilizados pela voz são alocados para o tráfego comutado por pacote. Essa técnica não apresenta impacto sobre a qualidade dos serviços de voz.
3. **Compartilhamento Parcial (PS)** – neste método os usuários de dados possuem alguns canais exclusivamente para transmissão de dados, como no CP. O restante é compartilhado com a transmissão de voz, como no CS. Essa técnica impacta tanto na garantia do serviço de dados quanto na qualidade do serviço de voz.

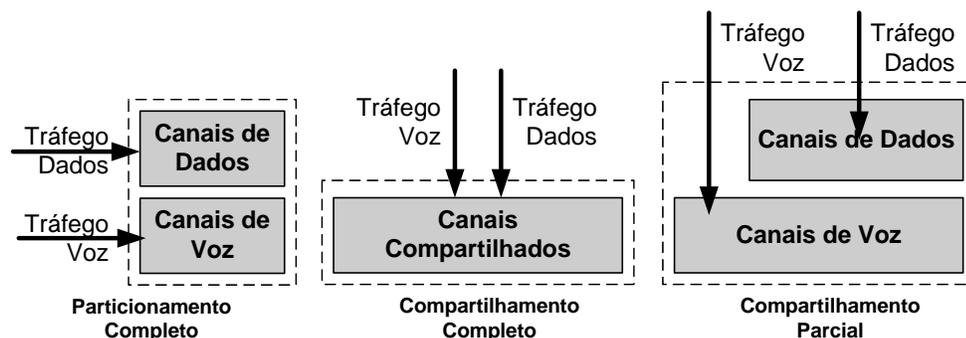


Figura 3.1 Técnicas de Alocação de Rádio em Redes 2½ G

O CP reduz a qualidade dos serviços de circuito prestados pela presença de canais comutados por pacote. Como todos os canais são obtidos do *pool* de canais disponíveis, o número de canais reservados para voz ficará reduzido, resultando em um aumento da probabilidade de chamadas bloqueadas e, portanto, redução da capacidade útil total.

Em contraste ao primeiro, o compartilhamento dinâmico dos canais entre transmissão de dados e voz não impacta na qualidade dos serviços de voz se o número de canais de dados for limitado aos canais com tráfego ocioso e os canais utilizados para comutação de pacotes forem liberados sempre que for necessário tráfego comutado por circuito. Dessa forma, a capacidade para os serviços de voz tende a se manter inalterada. No entanto, isso não é totalmente correto uma vez que o desempenho dos serviços de voz será degradado por causa do aumento da probabilidade de interferência dos sinais e da diminuição da área de serviço de célula devido à transmissão de pacotes de dados [107].

Para garantirmos a qualidade de serviço (QoS) nas redes que usam o método CP e CS é indispensável que se adote algum critério adicional de priorização da voz sobre os serviços de dados, sob pena do mesmo ficar prejudicado.

Nesse sentido, os canais lógicos são agrupados em Canais Mestre (ou MPDCHs) e canais escravos (SPDCHs). Os canais MPDCHs alocam canais PCCCHs (do inglês, *Packet Common Control Channel*) e um PBCCH (do inglês, *Packet Broadcast Control Channel*); ambos carregam informações de sinalização. Os SPDCHs carregam apenas PDTCHs (do inglês, *Packet Data Traffic Channel*), isto é, dados do usuário e sinalização específica de alguma alteração eventual.

Conforme a técnica adotada na alocação dos canais de rádio, os PDCHs podem manter-se inalterados (alocação estática), ou sofrer modificação por conta do aumento da vazão da rede (alocação dinâmica). Em redes como a tratada nesta tese, é ideal o uso de técnicas que dinamizem a alocação desses canais; isto é uma forma de tratar a qualidade do serviço prestado nessas redes, evitando a perda de desempenho.

3.2 - ALOCAÇÃO DINÂMICA DE CANAIS

Por conta das diferentes condições de tráfego é mais interessante o uso de uma adaptação dinâmica de canais, onde o número de PDCHs alocados varie conforme a vazão de dados nesta célula. Os canais liberados podem ser utilizados para transmissão de voz por meio dos procedimentos do serviço de comutação por circuito.

O procedimento para estabelecimento de uma chamada em uma malha de comutação por circuito leva de 3 a 5 segundos; tempo suficiente para a liberação de um PDCH pela rede de pacotes. Essa liberação pode ocorrer de duas formas:

- Liberação Imediata (*Immediate Release*): a transmissão de dados é cancelada imediatamente e o canal é liberado para ser alocado a um novo circuito de transmissão de voz.
- Liberação Tardia (*Delayed Release*): o usuário de dados continua sua transmissão até o envio completo de alguns *frames* ou até o final da transmissão do pacote para só aí ser liberado para o novo circuito de voz.

Essas diferentes políticas de alocação de canais refletem no retardo, na vazão e na utilização do espectro nessas redes. De modo que fica claro, a necessidade da escolha de uma solução que otimize tais recursos, assegurando um bom desempenho em ambos os serviços.

A alocação e a desalocação dos canais de voz estão intimamente ligadas à localização do terminal móvel e à sua mobilidade. Isso serve para distinguir a sinalização do estabelecimento de novas chamadas daquelas já em curso.

3.3 DIFERENÇAS ENTRE NOVAS CHAMADAS E *HANDOVERS*

Dois são os tipos de tráfegos oferecidos pelos serviços de comutação às redes móveis existentes:

- Novas chamadas: são as chamadas iniciadas na própria célula de transmissão; e,
- *Handovers*: são as chamadas trazidas de outras células.

Como a área de cobertura celular é irregular devido à forma de emissão da radiação nas antenas das estações rádio-base, e também devido aos prédios, árvores, montanhas e outras características do terreno, as áreas de cobertura adjacente se sobrepõem. Essa

sobreposição é importante para garantir a mobilidade dos sistemas móveis de uma estação base para outra, durante o curso de uma conversação. Quando o terminal móvel encontra-se no limite da área de cobertura de uma célula o tamanho e a qualidade do sinal se degradam. Neste ponto, o sinal da estação base vizinha (a nova estação base) é emitido com maior intensidade para o terminal móvel que o sinal da estação base original. A partir desse momento, a conversação deve passar a ser tratada pela nova estação base, antes que a transmissão com a estação originária seja inutilizada, causando a interrupção da conversa telefônica. Essa interrupção é contabilizada como uma falha na realização de *handover*.

Novas chamadas e *handovers* utilizam os mesmos recursos de rádio. Logo, estando os canais da ERB ocupada, o estabelecimento das novas chamadas ou mesmo a realização de *handover* podem falhar. O não estabelecimento destas novas chamadas denomina-se: chamadas bloqueadas, e a interrupção das já existentes quando da realização de *handover*: falha de *handover*. Se os percentuais de alocações de novas chamadas e do tráfego de *handover* se mantiverem iguais e nenhuma priorização for concedida, a probabilidade de uma chamada ser bloqueada ou de um *handover* falhar será a mesma.

Do ponto de vista do usuário, na falta de recursos é mais interessante que novas chamadas não sejam estabelecidas frente ao cancelamento imediato das chamadas em curso. Com esse aspecto, *handovers* são usualmente tratados com maior prioridade do que a inicialização das novas chamadas.

Essa prioridade passa a ser fundamental para a alocação de recursos em áreas de células relativamente pequenas (micro-células), onde o número de *handovers* é muito maior e, portanto, o impacto desse procedimento será maior nessas células. Isso poderá ser visto no capítulo 6 deste trabalho, onde o cenário de uma micro-célula é configurado para esta prova.

Três questões precisam ser consideradas para o gerenciamento do *handover*:

- Estratégias para Detecção de *handover*;
- Técnicas para Atribuição do canal;
- Transmissão pelo enlace telefônico.

3.4 - ESTRATÉGIAS PARA DETECÇÃO DE *HANDOVER*

Handovers desnecessários devem ser evitados devido ao seu alto custo de execução. A decisão sobre fazer ou não o *handover* ficou mais difícil com a evolução para células menores, onde a capacidade do sistema aumenta e a potência dos terminais móveis é menor.

Devido à dinamicidade do sistema de propagação, o sinal pode desvanecer por pequenos períodos de tempos. Durante esses breves períodos, é desaconselhável a realização de *handovers* porque o sinal pode retornar ao seu estado normal mais rapidamente que a efetivação do *handover*.

Várias estratégias tratam esse tipo de problema, das quais, três são usadas para determinar a qualidade de um canal:

- Indicador de Erro (WEI): métrica que indica se a rajada corrente foi demodulada propriamente na estação móvel;
- Indicador de Intensidade do Sinal Recebido (RSSI): mede a intensidade do sinal recebido. Deve ficar entre 80 a 100dBm;
- Indicador de Qualidade (IQ): está relacionado à interferência do sinal, à taxa de ruído (C/I) e aos efeitos da dispersão. O C/I deve variar entre 5 dB a 25 dB, aproximadamente.

O *handover* depende mais do Indicador de Erro que do Indicador de Tamanho do Sinal Recebido. Em outras palavras, se o primeiro for bom, então o *handover* não deve ser feito. Portanto, o Indicador de Erro deve ser monitorado por um tempo definido, uma vez que o Indicador de Tamanho do Sinal Recebido é um valor instantâneo. Logo, essas duas variáveis devem ser usadas durante a tomada de decisão sobre a realização de *handover*.

Em sistemas TDMA, a velocidade desse processo de mensuração depende da estrutura de *frame* do sistema de rádio.

São três as estratégias para detecção de *handover* em redes PCS:

- *Handover* Controlado pelo Terminal Móvel;
- *Handover* Controlado pela Rede;
- *Handover* Assistido pelo Terminal Móvel.

3.4.1 - *Handover* Controlado pelo Terminal Móvel (MCHO)

Nesta técnica, o terminal móvel monitora continuamente o tamanho do sinal, a qualidade do acesso à estação base transmitida e de todas as estações base candidatas, uma vez que o mesmo é quem controla toda a realização de *handover*. Em outras palavras, o terminal móvel examina, continuamente, a qualidade do enlace de rádio.

Quando certos critérios de medição são alcançados, o processo indica a necessidade de *handover* e busca selecionar um novo canal, conforme visto na figura 3.2. Finalmente, a estação móvel, em conjunto com a rede, executa o *handover*. A qualidade do enlace é obtida através de vários meios. Como parte do processo de demodulação, a estação móvel geralmente obtêm duas informações: RSSI e IQ.

O IQ mede se o canal corrente está disponível para a estação móvel, a cada frame, como resultado do processo de demodulação. Durante cada período de frame TDMA, quando a estação móvel não está transmitindo ou recebendo informação para a chamada corrente, a unidade tem tempo adequado para fazer uma variedade de medições IQ e RSSI para cada antena. O indicador WEI de *downlink* também está disponível para o terminal móvel. A estação base também pode enviar o indicador WEI de *uplink* para a estação móvel. Esta informação requer apenas 1 bit de *downlink* por rajada.

Handover entre canais de uma mesma estação base também podem ser tratados pelo MCHO. Esse tipo de *handover* é feito para evitar a atenuação da transmissão de *uplink*. Esta modalidade não será usada, se o *handover* precisar ser usado para solucionar problemas tanto na transmissão de *uplink* quanto de *downlink*.

3.4.2 - *Handover* Controlado pela Rede (NCHO)

Neste método, a estação base monitora o tamanho e a qualidade do sinal do terminal móvel. Quando esse sinal está abaixo de certo limite estabelecido, a própria rede inicia um processo de *handover* junto a outra estação base. A rede solicita a todas as estações base vizinhas que monitorem o sinal do terminal móvel e informem suas medidas. A rede escolhe uma nova estação base para o *handover*, informando tanto ao terminal

móvel (através de sua estação base) quanto à nova estação base que o *handover* será feito.

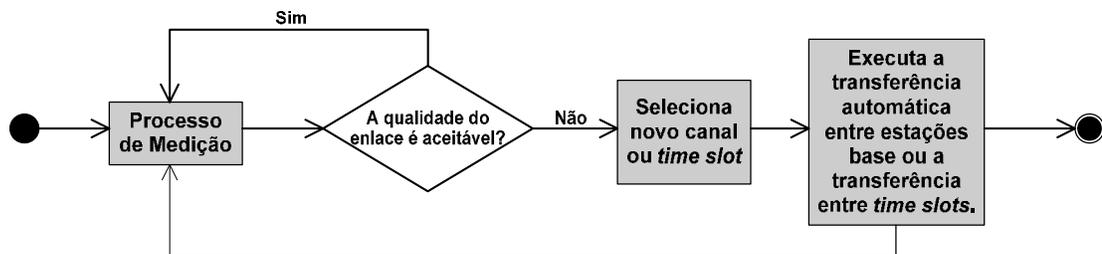


Figura 3.2 Processo de Manutenção da Qualidade do Terminal Móvel

Por causa do tráfego pesado de sinalização necessário para coletar as informações, e pela falta de recursos de rádio adequados nas estações base para fazer uma medição freqüente dos enlaces vizinhos, o *handover* é feito na ordem de segundos. Mas como as medições de sinal do terminal móvel não podem ser feitas frequentemente, a exatidão dos resultados fica reduzida. O tempo requerido para *handover* NCHO pode até ser superior a 10 segundos.

3.4.3 - *Handover* Assistido pelo Terminal Móvel (MAHO)

MAHO (do inglês, *Mobile-Assisted Handover*) é uma variante do NCHO, uma vez que a rede pede ao dispositivo móvel para medir o sinal das estações base vizinhas, enviando o resultado à estação base solicitante. De modo que esta possa decidir se um novo *handover* deve ser feito ou não. Essa estratégia é empregada pelos padrões GSM, IS-95 CDMA e IS-136 TDMA.

O GSM que utiliza o Esquema de *Handover* assistido pelo Aparelho Móvel, em que o processo de *handover* é mais descentralizado, diferentemente das redes UMTS (do inglês, *Universal Mobile Telecommunications System*), que usa o Esquema de *Handover* controlado pelo Aparelho Móvel (ou MCHO). O artigo [67] apresenta um algoritmo de *handover* entre essas redes. Esse algoritmo usa o Esquema de *Handover* controlado pelo Aparelho Móvel (ou MCHO). A razão é que quando o *handover* ocorre entre duas redes diferentes, é muito mais simples que ele seja feito pelo Aparelho Móvel do que pela rede. Na área de *handover*, o tráfego de pacotes pode migrar imediatamente para a rede UMTS, possibilitando a redução de ocupação no número de canais GPRS exclusi-

vos na BTS adjacente à UMTS. Essa liberação de canais aumenta a capacidade do serviço GSM nestas estações base, o que permitem uma maior capacidade na transmissão de voz.

No MAHO, tanto as estações móveis quanto os terminais supervisionam a qualidade dos valores de RSSI e WEI nos enlaces. No GSM, o terminal móvel transmite esses valores para a estação base, duas vezes por segundo. A decisão de quando e onde executar o *handover* é tomada pela rede, isto é, o BS e o MSC, ou o BSC. O tempo de realização do *handover* no GSM é de aproximadamente 1 segundo.

Independentemente de qual estratégia usar na detecção de *handover*, algumas das técnicas de atribuição abaixo deverá ser adotada.

3.5 - TÉCNICAS PARA ATRIBUIÇÃO DO CANAL

Os esquemas para atribuição de canais tentam alcançar um alto grau de utilização do espectro com o menor número de pesquisas em bases de dados e o mais simples algoritmo empregado, tanto na estação móvel quanto na rede.

As requisições de *handover* e o estabelecimento de novas chamadas competem para uso dos recursos de rádio. Isso pode causar o bloqueio de novas chamadas ou a falha em chamadas existentes. Note que o sucesso na realização de *handover* está intimamente ligado ao processo de atribuição do canal na tecnologia de rádio; ao qual, pode ser fixo, flexível, dinâmico ou algum outro.

Podemos classificar as estratégias de atribuição de frequência em:

- Atribuição Fixa: um conjunto de frequências de rádio é permanentemente atribuído a cada célula, ou seja, o número de canais é fixo.
- Atribuição Flexível: a frequência alocada sofre alteração conforme o volume de tráfego escalonado ou previsto.
- Atribuição Dinâmica: a frequência dos canais pode sofrer mudança a cada nova chamada iniciada.

Uma esquematização dessas estratégias é apresentada na figura 3.3.

Pelo uso das estratégias flexíveis e dinâmicas é possível aumentar a capacidade da célula, uma vez que elas utilizam o espectro disponível com maior eficiência. Infelizmente, o alto grau de complexidade dos algoritmos empregados nos BSC/MSB ainda é uma grande desvantagem.

Por essa razão, [12] utilizou a estratégia de atribuição fixa do espectro de frequência em seu trabalho.

Atribuição Fixa	Básico Fixo
	Empréstimo Simples
	Híbrido
	Empréstimo Ordenado
Atribuição Flexível	Escalonada
	Preditiva
Atribuição Dinâmica	Otimização Chamada a Chamada

Figura 3.3 Estratégias de Atribuição de Frequência

Além das estratégias para atribuição de frequência, várias técnicas para atribuição de canais foram desenvolvidas a fim de reduzir o número de perdas ou chamadas bloqueadas. Vários esquemas de atribuição de canais podem ser empregados, conjuntamente, para alcançar estes benefícios.

Uma dessas estratégias é o Esquema de Priorização de *Handover*. Esta estratégia prioriza as chamadas de voz advindas da realização de *handover*, frente à limitação do número de canais e aos diferentes tipos de tráfegos.

3.6 - ESQUEMAS DE PRIORIZAÇÃO DE *HANDOVERS*

Conforme visto, a probabilidade de *handovers* falhos pode ser reduzido por meio de esquemas de priorização dos pedidos de *handovers* sobre as tentativas de estabelecer novas chamadas. Na literatura, vários esquemas de priorização já foram estudados prevalecendo três dessas modalidades:

- reserva de um número de canais exclusivamente para a realização de *handovers*;

- empilhamento de solicitações de *handovers*;
- sub-taxação de uma chamada existente para acomodar uma solicitação de *handover*.

Em geral, o melhor esquema de priorização seria aquele que garantisse uma redução no número de *handovers* falhos, ao passo em que se reduzisse também o número de novas chamadas bloqueadas. Isso é algo inalcançável! Contudo, quanto melhor o desempenho alcançado no esquema de priorização, maior será a complexidade da sua implementação [108].

Inexistindo um esquema de priorização de *handovers* sobre as novas chamadas nas BTSs; tanto a probabilidade de bloqueio de novas chamadas quanto a probabilidade de *handovers* falhos serão iguais e teoricamente encontrados utilizando-se a Fórmula *Erlang-B*:

$$P_{cb} = P_{hf} = \frac{\frac{A_{off}^C}{C!}}{\sum_{k=0}^C \frac{A_{off}^k}{k!}} \quad (3.1)$$

onde C é o número de canais disponíveis, A_{off} é o total de tráfego ofertado ao canal k e A_{off}^C é o tráfego ofertado a todos os canais de alocação. A fórmula para o cálculo de A_{off}^k pode ser encontrada no capítulo 6 deste trabalho.

A figura 3.4 apresenta o fluxograma de execução de um modelo que não adota priorização sobre o tráfego de *handover*.

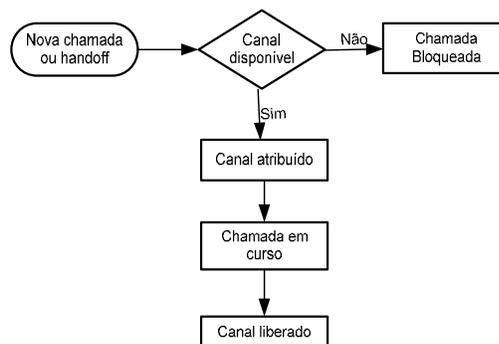


Figura 3.4 Fluxograma para Esquema sem Priorização [8]

O trabalho de [12] faz uma avaliação do tráfego de redes GSM/GPRS onde o cenário adotado na simulação contempla o uso dos canais que são alocados pelo Método de Compartilhamento Parcial (CP), utilizando técnicas de priorização de *handovers* sobre as novas chamadas.

3.6.1 - Reserva de Canais para *Handover*

Esta estratégia consiste em reservar um número de canais exclusivamente para realização de *handover*. Esses canais seriam reservados dentre os canais compartilhados, dividindo-os em: Grupo Comum de Canais (G_{comum}) e Grupo de Canais Reservados (G_{HO}). Os canais G_{comum} podem ser usados por novos canais e por *handovers*. O critério para a seleção dos novos canais de *handover* segue uma dentre as duas formas seguintes:

1. Pré-Reserva (RCS-pre)

Quando da chegada de um novo *handover*, se o G_{HO} estiver cheio é selecionado um canal em G_{comum} . Isso garante que mesmo com um volume de tráfego grande, sempre haverá um número mínimo de tráfego de *handover* aceito.

2. Pós-Reserva (RCS-pos)

Quando da chegada de um novo *handover*, primeiramente é verificado se existe canal disponível em G_{comum} . Caso contrário, o mesmo será alocado em G_{HO} . Essa técnica assegura uma prioridade extra ao *handover*.

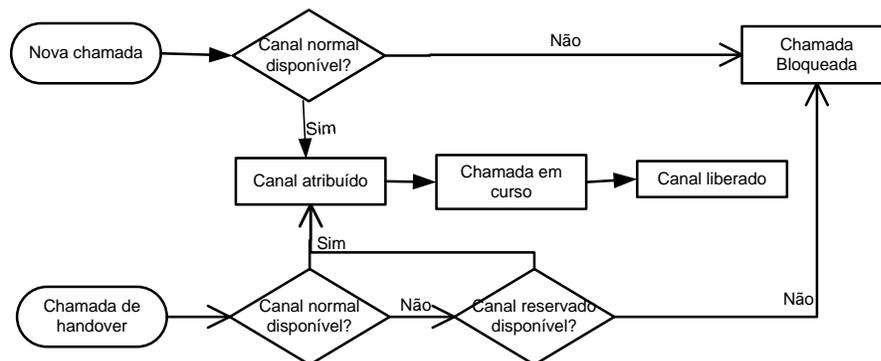


Figura 3.5 Fluxograma para Esquema de Canais Pós-Reservados para *Handover* [8]

Esquemas de priorização de *handovers* foram estudados em [82]. O número de G_{HO} pode ser fixo ou dinamicamente ajustável no BTS, BSC e MSC. A sua otimização requer conhecimento de estruturas de tráfego, pois a reserva em G_{HO} implica no aumento do número de novas chamadas bloqueadas. Para evitarmos isso, é preciso que se faça sua utilização conjuntamente com o empilhamento de novas chamadas.

Também é possível empilharmos chamadas de *handovers*, conforme veremos na seção abaixo.

3.6.2 - Esquema de Empilhamento de Chamadas

O Esquema de Empilhamento de *handovers* é possível, baseando-se no fato de que as células adjacentes nas redes PCS se sobrepõem, conforme ilustra a figura 6.5, no capítulo 6. Dessa forma, existe uma área onde as chamadas são tratadas pelas BS de ambas as células adjacentes. A essa área, dá-se o nome de Área de *Handoff* (ou *Handover*). O tempo de permanência de um aparelho móvel nessa área é chamado de Intervalo de Degradação.

A figura 3.6 mostra um fluxograma de todo o processo de estabelecimento do canal. A atribuição de um canal para novas chamadas é similar ao processo visto no esquema sem prioridade de canais, no início da seção 3.6. Se houver canal disponível para *handover*, o mesmo ocorrerá; caso contrário, ele será empilhado. Não havendo canal disponível em tempo hábil para a realização do *handover*, a conexão expirará e a chamada será interrompida abruptamente.

Caso algum canal seja liberado, o próximo *handover* a ser servido será extraído da pilha (se lá existir) baseada em uma dentre as duas políticas de empilhamento existentes para esse fim:

- Empilhamento com priorização (FIFO) ‘O primeiro a entrar, o primeiro a sair’; e,
- Empilhamento com priorização baseada na mensuração do sinal.

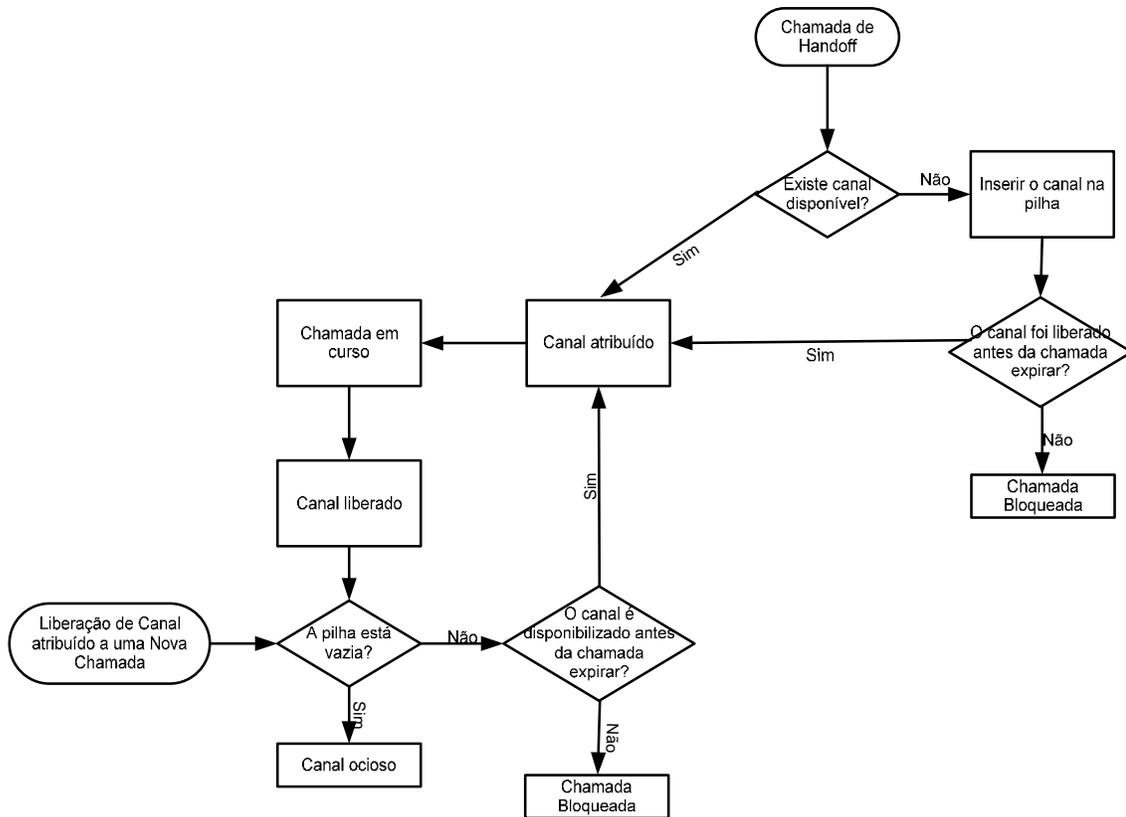


Figura 3.6 Fluxograma para o Esquema de Empilhamento de *Handovers* [8]

3.6.3 - Empilhamento com Priorização FIFO

Nesse esquema de priorização o próximo *handover* a ser servido será o mais antigo dentre os que estão na pilha, ou seja, o primeiro a chegar, será o primeiro a sair. Este é um esquema mais fácil de ser implementado, que o da mensuração do sinal.

Novas chamadas são consideravelmente menos sensíveis que os pedidos de *handovers*. Dessa forma, caso todos os canais estejam ocupados, a solicitação para uma nova chamada pode ser empilhada segundo o escalonamento ‘o primeiro a entrar é o primeiro a sair’ (FIFO). Esse empilhamento reduz consideravelmente a probabilidade de bloqueio de novas chamadas, mas ele pode causar o aumento de falhas em *handovers*, caso não seja usado um esquema de priorização dos mesmos [12].

3.6.4 - Empilhamento com Priorização Baseada na Mensuração do Sinal

Esse esquema usa uma política de priorização dinâmica não preemptiva. A prioridade é definida conforme a intensidade do sinal recebido pela BS, advindo da MS. Ou seja, a rede monitora dinamicamente o nível de força dos sinais de *handover* da pilha. Dessa forma, uma requisição de alta prioridade é aquela cujo intervalo de degradação está mais próxima de expirar.

3.6.5 - Esquema de Sub-taxação

Sub-taxação é o processo de temporariamente, dividir um canal já ocupado em dois outros com metade da taxa de ocupação do canal original onde o primeiro serviria ao tráfego já existente e o segundo, a nova chamada de *handover*.

Essa implementação é útil em casos onde não existam canais disponíveis para a realização do *handover*. Nesse contexto, a figura 3.7 apresenta um fluxograma que define a atribuição desses canais sobre o esquema de sub-taxação e [102] descreve um protocolo genérico para tal finalidade.

Para a realização do esquema de sub-taxação, é fundamental que a interface aérea garanta um protocolo de sinalização de modo que o aparelho móvel informe à rede, por meio da estação base ocupada, o seu pedido de acesso para a realização do *handover*. Nesse caso, realizar-se-á um mecanismo sub-taxando um canal existente (ex. 32 kbps) em outros dois (ex. 16 kbps); reduzindo-se, dessa forma, a vazão original. Podem existir situações onde seja inapropriado sub-taxar um canal; tal circunstância não pretende ser tratada neste trabalho por conta de sua grande complexidade. Praticamente todos os simuladores para redes GSM/GPRS modelam seus tráfegos de forma que qualquer chamada possa ser sub-taxada.

Se um codificador de voz de 16 kbps for temporariamente utilizado para chamadas de curta duração, essa potência não deverá impactar significativamente a qualidade da voz, garantindo sua tolerância. [8] afirma a existência de estudos feitos em probabilidade de blocagem de canais a fim de determinar o intervalo de tempo onde a qualidade da deteriorização ainda viabilize o uso de Esquemas de Subtaxação.

Pode ser uma desvantagem o maior custo dos aparelhos móveis adaptados para sub-taxação, mas os benefícios advindos do melhor desempenho dos recursos de rádio, compensam o custo financeiro.

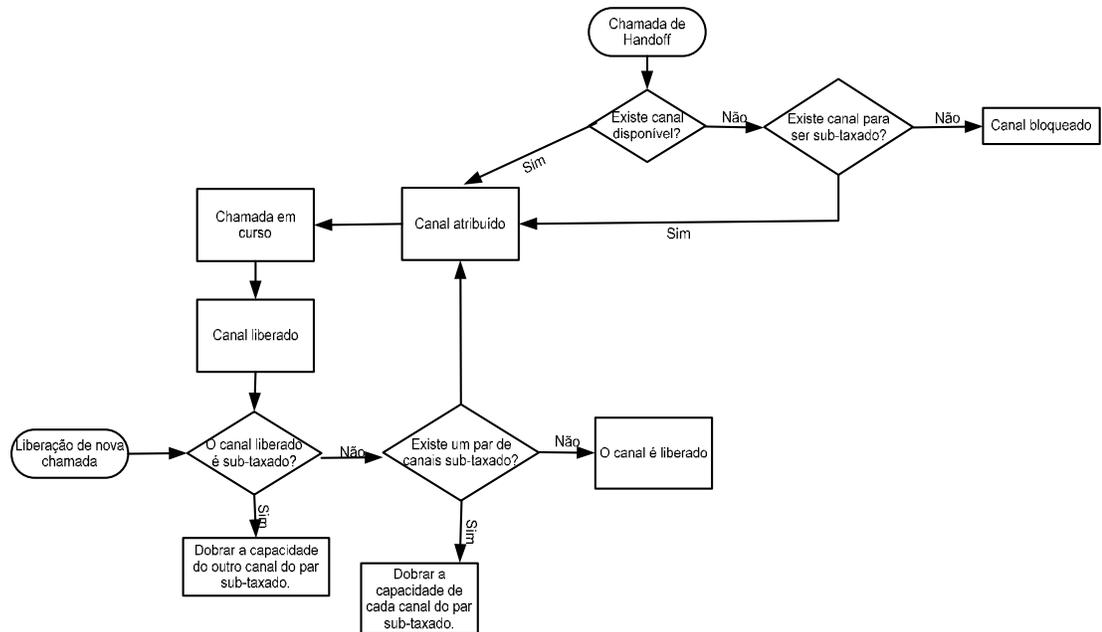


Figura 3.7 Fluxograma para Esquema de Sub-taxação [8]

3.6.6 - Esquema Híbrido

Ainda há possibilidade de combinarmos todos esses esquemas de priorização de *handovers*. Por exemplo, é possível reservarmos canais exclusivos para *handovers* ao mesmo tempo em que utilizamos uma pilha para armazenar solicitações de *handovers*. Essas duas medidas reduziram o número de *handovers* bloqueados, ao passo que a adoção de canais subtaxados reduziria o empilhamento.

Outra possibilidade é o uso simultâneo de pré-RCS e pós-RCS, conforme apresentado em [112]. Nesse caso, um número randômico entre 0 e 1 é gerado na BTS/BSC/MSB usando algum software instalado nele; se este número estiver entre 0 e 0.5 será usado o pré-RCS, caso contrário, usar-se-á o pós-RCS.

3.7 - TRANSMISSÃO PELO ENLACE TELEFÔNICO

Alguns artigos estudados ao longo da realização desse trabalho, apresentam diferentes tipos de estudos visando a melhoria na transmissão de dados pela rede telefônica móvel. Em geral, na maioria deles foram identificados trabalhos fazendo uso de buferização (empilhamento) sobre tráfego de dados e de voz. Outros tratam questões de re-roteamento do circuito de comunicação de voz, outros de sobreposição em células adjacentes e, por fim, de técnicas de inteligência artificial.

Excluindo o uso de re-roteamento, o trabalho aqui apresentado utiliza um pouco de cada um desses recursos.

3.7.1 - Analisando o Throughput e o Buffer para o GSM/GPRS

Em [109] é avaliado o *throughput* e a utilização de *buffers* em uma rede GPRS sob modelos de tráfego Internet, tais como WWW e e-mail. O desempenho do GPRS é comparado com as redes baseadas em circuito do GSM para demonstrar a melhora devido ao ganho de multiplexação.

Os resultados por ele alcançados demonstram que o GPRS alcança um aumento significativo do *throughput* se comparado ao HSCSD. Isso é possível devido à natureza ON/OFF do tráfego de pacotes de dados, resultado do ganho da multiplexação, quando comparado ao HSCSD. O GPRS aumenta a capacidade da interface aérea do GSM em mais de 300%.

O mesmo artigo demonstra que a taxa de perda de pacotes não é sensibilizada pelo aumento do tamanho do buffer disponível. Ele somente conduz a um longo retardo do empilhamento do pacote.

3.7.2 - Re-roteamento de Tráfego em Redes Telefônicas Móveis

Com o crescente número de usuários, aumenta a demanda por áreas de cobertura celular completa e com capacidade suficiente. Pensando nisso, é preferível a existência de células de diferentes tamanhos e sobrepostas implicando em múltiplas camadas de células. Uma rede telefônica com múltiplas camadas de células é dita conter uma HCS (Estrutura de Célula Hierarquizada).

Uma forma de aumentar a capacidade da rede GSM seria direcionando o tráfego comutado por circuito ou por pacotes de diferentes camadas para re-balanceá-la. Como o tráfego GPRS não pode ser re-selecionado, o tráfego comutado por circuito deve ser direcionado para outra camada (ex. de uma micro-célula para uma macro-célula).

3.7.3 - Análise de Desempenho de Redes GSM com Inteligência de Sobreposição e Subposição

Em [110] é apresentado um modelo analítico para as redes móveis celulares baseadas no GSM que aplicam o esquema de sobre-posição e sub-posição inteligentes para aumentar a capacidade pelo aumento do reuso da frequência, enquanto mantendo a Qualidade do Serviço (QoS).

A estrutura de múltiplas camadas da célula (IUO) é baseada na divisão da largura de banda da frequência em uma super camada e outra camada regular. A super camada é usada por estações móveis com boa taxa de interferência na portadora (C/I), enquanto a frequência regular é usada sobre toda a célula. O uso de IUO deve prover um ganho de mais de 40% da capacidade, segundo [111]. O artigo estuda o efeito de vários parâmetros no desempenho da rede usando IUO e garante o planejamento prático baseado nos resultados analíticos. Os parâmetros considerados incluem os parâmetros de rede, como a super área de cobertura e os parâmetros de mobilidade do usuário móvel como a taxa de movimentação móvel e a velocidade móvel média.

A alta densidade em uma estação base resulta no aumento da interferência limitando a qualidade e a capacidade. Nesse sentido, a subcamada adiciona capacidade enquanto que a sobre-camada garante a cobertura celular. Reduzir a área de cobertura da super camada tem um impacto negativo na utilização da célula.

As frequências da super camada garantem os serviços em áreas de tráfego pesado (pequenas cidades), onde a taxa C/I seja boa (área livre de interferência). A taxa C/I é calculada comparando o sinal *downlink* de todas as células vizinhas que usam a mesma super frequência.

Quando a carga diminui, pode ser alcançada uma melhor taxa de utilização. Enquanto isso, em cargas muito pesadas a movimentação tem um fator negativo na taxa de utilização das super frequências resultando na redução de todo o fator de utilização da célula.

Em [110], é feita uma análise de Markov baseada em investigações que focam o efeito desse esquema em uma célula, levando em conta o fator de cobertura das super frequências e o movimento das estações móveis.

Segundo [110], estão sendo feitos estudos planejando o uso de agrupamentos de vários níveis de frequências no modelo, bem como o estudo de seu efeito quando uma MS não tiver que usar a camada regular em primeiro lugar.

3.8 - CONSIDERAÇÕES

O sistema de rádio deve ter um canal físico para sinalização do sistema, para implementar esquemas de priorização de *handover*. Assim, o terminal móvel pode mandar requisições ao enlace, mesmo quando todos os canais de tráfego estão em uso. Este canal deve estar sempre disponível; portanto, não pode ser usado como canal de tráfego. Alguns sistemas PCS já reservam um canal para o propósito de *broadcast*; ao qual pode ser compartilhado pelo procedimento de priorização de *handover*.

Vários modelos analíticos e de simulação têm sido propostos para avaliar o desempenho dos esquemas de atribuição dos canais de *handover*. O RCS é fácil de implementar e ele reduz a probabilidade de *handovers* falhos; sendo mais eficiente que o NPS (do inglês, *Non-Priority Schema*).

Os esquemas de empilhamento têm a vantagem de buferizar as chamadas de *handover* na área de *handover*. A implementação para o esquema de priorização baseado em mensuração é mais complexo que o esquema FIFO (do inglês, *First-In First-Out*), mas o desempenho é idêntico, conforme [8]. Esquemas de empilhamento efetivamente reduzem as falhas de *handover*, aumentando, controladamente, o não estabelecimento de novas chamadas. A probabilidade de chamadas incompletas em FIFO e MBPS (do inglês, *Measurable Band Priority Schema*) é menor que para NPS. O esquema de empilhamento adiciona complexidade ao hardware e software para que o BS e o MS possam gerenciar as filas.

O esquema de subtaxação tem as menores probabilidades de terminação forçada e de chamadas incompletas. Esse benefício é ganho com o aumento da complexidade dos dispositivos de hardware e software que permitem a subtaxação do canal. O custo pode ser compartilhado com outras funções do PCS; por exemplo, a idéia de que canais subtaxados possam ser usados em chamadas emergenciais (911) quando todos os canais estiverem ocupados.

Para concluir, a seleção de um esquema de atribuição particular de canais para *handover* deve ser balanceada entre a complexidade e o desempenho da rede. Se a redução da terminação forçada for mais importante que o bloqueio de novas, então RCS, QPS (do inglês, *Queue Priority Schema*) e SRS (do inglês, *Sub-Rated Schema*) são melhores que NPS. Se a densidade da BS for alta em uma dada área de serviços PCS, então um esquema de empilhamento de *handover* pode ser uma boa escolha, porque as áreas de sobreposição entre as BS são muito largas.

O capítulo seguinte apresenta uma visão geral do processo de chamadas remotas de procedimento, para arquiteturas orientadas a serviços (SOA) baseado em protocolos de Internet. Essa arquitetura é usada na modelagem, implementação e implantação do estudo de caso dessa tese. Para a distribuição dos processos, dar-se-á uma visão geral de EJBs *Session Beans*, usados também na implementação deste modelo proposto.

4 – ARQUITETURA ORIENTADA A SERVIÇOS VOLTADO À INTERNET

As corporações sempre tiveram uma grande demanda para integrar os sistemas existentes (legados) às novas aplicações. Na maioria dos casos, essa integração acontece por meio de soluções tecnológicas proprietárias, sem um projeto e/ou arquitetura bem definida. Daí, surge a necessidade de prover uma solução que suporte conexão e distribuição de recursos e dados de forma flexível e padronizada.

A orientação a serviços apresenta uma visão onde os recursos são particionados e representados com muita clareza. Aplicado a arquitetura de tecnologia da informação, a orientação a serviço estabelece um modelo genérico, cuja lógica de automação e de negócio se juntam com a finalidade de transparência dos serviços junto às demais aplicações e plataformas que formam a base de conhecimento de uma organização. Esse modelo também pode ser aplicado na realização de uma tarefa, uma solução, uma empresa, uma comunidade e outros.

Todos os principais construtores de software estão oferecendo suporte para SOA. Como resultado, todas as principais plataformas de desenvolvimento suportam a criação de soluções orientadas a serviços [16].

Para a construção de uma aplicação SOA, é necessária uma mudança (ou adaptação) no conhecimento do negócio, já que a lógica de negócio precisa ser vista dentro de um contexto orientado a serviço. Aplicar esse contexto também necessita de uma mudança na lógica de automação, visto que as soluções precisarão ser construídas suportando orientação a serviço. Um ambiente técnico capaz de suportar lógica de automação orientada a serviço introduz novas tecnologias, infra-estruturas e requisitos governamentais.

Pela forma com que a lógica é automatizada através da orientação a serviços, investimentos existentes podem ser alavancados, a inteligência de negócio pode ser cuidadosamente expressa e a inerente agilidade na automação também pode ser alcançada.

SOA tem sido apresentado como a próxima fase na evolução da automação de negócio. Da mesma maneira que os sistemas de mainframe foram sucedidos por aplicações cliente-

servidor, ambientes cliente-servidor evoluíram para soluções distribuídas baseadas em tecnologias WEB. Na atualidade, o SOA orientado a serviços WEB está sucedendo as tradicionais arquiteturas distribuídas numa escala global.

4.1 INTEROPERABILIDADE NOS SERVIÇOS WEB

Conforme sugerido no capítulo 1 deste trabalho, o modelo proposto nessa tese deve preocupar-se com questões como: portabilidade, escalabilidade e distribuição de informações para outros sistemas legados. Todas essas características reunidas sugerem o emprego de uma arquitetura de sistemas projetada de forma a expor serviços através de uma visão bem definida de componentes, ou pequenas unidades modulares, focada na distribuição dos serviços implementados, e na interoperabilidade com outros sistemas já em uso.

Com base nessas necessidades foi empregado, nesta tese, o conceito de arquitetura SOA. Essa arquitetura é focada em como uma série de serviços são descritos e organizados. Com isso, tem-se suporte a descoberta e uso dinâmico e automatizado de serviços. Assim, a integração pode ocorrer durante a execução da aplicação e não durante o desenvolvimento, facilitando a expansão e uso de novos serviços.

Essas arquiteturas obedecem aos requisitos de: interoperabilidade, descrição do serviço, recuperação do serviço e segurança.

Interoperabilidade – é a base mais importante para uma integração simples entre aplicações, plataformas e protocolos de comunicação diferentes.

Descrição do serviço – é a definição para uso efetivo de um serviço. Além da necessidade de acessá-lo, é preciso ter uma *interface* bem definida, em uma forma independente de plataforma. Aplicações SOA têm essa preocupação como garantia à interoperabilidade. A linguagem WSDL, comentada adiante, serve para descrever serviços WEB.

Recuperação do Serviço – é um mecanismo que permite a busca de serviços disponíveis. Ele permite a integração em tempo de desenvolvimento e/ou em tempo de execução. Ele deve classificar os serviços baseados no que os mesmos serviços podem fazer e dizer e em como eles podem ser invocados. Em geral, essa característica é alcançada nos serviços WEB por meio do protocolo de comunicação SOAP (do inglês, *Simple Object Access Protocol*), também comentado adiante.

Segurança – é a proteção do serviço e da informação trafegada contra acesso não autorizado. Vários mecanismos de proteção podem ser empregados, conforme a plataforma em questão. O trabalho desta tese não trata essa última característica, apenas as demais.

4.2 COMPONENTES DO SOA

Para garantir a transparência das aplicações SOA junto às diversas plataformas interoperáveis, os componentes da figura 4.1 formam a arquitetura SOA:

Provedor do Serviço – Responsável pela criação e publicação das informações de *interface* e acesso do Serviço WEB.

Registrador do Serviço – Disponibiliza a *interface* e informações de acesso do Serviço WEB para potenciais usuários do serviço.

Requisitor do Serviço – Localiza os serviços publicados no registro, invocando os serviços que atendem as necessidades solicitadas.

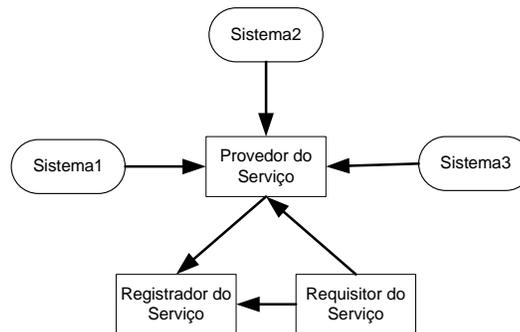


Figura 4.1 Interação entre os componentes SOA e Sistemas Legado

Dessa forma, um sistema descrito em uma linguagem que suporte essa arquitetura, solicita ao Provedor de Serviço uma funcionalidade disponibilizada pelo Registrador de Serviço através de uma *interface*. Aí, o Requisitor de Serviço fica responsável por invocar o serviço solicitado na referida interface.

4.3 CARACTERÍSTICAS DO SOA

Em geral, as aplicações SOA são auto contidas, onde cada serviço é apresentado como um pequeno bloco de código totalmente funcional e independente de outros. Essa independên-

cia faz com que o cliente e o servidor não precisem tomar conhecimento um do outro, apenas do formato e conteúdo da mensagem, que é por si só, auto-descritiva.

Em nenhum momento, o cliente se encontra acoplado a um servidor, mas sim, a um serviço; garantindo o baixo acoplamento dessas aplicações e permitindo a troca de objetos de forma dinâmica quando o serviço é solicitado.

Outra característica das aplicações SOA é a separação de responsabilidades. A *interface* é totalmente separada da implementação, o que permite a troca de objetos sem afetar o cliente.

A publicação, localização e invocação desses serviços é feita através da Internet, independente de linguagem e interoperáveis. Sendo baseados nos padrões abertos e de fácil utilização, como XML (do inglês, *Extensible Markup Language*) e HTTP, a arquitetura SOA pode facilmente expor aplicações já existentes.

4.4 A TECNOLOGIA DE SERVIÇOS WEB

Serviços WEB é uma tecnologia nova que aplica os conceitos de SOA. Seu desenvolvimento está focado no uso dos padrões da Internet, sendo independentes de plataforma e linguagens.

Esses serviços são auto contidos e modulares, podendo ser descritos, publicados, localizados e invocados por meio de uma arquitetura de serviços transparentes e tendo como base as seguintes tecnologias:

XML – Linguagem de marcação, usada para descrever qualquer conteúdo de uma forma estruturada.

SOAP – Protocolo de comunicação independente de plataforma, responsável pelo transporte de uma mensagem em XML.

WSDL – Linguagem de descrição, usada para especificar as operações disponíveis em um Serviço WEB, seus parâmetros e tipos de dados, assim como o acesso ao serviço.

UDDI (do inglês, *Universal Description Discovery and Integration*) – Usado para guardar, catalogar e recuperar informações sobre serviços disponibilizados por vários módulos de serviços WEB.

A WS-I (do inglês, *WEB Service Interoperability Organization*) é uma organização criada com o propósito de promover a interoperabilidade de Serviços WEB entre as várias plataformas, sistemas operacionais e linguagens de programação.

Os perfis básicos do WS-I listam uma série de requisitos nos quais os padrões de WSDL e SOAP/HTTP devem aderir para garantir a interoperabilidade. Esses padrões têm como base o transporte de informações que usam linguagens de marcações XML para a formatação estruturada dessas informações.

Conforme sugerido, as tecnologias de serviços WEB permitem a exposição de sistemas legados permitindo sua interoperabilidade com outros sistemas e plataformas. Elas capturam informações dos mais variados sistemas, desde Relatórios Meteorológicos, Planos de Viagens, etc.

O projeto [17] apresenta a construção de um banco de dados distribuído para a publicação de informações biológicas através do uso de uma arquitetura de serviços WEB [18, 19]. Atualmente, já se encontra disponível em [20] ferramentas de *software* para auxiliar na integração entre aplicações de negócio e serviços WEB em redes de telecomunicações, usando o protocolo de comunicação SOAP.

4.4.1 Protocolo SOAP

O protocolo SOAP é basicamente uma especificação em XML para a troca de informações em um ambiente descentralizado e distribuído. Sendo a principal forma de comunicação em SOA, seu objetivo é ser simples e extensível a outras aplicações. Independente do protocolo de transporte, do sistema operacional e da linguagem de programação, ele lida apenas com texto puro (XML), por isso não é possível ter referências remotas.

Uma mensagem SOAP é composta de um elemento envelopado, com zero ou mais elementos no cabeçalho do pacote e um corpo de elemento onde a informação verdadeiramente trafega. Assim, os três pilares do SOAP são:

Envelope: é a raiz do documento XML. Funciona como um *container* da informação, do endereço e do conteúdo da mensagem.

Cabeçalho da Mensagem: é um item opcional. Contém informações de controle da mensagem, como garantias de serviço ou segurança.

Corpo da Mensagem: contém a identificação da mensagem, assim como todos os seus parâmetros.

O exemplo abaixo foi extraído do estudo de caso desenvolvido nessa tese, e mostra uma requisição SOAP feita via HTTP. Nele podem ser encontradas as *tags* de envelope, cabeçalho e corpo da mensagem. Aqui, uma requisição feita para o serviço `obtemAmostraIdealMacrocelulaFuzzy` retorna um objeto do tipo `Amostra`.

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <obtemAmostraIdealMacrocelulaFuzzyResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <obtemAmostraIdealMacrocelulaFuzzyReturn
        xsi:type="ns1:Amostra"  xsi:nil="true"
        xmlns:ns1="http://gerente.TELCOM.sti.doutorado.dee.fat.unb.br" />
      </obtemAmostraIdealMacrocelulaFuzzyResponse>
    </soapenv:Body>
  </soapenv:Envelope>
```

4.4.2 Modelos de Mensagem

Caso a comunicação seja em um estilo mais orientado a mensagem, o modelo de comunicação será do tipo *document*. Por ele, qualquer tipo de documento pode ser enviado. Esse estilo provê uma pequena camada de abstração, e necessita de mais trabalho de programação. O parâmetro de entrada é um documento XML, e a resposta pode ser qualquer coisa, até mesmo nada. Este é um modelo de comunicação muito flexível.

A linguagem WSDL também viabiliza comunicação RPC (Chamada Remota de Procedimento) via SOAP. Assim, a API (do inglês, *Application Programming Interface*) usada para fazer chamadas remotas em linguagem Java é a JSR-101 Java API para XML baseada em RPC – JAX-RPC, também usada nesta tese.

Diferentemente da comunicação que pode ser RPC ou *document*, um padrão de codificação define como os dados da aplicação são traduzidos para um determinado protocolo. Esse processo é conhecido como Serialização e Deserialização.

Mensagens, parâmetros e APIs usadas são diferentes baseado no estilo de comunicação. Dessa forma, a decisão sobre qual estilo usar, RPC ou *document*, é feito em tempo de projeto.

Como o protocolo SOAP é independente de plataforma, a codificação SOAP diz ao ambiente de execução como traduzir as estruturas de dados construídas na linguagem SOAP XML.

Dois são os padrões de uso desse protocolo: *encoded* e *literal*. O padrão SOAP *encoded* permite a serialização e deserialização de dados do modelo de transmissão de dados SOAP. Já o *literal* representa uma mensagem XML que não carrega nenhuma informação de codificação.

Dessa forma, o modelo da mensagem SOAP é definido combinando-se os estilos (RPC ou *Document*) e as codificações (*encoded* ou *literal*) quando projetando o serviço WEB.

O uso do modelo *document/literal* provê a melhor interoperabilidade entre Java e outras linguagens, sendo também recomendada para comunicação Java-Java.

O modelo *RPC/literal* possui algumas limitações, principalmente no mapeamento de *Arrays*. Este modelo não é frequentemente usado em implementações Java [21]. Já o modelo *RPC/encoded* não provê uma boa interoperabilidade com implementações feitas em linguagens diferentes de Java.

Por fim, o modelo *document/encoded* não é usado na prática, por aplicações conhecidas; conforme afirma [21].

O perfil básico do WS-I indica o uso da codificação literal; dessa forma, apenas *document/literal* e *RPC/literal* devem ser usados por compatibilidade.

4.4.3 Elementos de uma WSDL

Enumerando, os elementos de uma WSDL qualquer são:

Types: local para a definição de tipos de dados usados pelo serviço; usa como padrão os tipos definidos no esquema XML.

Message: é uma abstração de uma definição de dados usados na comunicação do serviço, e é composto por uma ou mais partes.

Port type: é um conjunto de uma ou mais *operations*, suportada por uma ou mais *ports*.

Operation: é uma descrição abstrata de uma ação suportada pelo serviço, define a mensagem de entrada e de saída, além de um elemento *fault* (opcional).

Binding: é a especificação do protocolo e o formato de dados para um *port type*. A informação de *binding* contém o protocolo, modelo de invocação, identificador do serviço e codificação para cada *operation*.

O apêndice A mostra a WSDL definida para a realização do trabalho desta tese. O exemplo abaixo apresenta uma visão geral da organização desses elementos na descrição anexa.

```
<?xml version="1.0" encoding="UTF-8" ?>
<wSDL:definitions targetNamespace="..." xmlns="..." >
  <wSDL:types>
    <schema targetNamespace="..." xmlns="...">
  </schema>
```

```

</wsdl:types>
<wsdl:message ...>
</wsdl:message>
<wsdl:portType name="FachadaTelefonica">
    <wsdl:operation name="getInstancia">
        </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="..." type="...">
    <wsdl:operation name="...">
        <wsdlsoap:operation soapAction="" />
        <wsdl:input name="...">
            <wsdlsoap:body encodingStyle="..." namespace="..." use="encoded" />
        </wsdl:input>
        <wsdl:output name="...">
            <wsdlsoap:body encodingStyle="..." namespace="..." use="encoded" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="FachadaTelefonicaService">
    <wsdl:port binding="..." name="...">
        <wsdlsoap:address location="..." />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

A WSDL facilita a interoperabilidade, viabilizando o RPC via SOAP. Dessa forma, pode-se gerar a parte do cliente em uma plataforma de linguagem (por exemplo, .NET) e a parte do servidor em outra (por exemplo, J2EE – do inglês, *Java 2 Enterprise Edition*), viabilizando a comunicação entre arquiteturas diferentes.

4.4.3.1 Types

Types são elementos que definem os tipos de dados que são usados na troca de mensagens. Ele usa como padrão os tipos de dados definidos em XML-Schema.

Esse elemento é extensível, possibilitando que outros tipos de dados sejam montados a partir do tipo de dados padrão.

Abaixo, segue a declaração de dois tipos de dados: amostra e vetor de amostra. Ambos servem para armazenar os atributos do tráfego telefônico.

```
<wsdl:types>
  <schema targetNamespace="..." xmlns="...">
    <import namespace="..." />
    <complexType name="Amostra">
      <sequence>
        <element name="blockedCalls" type="xsd:double" />
        <element name="carriedTraffic" type="xsd:double" />
        <element name="gprs" type="xsd:double" />
        <element name="handover" type="xsd:double" />
        <element name="handoverFailure" type="xsd:double" />
        <element name="handoverUtilization" type="xsd:double" />
        <element name="offeredTraffic" type="xsd:double" />
        <element name="percHO" type="xsd:double" />
        <element name="percVO" type="xsd:double" />
        <element name="qps" nillable="true" type="xsd:string" />
        <element name="rcs" nillable="true" type="xsd:string" />
        <element name="signaling" type="xsd:double" />
        <element name="subrated" type="xsd:double" />
        <element name="totalCanais" type="xsd:double" />
        <element name="voiceUtilization" type="xsd:double" />
      </sequence>
    </complexType>
    <complexType name="ArrayOfAmostra">
      <complexContent>
        <restriction base="soapenc:Array">
          <attribute ref="soapenc:arrayType" wsdl:arrayType="tns1:Amostra[]" />
        </restriction>
      </complexContent>
    </complexType>
  </schema>
</wsdl:types>
```

```

        </complexContent>
    </complexType>
</schema>
<schema targetNamespace="..." xmlns="...">
    <import namespace="..." />
    <complexType name="FachadaTelefonica">
        <sequence />
    </complexType>
</schema>
</wsdl:types>

```

4.4.3.2 Messages

É composta por uma ou mais partes, e representa uma interação entre o requisitor e o provedor. Esse elemento é usado pelo elemento *operation*, várias operações podem referenciar a mesma message.

Operations e *messages* são modelados em separado para serem mais flexíveis e melhorar o reuso.

As duas tags <wsdl:message> abaixo mostram o serviço **reconfiguraInferencia MacrocelulaNeural**. Um objeto **Amostra** deve ser passado na requisição do serviço, enquanto um objeto **booleano** é retornado como resultado do serviço.

```

<wsdl:message name="reconfiguraInferenciaMacrocelulaNeuralRequest">
    <wsdl:part name="amostra" type="tns1:Amostra" />
</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMacrocelulaNeuralResponse">
    <wsdl:part name="reconfiguraInferenciaMacrocelulaNeuralReturn"
        type="xsd:boolean" />
</wsdl:message>

```

4.4.3.3 Port type

Representa um conjunto de operações e mensagens abstratas.

Existem quatro tipos de operações. O tipo de operação é definido pela presença e ordem dos elementos *input*, *output* e *fault*; são eles:

One-Way – Apenas a mensagem de *input* é enviada;

Request-Response – uma mensagem de *input* é enviada, seguida por uma de *output*;

Solicit-response – uma mensagem de *output* é enviada pela porta, seguida de uma mensagem de *input*;

Notification – a porta envia uma mensagem, sendo composta pela mensagem de *output*.

O exemplo abaixo, também extraído do apêndice A desta tese, mostra a definição de uma porta chamada *FachadaTelefonica*. Esta forma declara a operação *obtemAmostraIdealMicrocelulaFuzzy*, com mensagens de *input* e *output* definidas para essa operação.

```
<wsdl:portType name="FachadaTelefonica">
  <wsdl:operation name="obtemAmostraIdealMicrocelulaFuzzy" parameterOrder="trafegoOfertado percentualHandover">
    <wsdl:input message="impl:obtemAmostraIdealMicrocelulaFuzzyRequest" name="obtemAmostraIdealMicrocelulaFuzzyRequest" />
    <wsdl:output message="impl:obtemAmostraIdealMicrocelulaFuzzyResponse" name="obtemAmostraIdealMicrocelulaFuzzyResponse" />
  </wsdl:operation>
</wsdl:portType>
```

4.4.3.4 Binding

O elemento *binding* contém:

- informações específicas do protocolo de comunicação, como método de transporte (HTTP) e estilo da mensagem, cujo exemplo abaixo apresenta os valores SOAP/HTTP e RPC, respectivamente;

- Extensões do protocolo para as operações, como codificação.

Cada *binding* faz referência a um *port type*, todas as operações definidas em um *port type* devem ser vinculadas no *binding*.

No exemplo abaixo pode-se ver as seguintes informações: o nome do bind – *FachadaTelefonicaSoapBinding*. A conexão como sendo HTTP e o estilo RPC. A referência para a operação *obtemAmostraIdealMicrocelulaFuzzy* e a definição das mensagens de entrada e saída como SOAP *encoded*. Mensagens de erro não estão definidas neste exemplo.

```
<wsdl:binding name="FachadaTelefonicaSoapBinding" ty-
                                pe="impl:FachadaTelefonica">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="obtemAmostraIdealMicrocelulaFuzzy">
    <wsdlsoap:operation soapAction="" />
    <wsdl:input name="obtemAmostraIdealMicrocelulaFuzzyRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded" />
    </wsdl:input>
    <wsdl:output name="obtemAmostraIdealMicrocelulaFuzzyResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

4.4.3.5 Service e Port

Service é o nome que representa um conjunto de portas, você pode definir vários serviços em um WSDL. O exemplo mostrado no apêndice A declara um serviço com nome *FachadaTelefonicaService*, conforme mostrada abaixo.

Port é a descrição de um ponto final de acesso; especifica um endereço único para um *binding*. A porta chamada *FachadaTelefonica* referencia o *binding* *FachadaTelefonicaBinding* sendo o seu endereço o valor do atributo *location* de *soap:address*.

```
<wsdl:service name="FachadaTelefonicaService">
```

```

<wsdl:port binding="impl:FachadaTelefonicaSoapBinding"
            name="FachadaTelefonica">
  <wsdlsoap:address loca-
    tion="http://localhost:8080/STIServer/services/FachadaTelefonica" />
</wsdl:port>
</wsdl:service>

```

4.4.3.6 Chamadas RPC

O RPC representa uma chamada síncrona a uma operação e retorna uma resposta. Esse é o modelo mais explorado pelas ferramentas. Para fazer uma chamada usando RPC, as seguintes informações são necessárias:

- URI (do inglês, *Uniform Resource Identifier*) do serviço;
- Nome do método;
- Parâmetros do método;
- Dados de cabeçalho (opcional).

Quando se usa HTTP como o protocolo de transporte de uma mensagem SOAP, uma chamada RPC se torna um HTTP *request* enquanto que a resposta RPC se torna um HTTP *response*.

4.4.3.7 Mapeamentos

Um mapeamento define uma associação entre um nome XML qualificado e o nome de uma classe. Dessa forma, o mapeamento é específico de cada linguagem. O mapeamento especifica como um elemento XML deve ser convertido em uma classe, dada uma determinada regra de codificação, e vice-versa.

Todo ambiente SOAP com Java deve definir uma tabela com os mapeamentos de conversão de XML para Java e de Java para XML.

A API JAX-RPC suporta as seguintes classes J2SE (do inglês, *Java 2 Standard Edition*) na interface de mapeamento:

- java.lang.Boolean;
- java.lang.Byte;
- java.lang.Double;
- java.lang.Float;
- java.lang.Integer;
- java.lang.Long;
- java.lang.Short;
- java.lang.String;
- java.math.BigDecimal;
- java.math.BigInteger;
- java.net.URI;
- java.util.Calendar;
- java.util.Date.

Essa mesma API suporta os seguintes tipos primitivos da linguagem Java:

- boolean;
- byte;
- double;
- float;
- int;
- long;
- short.

Vetores também podem ser passados pela interface dos serviços WEB, tais como int[] e String[], por exemplo. Ou mesmo vetores multidimensionais, como BigDecimal[][].

Uma classe qualquer também pode ser passada entre um cliente e um serviço remoto, como parâmetro de um método ou valor de retorno. Mas, para ser suportado pelo JAX-RPC é fundamental que:

- a classe tenha um construtor definido;
- a classe em linguagem java não pode implementar java.rmi.Remote;
- seus campos devem suportar os tipos JAX-RPC, devendo implementar as assinaturas de método do SEI (Interface Final do Serviço);
- a classe deve ser pública, não podendo ser final ou abstrata; e,
- a classe não deve definir o método *finalize*, em linguagem Java.

4.4.3.8 Códigos e tratamentos de erros

O SOAP define um pequeno conjunto de quatro códigos de falhas, apesar da especificação SOAP permitir o uso de códigos de falhas personalizados, o perfil básico do WS-I permite apenas o uso desses quatro:

soapenv:Client – indica que a mensagem não foi enviada na formatação correta;

soapenv:Server – indica que houve algum problema na entrega da mensagem;

soapenv:VersionMismatch – indica que algum *namespace* definido no envelope é inválido;

soapenv:MustUnderstand – indica erro ou falha ao processar algum cabeçalho com o atributo `mustUnderstand` igual a 1.

Enquanto isso, ***Fault*** é um elemento definido dentro da *tag body*, cuja função é reportar erros. Deve aparecer em *body* uma única vez e possui os seguintes campos:

Faultcode: identificador do tipo erro;

Faultstring: descrição do erro, obrigatório dentro de um *fault*;

Faultactor: opcional, indica a URI onde ocorreu o problema, *actors* que não representam o destino final de uma mensagem devem incluir o *faultactor* caso ocorra algum problema;

Detail: contém informações detalhadas do erro para a aplicação cliente, não deve ser usado para carregar informações sobre erros encontrados durante o processamento de *header*, apenas do *body*.

4.5 SERVIÇOS WEB CORPORATIVOS

A especificação de JSR 109 (do inglês, *Java Specification Request*) define a forma de empacotar, o deploy e o modelo de programação para Serviços WEB em um ambiente J2EE. Esse padrão torna o desenvolvimento de Serviços WEB portátil entre servidores de aplicação J2EE 1.4.

A figura 4.2 mostra a estrutura de módulo empregada para o empacotamento da aplicação servidora. Ela também mostra os arquivos de configuração do serviço WEB no ambiente do servidor feita através de descritores (XML) e dos arquivos .jar no pacote lib.

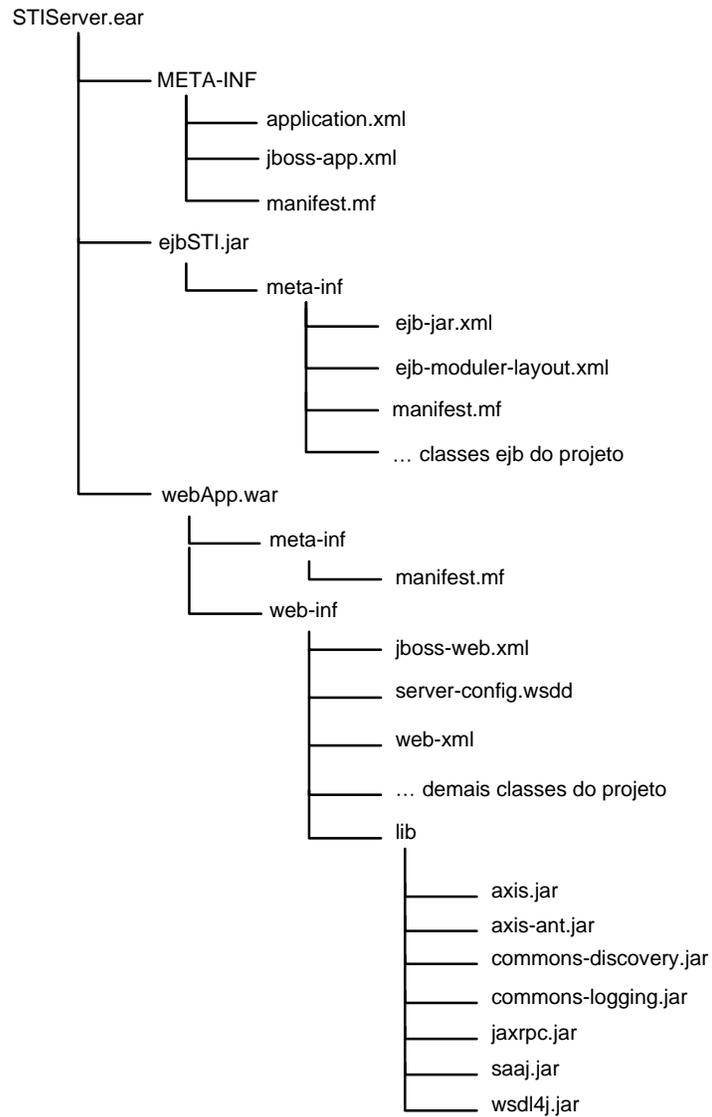


Figura 4.2 Estrutura de empacotamento do projeto

Na estrutura de empacotamento desta tese foi empregado o uso de EJBs para implementar uma arquitetura que possibilitasse a distribuição dos componentes de processo.

4.5.1 Componentes EJBs

Em uma aplicação J2EE típica, os EJBs (do inglês, *Enterprise Java Beans*) contêm a lógica do negócio do aplicativo e dos dados corporativos ativos. Embora seja possível o uso de objetos Java padrão para construir a lógica de negócio e dos dados corporativos, usar EJBs

resolve, de forma automática, muitos dos problemas de escalabilidade, gerenciamento de ciclo de vida e gerenciamento de estados.

Um EJB é basicamente um componente gerenciado que é criado, controlado e destruído pelo contêiner J2EE em que vive. Esse controle permite que o contêiner controle o número de EJBs existentes e os recursos que estão usando, como memória e conexões de banco de dados. Cada contêiner manterá um conjunto de instâncias de EJBs que já estão prontas para serem atribuídas a um cliente. Quando um cliente não precisar mais de um EJB, a instância do EJB voltará para o conjunto e todos os seus recursos serão liberados. Em momentos de carga pesada, até as instâncias do EJB que ainda estão em uso pelos clientes voltarão para o conjunto, para que possam atender outros. Quando o cliente original fizer outra requisição de seu EJB, o contêiner reconstituirá a instância do EJB original para atender à requisição. Esse uso do conjunto e a reciclagem de instâncias do EJB significam que poucas instâncias e os recursos que elas utilizam podem ser compartilhadas entre muitos clientes. Isso maximiza a escalabilidade do aplicativo com base em EJBs.

Em geral, os componentes EJBs provêm suporte para serviços a nível de sistema, tais como transações, segurança e conectividade com bancos de dados com certo nível de transparência. Embora, a forma de desenvolvimento permite aos desenvolvedores se concentrarem na implementação da lógica do negócio, evitando que eles se preocupem com a codificação dos descritores de distribuição.

Exemplos de descritores de distribuição são encontrados no apêndice C desta tese. Lá, são apresentados os dois *EJBs Session Beans*, construídos para possibilitar a distribuição das informações ao longo da plataforma de operacionalização.

O capítulo seguinte apresenta o modelo de distribuição proposto nesta tese que tem por objetivo monitorar, analisar e controlar, por meio de simulação, diferentes cenários. Sobre todos esses cenários são simuladas a maioria das técnicas de alocação de canais apresentadas nesse capítulo.

5 - MODELAGEM ADAPTATIVA PARA REDES DE COMUNICAÇÕES MÓVEIS

De forma bem simples, a telefonia móvel celular originou-se da necessidade de garantir aos clientes de telefonia fixa a liberdade de mobilidade tão almejada por estes usuários.

Dentro de um processo evolutivo, a telefonia móvel teve grande parte de sua infra-estrutura inicial difundida ao longo dos anos 80, em todos os países desenvolvidos e em desenvolvimento. Conforme apresentado na seção 2.1, esses telefones apresentavam algumas características particulares à limitação da época. Uma dessas características era a limitação de serviços unicamente voltados à transmissão de sinais de voz.

A partir de então, a telefonia móvel vem sofrendo constantes alterações; a limitação dos recursos que outrora permitiam, apenas, a transmissão de voz, vem abrindo espaço à convergência com outras mídias. Essa convergência têm sido o grande desafio atualmente.

A telefonia móvel está convergindo para uma integração dos serviços de voz, dados, imagens e vídeos, que deverão ser fornecidos com diferentes perfis de qualidade de serviço. Alguns desses serviços são mais sensíveis ao atraso que outros, o que torna necessário o uso de esquemas de priorização. Sendo assim, baseado nas diferentes Qualidades de Serviços dessas mídias é fundamental um ambiente que se preocupe com questões como controle, monitoramento e desempenho; otimizando o compartilhamento dos recursos e a comunicação entre as diferentes arquiteturas de protocolos envolvidas.

Evidenciando um melhor dimensionamento dessas redes, este capítulo apresenta uma plataforma de monitoramento, avaliação e controle; modularizada com os seus processos distribuídos ao longo de toda a infra-estrutura telefônica, cuja funcionalidade inicial é garantir a integração de serviços de voz e dados em quaisquer plataformas heterogêneas. O estudo de caso desenvolvido abrange uma plataforma de distribuição das interfaces dos serviços WEB e sua portabilidade.

Para isso, uma plataforma foi especificada, projetada e implementada de modo a que sua arquitetura pudesse evoluir e se adequar, com certa facilidade, aos avanços tecnológicos.

5.1 - ESPECIFICAÇÃO DO MODELO TELEFÔNICO

A Engenharia de Serviços de Telecomunicações trata de aspectos de especificação, projeto, implementação, gerenciamento e validação dos serviços de telecomunicações, além de sua implantação sobre arquiteturas de rede existentes e futuras [3]. Seu maior desafio tem sido o de orientar na construção de uma infra-estrutura de comunicação baseada em duas características básicas: (i) a integração dos mais variados serviços em um único sistema de comunicação, e (ii) a criação de um ambiente no qual esses mesmos serviços sejam facilmente criados, alterados e adaptados às novas necessidades e condições. É cada vez mais difícil encontrarem-se arquiteturas únicas, genéricas o suficiente para acomodar os diferentes serviços e adaptações através de parâmetros simples ou opções predefinidas. Este trabalho, apresenta o modelo de uma arquitetura que viabiliza a aglomeração de diferentes protocolos e serviços; os quais são facilmente modularizados facilitando sua re-engenharia para novos cenários e infra-estruturas telefônicas.

O ponto de partida foi a especificação dos requisitos funcionais e não funcionais que comporiam o modelo. Como requisitos de funcionalidade, o modelo deveria atender aos princípios de monitoração, avaliação e controle do tráfego de rede para condução de uma política de gerenciamento continuado do desempenho do ambiente telefônico.

Para atender a uma função de monitoramento, o modelo deve compor-se de um módulo que viabilizasse a coleta de informações da rede em tempo real. Esse módulo é chamado Módulo de Coleta, e será melhor descrito na seção 5.6. As funções que caracterizam a avaliação do tráfego telefônico estão constituídas em um módulo para esse fim. Por último, o modelo também contém um módulo para permitir a re-configuração de dispositivos da rede ou de infra-estruturas de rádio transmissão.

Esse modelo distribuído deve servir como um *framework* para a integração com diversos outros serviços. Dessa forma, características como: modularidade dos processos, transparência na localização dos serviços, heterogeneidade de linguagem e máquina devem ser considerados requisitos não funcionais para a concepção do modelo.

Segundo [22], um *framework* difere de uma aplicação pelo fato de algumas de suas partes serem adaptadas a novos serviços. As redes telefônicas então, são repletas de serviços para

as mais variadas finalidades gerenciais e operacionais. Esses novos serviços, específicos em sua natureza, dão origem a outros pontos de complementação e modificação que podem ser estendidos em tempo de operação.

Modelos de objetos e arquitetura têm sido cada vez mais tratados de forma integrada pela engenharia de serviços de telecomunicações. Os principais trabalhos baseiam-se no suporte oferecidos por ambientes em plataformas CORBA [5] ou pelo ODP [6]. O modelo aqui faz uso de uma vertente mais nova na conceituação de objetos distribuídos e balanceamento de carga. Este modelo utiliza protocolos comuns à própria Internet para comunicação com a rede WEB mundialmente difundida.

A adaptabilidade é encontrada com mais frequência em trabalhos relacionados a redes inteligentes (IN, do inglês, *Intelligent Networks*) [23], redes programáveis [24, 25] e redes ativas [26, 27]. Comparações entre diferentes propostas de arquiteturas para redes ativas e programáveis foram feitas em [28]. Esta tese não compara quaisquer dessas arquiteturas, mas simplesmente propõe um modelo arquitetural onde a adaptabilidade esteja presente sobre o nível de distribuição das informações e sobre o nível de inteligência inferida sobre os dados. Neste modelo, esta característica do sistema de comunicação é vista sob: (i) a distribuição das funcionalidades para o ambiente interno ou externo à corporação através de protocolos comuns da WEB; (ii) a flexibilidade na substituição do processo inteligente de avaliação dos canais telefônicos, por outros com igual objetivo; e (iii) sob o prisma de uma base de informações distribuída que regula e auxilia o funcionamento de todo o sistema. Um exemplo clássico deste último ponto é a introdução das bases de informação de gerenciamento, segundo padrões já estabelecidos no mercado.

O maior número de serviços telefônicos e, por conseguinte, o volume crescente e o aumento da complexidade das informações manipuladas e contidas na base de dados da rede telefônica, criaram a necessidade de novos modelos e ferramentas para criação, manutenção e manipulação desses serviços.

5.2 - META MODELO PARA SERVIÇOS DE TELECOMUNICAÇÕES ADAPTÁVEIS

Este trabalho parte da adaptação de uma concepção generalizada de um modelo telefônico, com similaridades de [30]. Para a generalização deste modelo, os elementos básicos necessários são: (i) os componentes usuários, que correspondem a entidades que utilizam diretamente os serviços, e (ii) os provedores, que são os responsáveis pelo funcionamento dos serviços durante sua operação. Usuários e provedores são, geralmente, equipamentos e produtos de *hardware* e *software*. Estes compõem o ambiente de oferecimento de serviços a um conjunto de usuários e aos respectivos provedores. A figura 5.1 ilustra um ambiente de oferecimento de serviços com um provedor e quatro usuários.

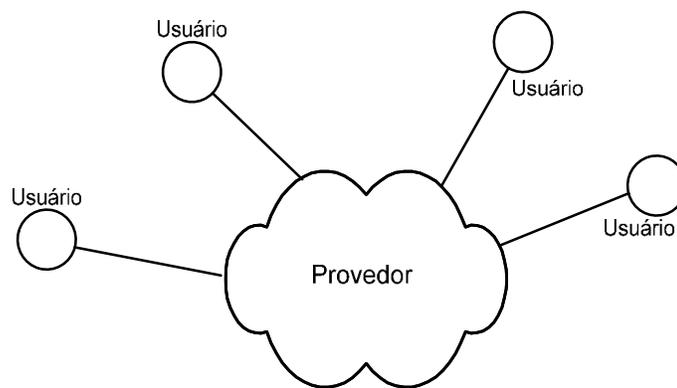


Figura 5.1 Ambiente de oferecimento de serviços

A definição destes serviços de comunicação é abstrata no que diz respeito à dispersão dos usuários. Usuários podem, por exemplo, estar tão próximos quanto os objetos implementados em uma mesma linguagem de programação e ativados em um mesmo processo de um sistema operacional de uma máquina isolada, ou mesmo tão distante quanto objetos localizados em máquinas de sub-redes distintas.

Cada componente tem uma *interface*. Essas *interfaces* definem como e quais são as informações trocadas entre os componentes, estabelecendo o seu comportamento observável perante o provedor de serviços e os outros componentes, i.e., suas formas de cooperação. Neste trabalho, a *interface* adotada entre os componentes de usuário e o provedor de serviço é a de distribuição de serviços WEB.

Para que haja cooperação entre essas *interfaces*, alguma forma de associação deve existir entre os componentes que desejam se comunicar. Segundo [3], essas associações podem ser explícitas ou implícitas.

Nas entidades implícitas, geralmente, não existe uma entidade que represente toda a associação. Aqui, os usuários utilizam, diretamente, alguma identificação dos componentes com quem se comunicam. A partir dessa identificação, em uma comunicação com associação implícita, o provedor é responsável por encaminhar os fluxos de informação aos destinos. Um exemplo de comunicação implícita adotada neste trabalho é a associação existente entre os termos de uma variável lingüística e a própria variável a qual esses termos pertencem. Outra associação implícita pode ser vista entre o valor de um parâmetro qualquer coletado da rede e a amostra que compõe todos esses parâmetros. O primeiro é uma associação no nível do Módulo de Inferência, visto na figura 5.10. O segundo faz parte do Módulo de Coleta, visto na mesma figura. Todos estes módulos serão explicados na seção 5.6; cada um deles, formado por um conjunto de inúmeras associações implícitas.

Nas associações explícitas (figura 5.2), por outro lado, as entidades especiais são criadas a fim de conectar pontos terminais dos fluxos das informações (componentes). Uma associação explícita tem um ponto terminal fonte e um ou mais pontos terminais de destino, correspondendo, a uma abstração ponto-a-multiponto unidirecional denominada *pipe*. Na tese aqui apresentada, os *pipe* estão localizados nas interfaces de cada módulo do modelo proposto. Cada módulo tem como interface uma estrutura de implementação do padrão de projeto denominado *facade* [29] que se comunica com as demais associações internas do mesmo módulo.

Componentes que se comunicam através de associações explícitas são denominados *filtros*. Os filtros são componentes que referenciam diretamente os *pipes* e não a outros componentes, como na associação implícita. Na tese aqui apresentada, os filtros são caracterizados pela implementação através de uma infra-estrutura baseada em serviços WEB. Esses serviços acessam o padrão *facade* que implementam os *pipes* para manipular as associações implícitas.

Aos *pipes* podem ser associadas especificações de QoS utilizadas para definir e regular a forma de comunicação entre os filtros que os utilizam. Os *Pipes* que são associados a especificações de QoS passam a ser denominados *MediaPipes* [30] e poderão ser encontrados no Plano Gerencial apresentado na figura 5.9.

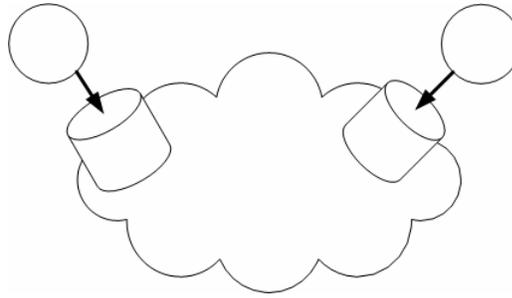


Figura 5.2 Representação gráfica de uma associação explícita entre os filtros.

A implementação de um provedor (servidor) é realizada através da utilização de elementos denominados *componentes de implementação do serviço* (ou *componentes do serviço*), conforme afirma [30]. Para a realização de suas funções, esses componentes comunicam-se entre si e utilizam serviços fornecidos por outros provedores mais primitivos, cada qual denominado *provedor de infra-estrutura do serviço*, conforme ilustrado na figura 5.3. Na tese aqui proposta, um exemplo de provedor de infra-estrutura de serviço é o Componente de Geração de Tráfego Telefônico, embutido dentro do sub-módulo: Modelo de Tráfego (figura 5.10).

Além disso, todo provedor de serviços deverá fornecer uma forma de acesso para que usuários se comuniquem com componentes de implementação desses serviços. Essa forma de comunicação varia conforme a distribuição dos serviços e a linguagem de programação usada. Na tese aqui proposta, esse acesso emprega o protocolo de comunicação SOAP (discutido no capítulo 4 e na seção 5.3.2, adiante) para componentes distribuídos, e chamada a procedimento local para a comunicação entre processos localizados num mesmo dispositivo físico.

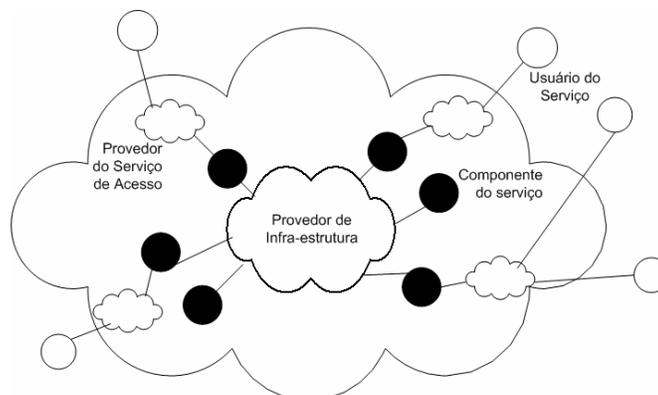


Figura 5.3 Infra-estrutura e acesso a provedores de serviços

Uma das grandes dificuldades encontradas em arquiteturas de QoS esta na diversidade com que os aspectos de qualidade são especificados e, posteriormente, da forma com que eles são abordados nos diferentes sistemas que compõem o ambiente de comunicação e de processamento. A modelagem de serviços de acesso por meio da abstração de um provedor, abstração que é idêntica à utilizada na modelagem da comunicação entre quaisquer usuários independentes de sua dispersão, propõe uma forma de tornar mais homogêneo o tratamento da QoS em ambientes de processamento e de comunicação [3]. Alguns *frameworks* de QoS [31] e *multicast* [32] confirmam essa homogeneidade através de uma modelagem única. A modelagem proposta neste trabalho dá continuidade a essa linha de proposição, onde o modelo aqui adotado pode ser extensível a demais infra-estruturas de telefonia móvel. Ademais, essa característica ganha importância fundamental devido ao rápido processo evolutivo da Engenharia de Serviços de Telecomunicações.

5.3 - A COMUNICAÇÃO EM AMBIENTES DISTRIBUÍDOS

Diversas tecnologias são apropriadas para resolver os problemas de heterogeneidade entre plataformas operacionais e mecanismos de gerência para a distribuição das informações. Dentre eles podemos citar o CORBA [5] e o ISO ODP [6] como referências na distribuição dos objetos de plataformas distribuídas e heterogêneas. Uma abordagem dessas plataformas de componentes distribuídos baseado em ODP é encontrada em [33, 34]. Já esforços na elaboração de plataformas baseadas no padrão CORBA são descritos em [35, 35].

A indústria de sistemas abertos não foi capaz de prover respostas às necessidades crescentes da gerência em ambientes distribuídos corporativos. É necessário um modelo que permita gerenciar a empresa como um todo. Um ambiente corporativo é composto por sistemas heterogêneos produzidos por vários fornecedores. Tais fornecedores operam em sistemas operacionais diferentes, com protocolos diferentes e executam aplicações diferentes escritas com linguagens de programação diferentes. Além disso, as novas tecnologias telefônicas devem ser capazes de integrar e interoperar com as tecnologias antigas.

O problema não é ocasionado pela falta de padrões. Em diversas áreas, existem padrões aceitos pela comunidade. Muitas outras, no entanto, ainda não contemplam implementações. E é obvio ser necessário algo mais do que simplesmente um conjunto de padrões [36].

Como visto, os padrões não resolvem estes problemas por si só. Existe uma quantidade relativamente alta de grupos trabalhando em padrões para a área de gerência. Muitos deles estão trabalhando na mesma área sem que haja qualquer tipo de coordenação entre si. Em virtude disto, aparece uma diversidade de modelos para gerência (ISO, IETF, DMTF, NMF, OSF) [37, 38, 39, 40, 41]. Esta variedade de padrões fez com que alguns fossem mais utilizados em áreas de problemas específicos.

Este trabalho tem por objetivo definir um modelo unificado que facilite o projeto, desenvolvimento e a integração de aplicações de gerência tendo em vista os ambientes complexos existentes nas corporações telefônicas.

Por conta da migração das aplicações para ambiente WEB, alguns sistemas de gerenciamento também têm seguido caminhos similares aos adotados nessa tese, muito embora a grande maioria ainda adote os modelos de gerência anteriormente descritos. Como exemplo, a própria SUN *Microsystem* conta, agora, com um portfólio aberto e gratuito para o gerenciamento de aplicações de Internet usando JAVA [42] sobre a plataforma Solaris. Com esse novo paradigma, a própria SUN poderá vir a adotar o uso de Serviços WEB nessa plataforma, conforme conveniência.

5.3.1 - A Gerência de Recursos aplicada ao modelo

A gerência de recursos por meio da WEB é útil na integração de serviços diversos, administrada por operadores dentro e fora da empresa. Essa gerência pode ser feita por meio de serviços intitulados *WEB Services*, que estão sendo considerados a próxima onda da revolução da internet.

Estes serviços constroem pontes entre componentes de aplicação e entre quaisquer outras plataformas que suportem os mesmos. A diferença entre esses serviços e os componentes tradicionais não está apenas nos protocolos, mas no fato de que esses serviços podem trazer os seus próprios dados e também as funcionalidades "de retaguarda" com ele. Daí a necessidade de cuidar para que seja feita uma boa modelagem desses serviços, a fim de evitar risco de violação desautorizada aos dados. Em geral, o acesso a serviços WEB exige algum processo de autenticação que reduza esses riscos.

Esses serviços usam mecanismos de codificação de dados, como HTTP e XML; que facilita a larga difusão e no uso desses sistemas abertos. O desenvolvimento com base em programação orientada à objetos melhorou a modelagem de elementos do sistema, o encapsulamento dos dados e a funcionalidade desses objetos. Esse desenvolvimento fornece uma estrutura padronizada e rica em serviços, facilmente transformados em aplicativos. Os Serviços Web utilizam protocolos empregados na Internet para tornar as instâncias de componentes facilmente acessíveis dentro e fora da empresa.

Por meio dos Serviços WEB não é preciso se preocupar em como eles são implementados, mas apenas com o uso dos resultados desse serviço. Muitos serviços web têm aparecido, fornecendo variadas escalas de funcionalidade, desde infra-estrutura de baixo nível até serviços corporativos de alto nível. Por tudo isso, eles podem ser vistos como o próximo estágio na revolução dos *softwares* em geral.

5.3.2 - Os Serviços WEB utilizados

Os serviços WEB usam as várias funcionalidades remotas de sistema e os conecta em um serviço único, unificando esses diferentes módulos.

Embora não seja novidade, a comunicação entre serviços tem apresentado desafios difíceis de superar. Um deles é o intercâmbio entre diferentes mecanismos de distribuição, como CORBA, RMI (do inglês, *Remote Method Invocation*) e DCOM (do inglês, *Microsoft Distributed Component Object Model*). Outra dificuldade é a integração com outros aplicativos, como sistemas legados; integração entre fornecedores e integração entre versões. O negócio com base na *web* exige desenvolvimento entre empresas, alta flexibilidade para acomodar uma alta velocidade de mudanças e operação segura através do *firewall* da empresa.

O modelo de serviços WEB é neutro quanto à linguagem e à plataforma; portanto, desenvolvedores de qualquer parte podem construir e utilizar estes serviços. Como os principais fornecedores de aplicativos, plataformas e tecnologias estão adotando o conceito de serviços WEB, esses serviços formarão uma grande parte do desenvolvimento de aplicativo nos próximos anos.

O que garante a heterogeneidade desses serviços é o uso de protocolos baseado na WEB. O protocolo SOAP combina XML com HTTP e cria um formato de empacotamento combinando XML e MIME (do inglês, *Multipurpose Internet Mail Extensions*). Essas mensagens podem ser transportadas por HTTP, SMTP e outros. Atualmente, o SOAP está na versão 1.2.

O WSDL é uma linguagem que descreve serviços WEB em formato XML. Ele define operações, tipos de dados e informações de ligação. Quando um serviço WEB é criado, informações sobre sua interface e localização são armazenadas em um registro. O consumidor do serviço WEB pode recuperar essas informações e usá-las para chamar este serviço. A figura 5.4 apresenta uma abstração dessa interação. Nela, o serviço WEB é a forma de manipulação das funcionalidades do sistema telefônico proposto, através da própria Internet; este serviço é registrado em um *container* (repositório) de serviços da Internet que identifica o aplicativo considerado na arquitetura funcional do modelo telefônico proposto nesta tese.



Figura 5.4 Interação entre serviços WEB, registro e consumo do serviço.

No lado do servidor, um adaptador e um roteador serão exigidos para aceitar as mensagens SOAP que chegam e são distribuídas para implementação do serviço apropriado, conforme demonstra a figura 5.5. Isso equivale ao papel do ORB (do inglês, *Object Request Broker*) no CORBA e na RMI ou do SCM (do inglês, *Service Control Manager*) no DCOM.

Os mecanismos de serviço WEB podem ser usados como uma maneira conveniente para integrar componentes refinados existentes. Esses serviços podem ser usados de modo intei-

ramente dinâmico; nesse caso a descrição do serviço é descoberta e chamada em tempo de execução. Essas informações também podem ser localizadas em tempo de compilação, como parte da descrição do serviço e são ligadas ao aplicativo do cliente para que ele não tenha necessidade do registro em tempo de execução.

5.3.3 - Modelo de uso dos Serviços WEB Orientados a RPC

O RPC (do inglês, *Remote Procedure Call*) feito com base na WEB não difere daqueles feitos sobre o IIOP (do inglês, *Internet Inter-ORB Protocol*), DCOM ou JRMP (do inglês, *Java Remote Method Protocol*). O protocolo SOAP é quem implementa esse mecanismo de RPC com base em XML para objetos distribuídos.

O SOAP usa mecanismos de transporte de propósito geral, padrão aberto de mercado, como HTTP, junto com uma descrição de chamada a métodos baseada em texto do tipo XML.

Uma mensagem SOAP, com base em HTTP, é enviada para um roteador SOAP que processa esse protocolo, assumindo a forma de uma *servlet* HTTP, para implementações em Java. O roteador SOAP examina o cabeçalho e decide a rota adequada de caminho da mensagem.

Existem algumas vantagens no uso de serviços WEB em vez de CORBA, RMI, ou DCOM. A primeira delas é que os protocolos HTTP e XML são abertos e, portanto, não estão ligados a nenhum fornecedor.

É sabido que os demais protocolos têm muita dificuldade em penetrar na maioria dos *firewalls*. Entretanto, o HTTP (e o SMTP) tem direito de acesso geral através da maioria dos *firewalls*, o que torna mais fácil integrar aplicativos entre as organizações.

A figura 5.5 mostra a operação de um roteador SOAP, de acordo com o modelo proposto. Note que a descrição do serviço WEB (WSDL) é usada pelo cliente e pelo servidor para ajudar a determinar o mapeamento correto entre a linguagem adotada e o arquivo XML para chamadas a métodos e tipos de parâmetros.

O HTTP foi originalmente criado para transportar mensagens HTML simples. Ele não fornece a sofisticação exigida para um ambiente de chamadas distribuídas. O uso de mensagens em formato XML estrutura esse ambiente e permite o intercâmbio de clientes e servidores destes serviços oferecidos por diferentes fornecedores.

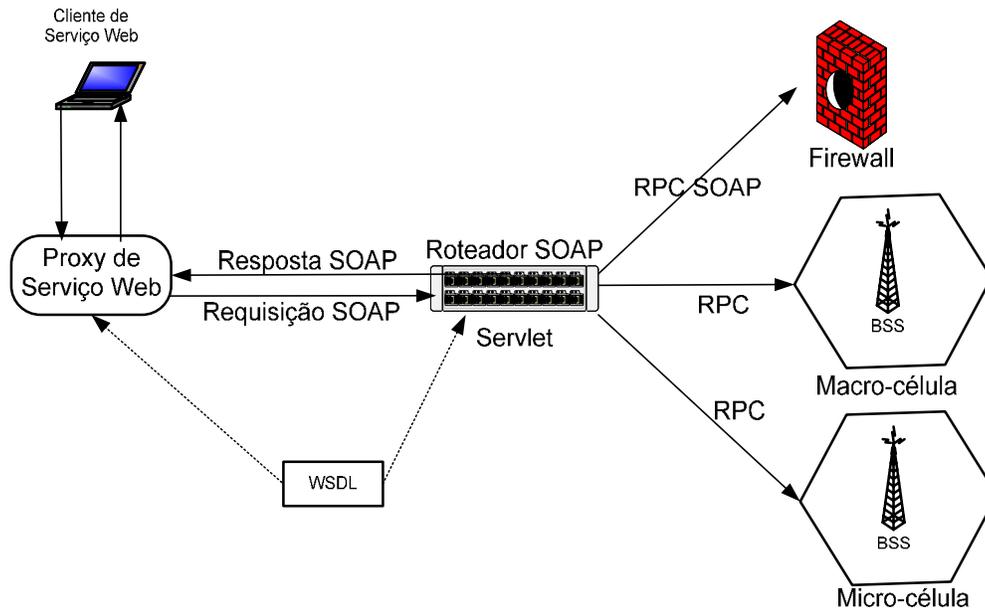


Figura 5.5 Um roteador SOAP com base em Serviços WEB

Do ponto de vista do desenvolvedor, uma das grandes vantagens do protocolo SOAP é a capacidade de penetrar em *firewalls*. Entretanto, do ponto de vista do administrador, isso apresenta os mesmos problemas que o RPC tradicional, ou seja, a capacidade de ter como destino uma chamada à função em um serviço exposto. Embora o SOAP represente uma simples chamada a uma funcionalidade do lado do servidor, deve-se tomar muito cuidado para garantir a segurança adequada aos serviços expostos. O histórico de segurança global dos serviços WEB ainda está em fase de desenvolvimento e o controle transacional e concorrential de vários desses serviços trocando informações entre si ainda é inexistente.

Conforme exposto, a incorporação de serviços WEB para transmissão de informações referentes ao gerenciamento de recursos telefônicos possibilita a heterogeneidade de máquinas e de aplicativos, e a união entre as redes telefônicas e de comunicação de dados, auxiliando na evolução das tecnologias de transmissão móveis celulares.

5.4 - SISTEMA DE DISTRIBUIÇÃO DO GERENCIAMENTO DAS INFORMAÇÕES

Qualquer sistema de gerenciamento deve suportar tarefas de gerenciamento solicitadas pelo provedor do serviço e garantir a interação entre o provedor do serviço e seus clientes e/ou entre outros serviços providos. Preferencialmente, esses sistemas de gerenciamento devem assegurar o uso de padrões abertos para as funções de gerenciamento comuns nessas plataformas. O desenvolvimento para um ambiente como este deve ter, como modelo, o diagrama da figura 5.6.

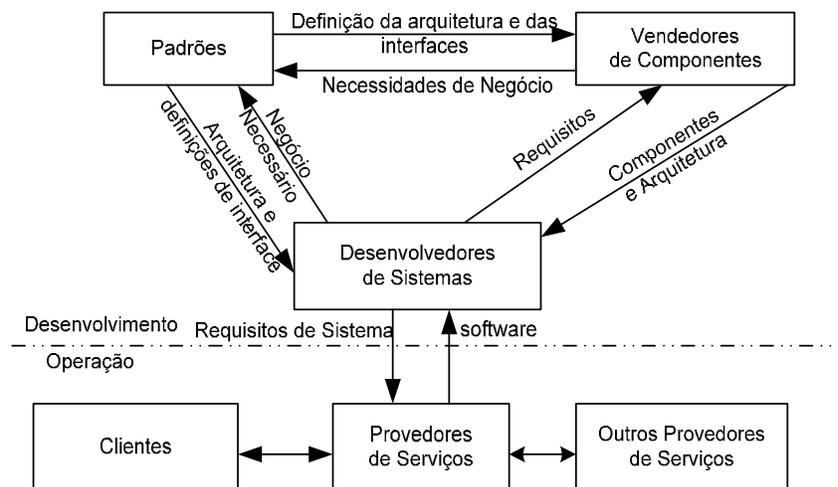


Figura 5.6 Modelo de Negócio para o desenvolvimento de padrões abertos [43]

Ultimamente, fica evidenciado o uso de padrões abertos no reuso de componentes para sistemas abertos, de modo que sejam reduzidas as barreiras na construção de componentes reusáveis, de multi-plataformas e execuções concorrentes. Os requisitos chave para a integração desses componentes são:

- integração com o sistema legado por um custo efetivo;
- interação entre componentes, mesmo que eles tenham sido implementados usando diferentes distribuições ou tecnologias de programação;
- integração entre componentes definidos em diferentes interfaces: IDL, GDMO ou SMI;
- integração de componentes que tornem claras as necessidades de processos de negócio específicos;
- integração com mecanismos que suportem mudanças tecnológicas;

- integração com mecanismos que minimizem os obstáculos para adaptar um componente a uma nova aplicação.

Este trabalho apresenta um modelo (seção 5.6 e apêndices I, II e III) que visa possibilitar a integração entre componentes de diferentes tecnologias de programação; propondo um modelo distribuído para a comunicação entre os diversos módulos que compõe a aplicação. Tal modelo possibilita a distribuição de objetos gerenciados ao longo de uma malha telefônica, garantindo uma comunicação transparente com o administrador. Esta distribuição de objetos gerenciados pode ser vista na seção 5.7.

Algumas tecnologias são adotadas para resolver problemas de distribuição remota de objetos, entre as diversas plataformas operacionais e os mecanismos de gerência para a distribuição das informações em redes de telecomunicações. Dentre elas, foi citada a utilização de Serviços WEB, mas também é empregada uma arquitetura J2EE, como referência nessa distribuição e heterogeneidade de máquina, empregando EJBs como fontes de encapsulamento das funcionalidades dispersas entre vários objetos.

Essa tendência ocorre porque o desafio atual para os vendedores de equipamentos, provedores de serviços e operadores de rede está em como integrar os vários serviços, *softwares* e *hardwares*, de maneira eficiente. A integração entre o TMN (do inglês, *Telecommunications Management Network*) e o CORBA tem sido uma solução. Entretanto, quando esses provedores de serviços são distribuídos numa escala de Internet, os pacotes IOP apresentam dificuldade em atravessar *firewalls*. Por esse motivo, a implementação apresentada no capítulo 6 não faz uso de CORBA.

Ao contrário, os Servidores WEB usam HTTP ou SMTP que é permitido pela maioria dos *firewalls*. Portanto, é válido considerar o uso desses serviços nas grandes aplicações de rede.

Esse trabalho introduz uma arquitetura que faz uso de Servidores WEB no ambiente de TMN e Internet, permitindo que diferentes serviços interajam de maneira conveniente.

A prática no uso de *Enterprise Java Beans* vem focada nas novas tendências da indústria telefônica em criar soluções de suporte operacional a partir da reutilização dos componen-

tes de gerenciamento. Essas soluções devem assegurar a integração com o sistema legado e garantir o fluxo de informações necessárias para satisfazer aos requisitos de negócio.

A tese aqui apresentada contempla a adoção de um modelo componentizado JAVA EJB, apresentada no apêndice B, projetado para rodar dentro do Java *Runtime Environment*, portanto requer uma Máquina Virtual Java. A comunicação entre os vários EJBs é feita localmente, quando eles estão em um mesmo módulo do sistema; ou remotamente, quando em módulos diferentes (figura 5.10). A comunicação deles com outras arquiteturas, todas conectadas de modo a cooperarem entre si, é feita por uma *interface* de serviço WEB, possibilitando um ponto para entrada de mensagens externas ao sistema.

Todos esses recursos de comunicação entre processos estão diagramatizados na figura 5.7. Ela apresenta, segundo os padrões da UML (para a elaboração de Diagramas de Distribuição) [44], uma macro-visão dos nós de processamento em tempo de execução e dos componentes que lá residem. Este diagrama dá uma visão estática do sistema, modelando a topologia de hardware sobre o qual ele opera.

Neste diagrama, os módulos: Controlador de Microcélula, Controlador de Macro célula, Centro de Comutação Móvel, Microcélula ERB e Macro célula ERB estão mostrados dispersos em diferentes máquinas.

Segundo a concepção aqui idealizada, o Controlador de Micro-célula e o Controlador de Macro-célula estão representando os dispositivos instalados na máquina do cliente do serviço tratado, conforme figura 5.5.

O processo de Centro de Comutação Móvel mostra que outras máquinas devem realizar as atividades de roteamento da transmissão das informações solicitadas pelo cliente. Elas estão caracterizadas na figura 5.5 por meio do Roteador SOAP.

Por fim, os processos localizados na Micro-célula ERB e na Macro-célula ERB estão representando as células de transmissão do sistema telefônico e modeladas como Micro-célula e Macro-célula na figura 5.5, respectivamente.

O script de linguagem WSDL, responsável por descrever as funcionalidades do serviço WEB, responde pelas informações trafegadas entre os controladores e o centro de comutação da figura 5.7. A comunicação entre o centro de comutação e as ERBs devem ser feitas por meio de chamadas remotas de procedimento, conforme comparação entre essas duas figuras.

O script de linguagem WSDL empregada nesta tese encontra-se no apêndice I.

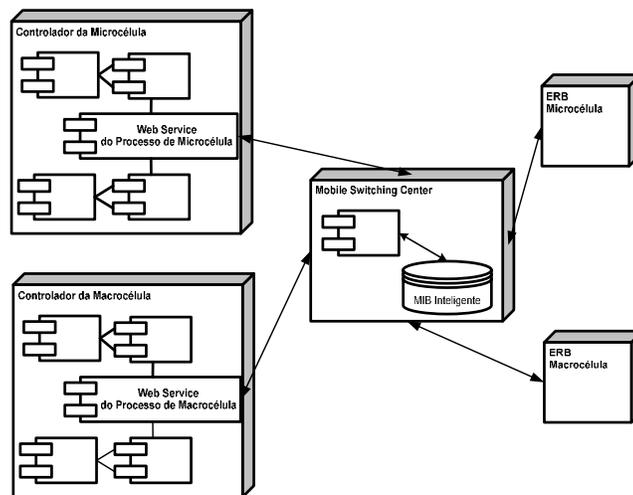


Figura 5.7 Modelo de Distribuição de Processo

5.5 - AMBIENTE DE SOBREPOSIÇÃO DO MODELO DE INTELIGÊNCIA

A figura 5.8, adaptada de [45], apresenta uma conceituação de toda a infra-estrutura de comunicação em redes telefônicas. Nela é mostrado o processo de transferência de informações dos enlaces de rádio para a rede pública de comutação telefônica. Essa arquitetura generaliza toda a estrutura de redes, serviços e suas interfaces, algumas descritas no capítulo 2.

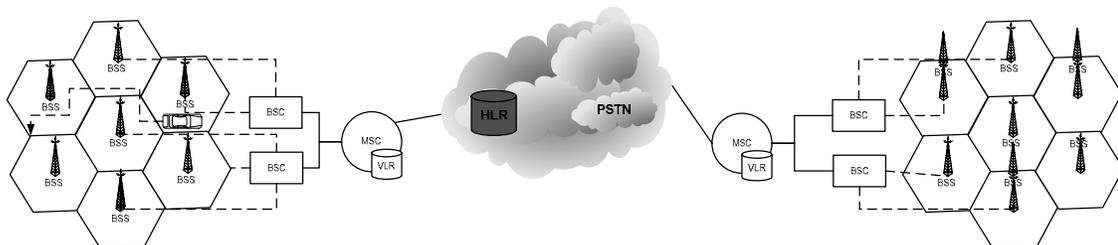


Figura 5.8 Arquitetura das Redes PCS

Considerando que os serviços disponíveis pela rede telefônica são conduzidos pelos vários módulos que a compõe, o tráfego por ele gerado deverá ser monitorado sobre todos esses módulos, ou seja, nas várias interfaces de comunicação. Para isso, é necessária uma plataforma de gerência que administre toda a malha de tráfego; formado tanto por um monitoramento individualmente em cada estação base, quanto monitoramento sobre conjuntos de módulos de transmissão. Aqui chamados de monitoramento distribuído e centralizado, respectivamente.

Isso resulta em uma gerência composta de funcionalidades centralizadas e descentralizadas. Na gerência descentralizada cada estação-base responde (administra) apenas pela conduta de seu tráfego e na gerência centralizada, o comportamento do tráfego deve ser avaliado sobre todo um conjunto de células, seja essa avaliação feita sobre um conjunto de BSCs ou uma MSC qualquer. O módulo de inferência do modelo da figura 5.10 pode ser aplicado sobre esses dois enfoques; ou seja, ela pode ser tanto centralizada, portanto ficar localizada na MSC, ou ela pode ser descentralizada, e possuir alguns módulos instalados em cada BSC. Nesta tese, o modelo de gerenciamento de tráfego distribuído faz uso de ferramentas inteligentes (lógica *fuzzy* e redes neurais artificiais) para a avaliação do tráfego instantâneo que trafega sobre redes PCS.

Para obter um modelo distribuído é necessário instanciar módulos gerenciais sobre os BSCs e/ou os MSCs. É importante que as informações trafeguem seguindo uma estrutura hierárquica, onde as BSCs respondam para o MSC sobre o andamento dos mesmos. Para isso, módulos de coleta (figura 5.10) devem ser instanciados em cada BSC.

Por sua vez, o MSC é composto de comutadores, que se relacionam entre si formando uma grande rede. Nessa rede encontraremos comutadores com função de SGSN, outros com função de GGRS, e outros cujas funcionalidades são de simples roteadores ou servidores de aplicação. Pelo modelo aqui proposto, alguns comutadores devem ser adaptados para o tráfego de protocolo SOAP, referente aos serviços WEB, conforme demonstrou a figura 5.5.

Diante de tantas questões a serem tratadas e observadas pelo *framework*, o mesmo foi avaliado sobre a ótica de um protocolo para aplicação gerencial em redes de telecomunicações.

5.6 - MODELAGEM PROPOSTA PARA OTIMIZAÇÃO INTELIGENTE DE CANAIS

O modelo de abstração apresentado na seção 5.2 foi idealizado para acomodar os requisitos-chaves de integração com outros componentes, conforme seção 5.3 e 5.4. Toda essa vinculação leva à definição de um modelo adaptável (figura 5.10) às mais variadas infra-estruturas telefônicas, de tal forma que este mesmo modelo, deve ser visto sobre a ótica de um conjunto de protocolos de comunicação, conforme figura 5.9.

Experiências com projetos de redes de comunicação demonstram a idéia de se estruturar a rede como um conjunto de camadas hierárquicas, cada uma construída utilizando as funções e serviços oferecidos pelas camadas inferiores. Da mesma forma que o projeto de protocolos de redes de comunicação é mais eficiente se estruturada em níveis, o modelo de aplicação proposto nesta tese está idealizado segundo a organização de níveis apresentada na figura 5.9. Segundo [46], uma vez definida claramente as interfaces entre os diversos níveis, uma alteração na implementação de um nível pode ser realizada sem causar impacto na estrutura global.

O modelo de referência da figura 5.9 é composto por três camadas sobrepostas à rede de telecomunicações, repartidas em dois planos de administração dos serviços apresentados. Dessa forma, uma infinidade de funcionalidades que possam ser incorporadas ao modelo, não deverá impactar nos serviços previamente oferecidos.

As camadas estão assim divididas: Camada de Controle, Camada de Inferência, Camada de Adaptabilidade ao Meio e Camada Física.

A Camada Física está no nível 1 (um), ou nível mais baixo da aplicação. Essa camada representa a infra-estrutura onde a transmissão telefônica se realiza. O capítulo seguinte apresentará um monitoramento feito sobre redes PCS do tipo GSM/GPRS. Seguindo essa

concepção de modelagem, outras redes (como por exemplo, IS-95B e W-CDMA, apresentadas no capítulo 2) podem ser monitoradas e avaliadas segundo este modelo de abstração.

A Camada de Adaptabilidade ao Meio (nível 2) interage diretamente sobre a malha de infra-estrutura telefônica, oferecendo as informações necessárias à Camada de Inferência para o tratamento da aplicação. Essa camada é dividida nas seguintes subcamadas: Subcamada Dependente do Meio Físico e Subcamada de Filtragem de Informações.

A Subcamada Dependente do Meio Físico faz a coleta de todas as informações gerenciais disponíveis na malha telefônica. Nessa camada são instalados serviços para coleta de objetos gerenciados das interfaces de redes telefônicas, em modo promíscuo.

A Subcamada de Filtragem de Informações estabelece a filtragem (ou convergência) das informações necessárias para o monitoramento em questão. Nesta tese, foi feito monitoramento sobre as interfaces de rede referentes à Gerência de Desempenho. Por meio dessa subcamada, outras interfaces de rede poderiam ser filtradas, como por exemplo, as interfaces para monitoramento dos objetos de segurança dessas mesmas redes, em um ambiente real de telefonia.

A Camada de Inferência (nível 3) é responsável por serviços de análise e avaliação do tráfego. Nesta camada, são feitas inferências sobre os valores dos objetos coletados que caracterizam os atributos monitorados da rede.

De acordo com as conclusões da inferência do nível 3, o administrador pode estabelecer tomadas de decisões que automatizem a alteração sobre alguns dos objetos monitorados. Esse controle, possibilitando a administração direta do administrador, está na Camada de Controle (nível 4). Nela, é possível criar ferramentas de automatização para o monitoramento e controle da rede telefônica.

Além dessas quatro camadas, dois planos de administração foram idealizados: Plano do Usuário e Plano Gerencial. Isso permite a construção de restrições de acessibilidade aos atributos da rede. O Plano do Usuário permite a administração das questões de monitoramento, avaliação e controle da rede. Por este plano, o administrador deve ter acesso direto, apenas, manipulando as ferramentas disponíveis na Camada de Controle.

Diferentemente do Plano do Usuário, o Plano Gerencial deve permitir a reprogramação de serviços disponíveis nas camadas inferiores. Essa conceituação permite a implantação de nós programáveis sobre a arquitetura proposta nesse trabalho. Abordagens de gerenciamento em redes programáveis foram usadas em [47, 24].

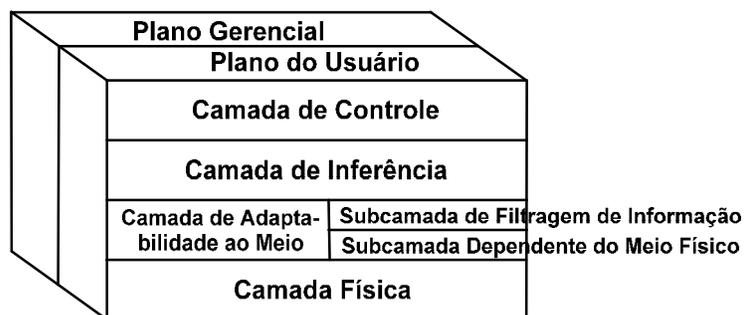


Figura 5.9 Protocolo de Telefonia Proposto

Conforme citado, este trabalho apresenta um ambiente de controle e monitoramento do tráfego de alocação de canais em redes telefônicas, cujas características são:

- 1) o uso de ferramentas de inteligência artificial na gerência da alocação dinâmica de canais;
- 2) monitoramento de informações de tráfego de dados e de voz, para a inicialização de novas chamadas iniciadas e de *handover* compartilhado entre elas; e,
- 3) o uso de paradigmas de distribuição de informações gerenciais inferidas a partir das ferramentas de inteligência e dos atributos dos objetos das células gerenciadas. Como paradigma de distribuição, foi adotado o conceito de serviços WEB [48] em conjunto com princípios do armazenamento de objetos gerenciados SNMP (do inglês, *Simple Network Management Protocol*) [49, 50, 51, 52, 53, 54, 55].

A figura 5.10 apresenta o modelo em questão. Tendo-o como base, o tráfego de *downlink* é otimizado por meio de técnicas inteligentes que possibilitam a alocação dinâmica dos canais de comunicação; de forma a garantir o maior volume de tráfego possível com a menor perda de informações.

O Modelo apresentado pode ser separado em três módulos fundamentais:

- módulo de coleta;
- módulo de inferência; e,
- módulo de configuração.

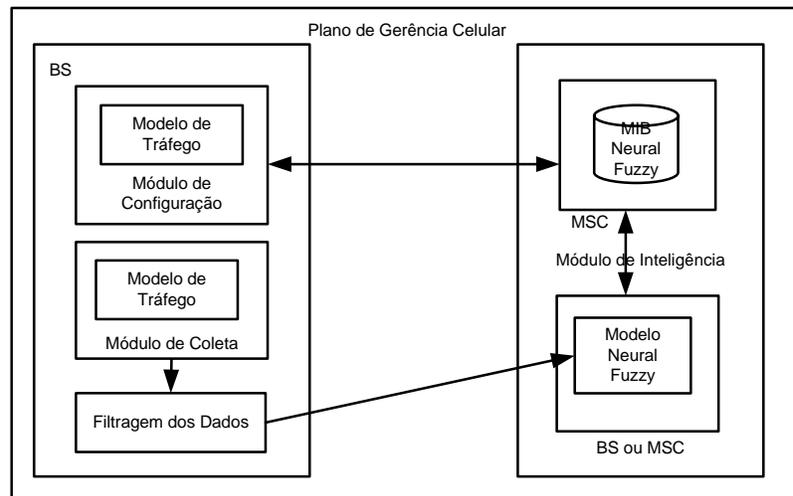


Fig 5.10. Modelo proposto para Alocação Dinâmica e Inteligente de Canais

O **Módulo de Coleta** é responsável pela coleta das informações que caracterizam o tráfego celular. Essas informações servem como parâmetros para o estudo do comportamento da rede que deve ser mapeado para o **Módulo de Inteligência** (ou **Inferência**). As informações dos atributos do módulo de coleta são:

- quantidade de canais totais disponíveis;
- capacidade de tráfego oferecido (em *Erlangs*);
- percentual de bloqueio de novas chamadas telefônicas;
- percentual de requisições de *handovers* falhos;
- tipo de esquema de reserva de canais (pré ou pós reserva);
- forma de empilhamento das chamadas de *handover* (FIFO, MBP ou nenhum);
- quantidade de canais sub-taxados;
- capacidade de tráfego efetivamente aproveitada.

Este módulo de coleta está conceitualmente mapeado, diretamente, na Camada de Adaptabilidade ao Meio. A Subcamada Dependente do Meio Físico faz a coleta de todos os atributos de rede acima citados, enquanto que a Subcamada de Filtragem de Informação passa para a camada superior, apenas os atributos que efetivamente farão parte de alguma(s) regra(s) de inferência. Para a implementação discutida no capítulo 5, todos os atributos acima descritos estão sendo manipulados pela Camada de Inferência.

Essas informações das estações base são, então, encaminhadas ao **Módulo de Inteligência**, por meio de um ambiente de distribuição que dispara a execução das ferramentas de inteligência utilizadas. Estas ferramentas funcionam como serviços de rede instalados na Camada de Inferência da figura 5.9.

O **Módulo de Inteligência** (ou de **Inferência**) é atualmente composto por duas ferramentas de avaliação de tráfego. A primeira delas é um controlador *fuzzy* cujo conjunto de regras está no apêndice D. A segunda refere-se a uma rede neural, cujo modelo está representado na figura 5.11. De acordo com a estrutura em que essa arquitetura está sendo proposta, outras ferramentas (serviços) podem vir a ser adotadas nesse módulo ou camada de protocolo; como exemplo, o uso de meta-heurísticas do tipo Algoritmos Genéticos para análise dessas informações de tráfego.

A primeira característica observada é quanto à quantidade percentual de tráfego de novas chamadas e de *handovers* ocorridos durante o intervalo de tempo entre coletas consecutivas. Em geral, a quantidade percentual de novas chamadas tende a ser superior as requisições de *handover*, sob pena de, em caso contrário, ter ocorrido um mau dimensionamento da malha telefônica.

Conforme será visto no capítulo 6, é necessária a reserva de canais de dados, canais exclusivos para *handover*, sinalização e outros compartilhados entre os vários tráfegos. Neste trabalho, foi realizado estudo abordando a configuração dinâmica dos mesmos; podendo, num momento um dado canal ser utilizado para um tipo qualquer de tráfego e em outro momento posterior, para outro tipo.

Por meio da estrutura de controle *fuzzy*, procurou-se fazer com que os canais chegassem a um perfil de comportamento onde o tráfego desperdiçado por meio do bloqueio de novas

chamadas ou de falhas na realização de *handover* fosse o menor possível. Nesse caso, o controlador *fuzzy* ficou responsável por gerar uma curva de maximização do tráfego com redução dos erros.

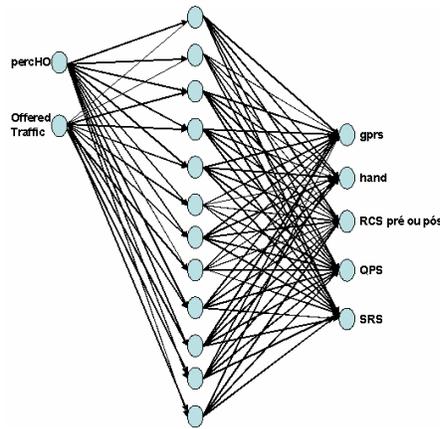


Fig 5.11. Rede Neural adotada

Este trabalho também adota uma rede neural perceptron multi-camadas com algoritmo de treinamento *feed-forward backpropagation*. Essa rede é composta por 2 neurônios de entrada, 12 neurônios intermediários e 5 neurônios de saída. A figura 5.11 apresenta a estrutura da rede neural utilizada.

O **Módulo de Configuração** é responsável por adotar medidas que visem alterar o padrão de configuração dos canais, buscando adequá-los à nova conjuntura de tráfego existente. O foco principal adotado aqui é a necessidade de garantir que a capacidade de tráfego suportada (*carried traffic*), definida pela equação 6.8, possa minimizar as perdas relativas de bloqueio de novas chamadas e a recusa de novas requisições de *handovers*, sem inviabilizar a transmissão de mensagens puramente de dados. Para garantir a mínima condução do tráfego de dados, foi utilizada uma quantidade mínima de canais alocados, para qualquer que seja o volume de tráfego de voz. Este módulo está intimamente vinculado à Camada de Controle na figura 5.9.

O treinamento da Rede Neural, feito no **Módulo de Inteligência**, permite que os seus neurônios de saída possam informar quantitativamente os valores referentes ao novo padrão celular, para:

- quantidade de canais de dados;
- quantidade de canais exclusivos para *handovers*;
- esquema de reserva de canais para *handovers*, do tipo pré ou pós reserva;
- esquema de uso de empilhamento de chamadas de *handover*;
- quantidade de canais sub-taxadas para aumentar a capacidade de tratamento de *handovers*.

Esse Módulo de Inteligência tem a particularidade de adaptar-se facilmente a qualquer alteração nas regras de inferência, sejam elas regras referentes à configuração da estrutura da rede neural apresentada no apêndice E desta tese, ou à construção do controlador *fuzzy*, presente no apêndice D.

Essa adaptabilidade é fruto da construção dinâmica das regras *fuzzy* e da estrutura da rede neural, uma vez que essas informações estão armazenadas na MIB Inteligente definida e simulada neste trabalho. Essa MIB adota os fundamentos de padronização na construção de MIBs em ambiente de gerenciamento de redes TCP/IP [54].

5.7 - BASE DE INFORMAÇÕES DE GERENCIAMENTO (MIB) DO MODELO TELEFÔNICO

A Base de Informações de Gerenciamento (MIB) é um modelo que define precisamente as informações que estão acessíveis utilizando um protocolo de gerenciamento. Usando uma estrutura hierárquica, conforme visto em resumo na figura 5.12 e com mais detalhes no apêndice F, a MIB modela através de variáveis cada um dos dispositivos gerenciáveis da rede.

A MIB é definida segundo um conjunto de regras chamado SMI (do inglês, *Structure of Management Information*), definida em [52] e [53]. O SMI define como as informações de gerência são agrupadas e nomeadas, as operações e tipos de dados permitidos e a sintaxe para a definição das MIBs.

No SNMP um nó gerenciável é representado por meio da linguagem ASN.1 (do inglês, *Abstract Syntax Notation 1*) cuja função é:

- definir as informações trocadas pelo protocolo de gerência;
- definir as estruturas de dados que representam os nós gerenciados;

Em essência, um modelo de informação é composto pelos seguintes elementos [55]:

- Modelo para a estruturação da informação;
- Modelo para representação de nomes;
- Modelo para representação dos dados;
- Modelo para representação de relacionamentos.

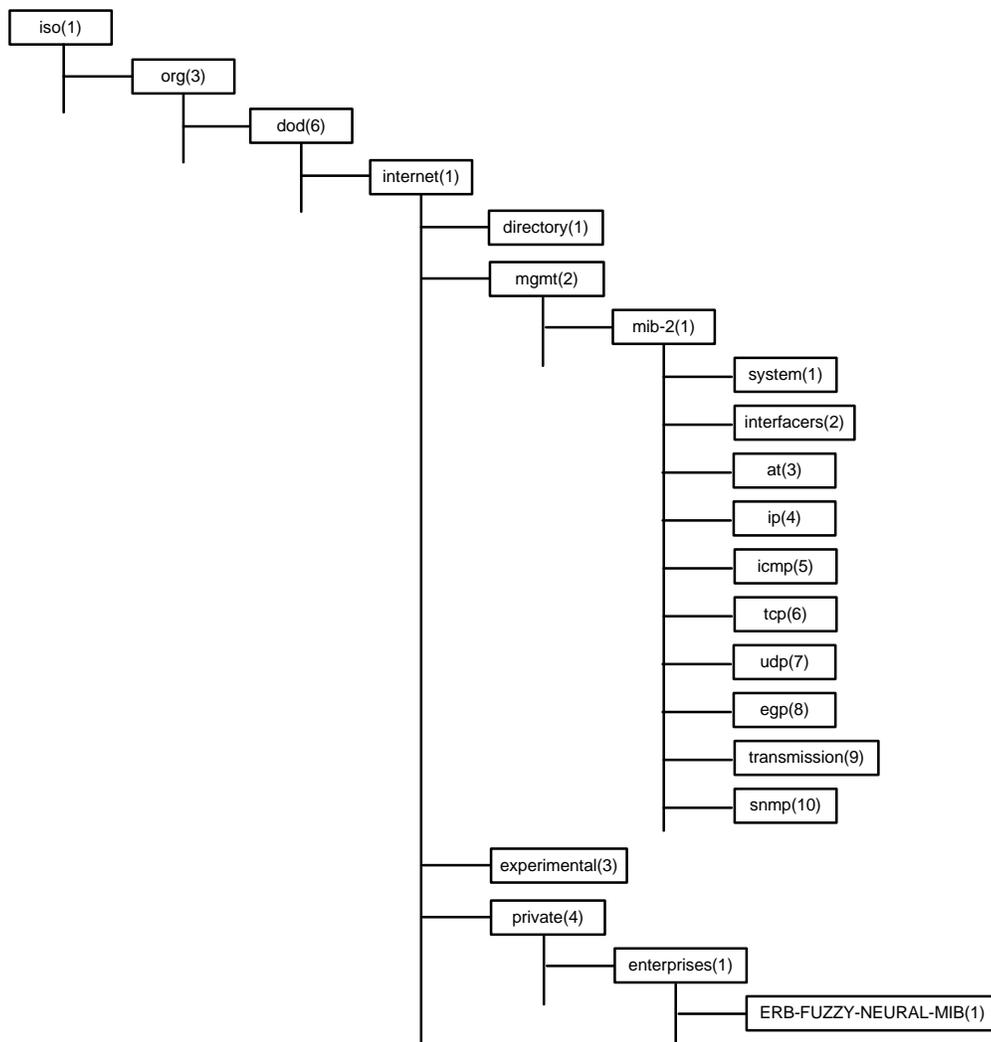


Figura 5.12 Representação da MIB em árvore

As MIBs apresentam três desses modelos de representação e o gerente se encarrega de implementar o modelo de estruturação da informação. Neste contexto, o Modelo de Estruturação da Informação busca proporcionar uma organização na forma de como a informação está disposta. Já os nomes dos objetos identificam a origem e o destino das operações de gerência e das notificações. Com isto é importante que o nome identifique de modo distinto cada um dos recursos que fazem parte do sistema.

A gerência OSI e Internet utilizam a ASN.1 como forma de definição dos tipos de dados associados com os atributos e parâmetros. A gerência OSI além da ASN.1, ainda utiliza uma sintaxe para a organização das classes dos objetos, descritas em Guidelines for Definition of Managed Objects – GDMO [56]. GDMO é a linguagem de *templates* utilizada para definir as características de objetos gerenciados.

A grande vantagem no uso de ASN.1 é a não dependência de plataforma. Em ASN.1 as informações são distribuídas segundo um modelo baseado em árvore, onde cada objeto contém as seguintes informações:

- Um OID (identificador de objetos);
- Um texto curto que define este objeto.

Identificadores de Objetos (OIDs) são uma seqüência de inteiros, não negativos, separados por um ponto, representando a posição do objeto na hierarquia da árvore [51]. OIDs são escritos em uma das formas seguintes:

Sintaxe:

“{ { { <nome> [(“ <número> ”)] } | [<nome> (“ <número> ”)] ... }”

ou

<número> [“.” <número>] ...

Exemplo:

{ iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) ERB-FUZZY-NEURAL-
MIB(1) }

ou 1.3.6.1.4.1.1

Um objeto dessa MIB é uma abstração de uma característica da rede telefônica ou das ferramentas de inteligência, com sintaxe e semântica própria. A sintaxe define uma instância do objeto enquanto que a semântica define o comportamento do objeto naquele momento. Os objetos são estáticos, compilados a partir de uma linguagem de descrição para um formato binário que agentes e aplicações de gerência podem compreender. Um objeto instanciado é também conhecido como variável. O valor de uma dessas variáveis é a menor unidade que apresenta alguma informação gerenciável.

A construção de objetos, de forma similar, foi anteriormente proposto por [57, 58, 59] e [60]. A diferença está no fato que [57, 58, 59] faz essa abordagem sobre o gerenciamento remoto (MIB RMON) de redes TCP/IP; enquanto [60] enfatiza o gerenciamento sobre redes ATM.

As ERBs a serem gerenciadas e as ferramentas de inteligência utilizadas são modeladas como um objeto e armazenadas na MIB. A idéia é ter para cada ERB gerenciável, uma MIB que será uma abstração desta, juntamente com as ferramentas de monitoramento da mesma. Por exemplo, as informações do tráfego de uma ERB espelham seus valores na MIB e, por ser este um padrão de mercado, essas informações podem ser manipuladas por uma infinidade de plataformas de gerência.

O apêndice F apresenta toda a diagramatização, em árvore, da estruturação dessa MIB; além do script de descrição da MIB de acordo com a SMIV2.

5.8 - CONSIDERAÇÕES

Nos últimos anos, tem sido impressionante o desenvolvimento dos sistemas de comunicação móveis. Vários serviços têm sido ofertados e em tempos de transmissão menores. Além da voz, os sistemas atuais permitem a troca de mensagens curtas e acesso à Internet. Exemplo disso são os sistemas GSM e IS-95 [61], comentados no capítulo 2.

Enquanto isso, as futuras gerações oferecerão serviços de faixa larga, como a multimídia. Nele, será possível a integração de serviços de voz, dados, vídeo, multimídia e acesso à Internet [62, 63, 64], cuja integração voz e dados caracteriza o foco desse trabalho.

Grandes esforços para melhorar o desempenho das redes sem fio têm sido despendidos por causa da demanda crescente de largura de banda e o aumento no número de usuários. Nesse contexto, os serviços de voz e de Internet possuem requisitos diferenciados. Enquanto o serviço de voz tem igual largura de banda para *downlink* e *uplink*; a Internet apresenta tráfego assimétrico, ou seja, larguras de banda diferentes para os tráfegos de *uplink* e *downlink*.

Uma das características do tráfego nos sistemas móveis é o fato dele ser *full duplex*, ou seja, comunicação nos dois sentidos. São duas as técnicas de duplexação empregadas nos sistemas telefônicos: FDD e TDD. [65, 66, 67, 68].

Preocupados com essa diferenciação entre os serviços, surge a seguinte pergunta: Como desenvolver um CAC (Controle de Admissão de Chamadas) para sistemas de comunicação móveis multi-serviços com suporte a tráfego simétrico e assimétrico? Os trabalhos a seguir tentam responder essa questão. [69, 65, 66, 70, 71, 63, 72, 73, 74, 75, 76].

Sistemas Móveis como: AMPS, GSM, IS-95, etc.; são usados para carregar serviços simétricos, como telefonia, uma vez que a largura de banda do *uplink* e *downlink* não podem ser modificadas facilmente. FTP e navegação WEB são exemplos de tráfego assimétrico.

Tráfego assimétrico, suportado em sistemas que são unicamente FDD, resultam em desperdício de muita largura de banda se o tráfego *uplink* ou *downlink* forem expressivos [77]. Para esses casos, o TDD é melhor para tráfego assimétrico, uma vez que a largura de banda ou os *slots* de tempo atribuídos para *uplink* e *downlink* podem ser modificados mais facilmente.

A alocação desses canais TDDs seguem o uso de diferentes técnicas que podem ser combinadas entre si. A combinação dessas técnicas é o foco do capítulo 6.

6 - SIMULAÇÕES E IMPLEMENTAÇÃO COMPUTACIONAL DO MODELO

Com a introdução das Redes Digitais de Telefonia Celular (2ª geração), por volta dos anos 90, o número de usuários cresceu enormemente, aumentando a concorrência no mercado e levando ao surgimento de novos padrões. Isso criou uma série de incompatibilidades entre os sistemas; algo que estão sendo gradualmente resolvidas por meio da adaptação de protocolos que garantam a conversação entre diferentes plataformas.

Um maior número de usuários requer a adição de novas frequências de transmissão. Como, muito provavelmente, a operadora não disporá de nova largura de banda para a alocação adicional de dados e voz na malha telefônica, o espaço de frequência existente deverá ser replanejado. Por essa razão, este trabalho implementa a Técnica de Compartilhamento Parcial para alocação de canais. Técnica esta que já foi vista na seção 3.1.

Este trabalho focaliza as redes PCS de 2½ Geração, em que duas categorias de transmissão são evidenciadas: transmissão de voz e de dados. Para a transmissão de voz, são vistas duas modalidades de estabelecimento de comunicação: o estabelecimento de novas chamadas e a realização de *handovers*.

Do ponto de vista do usuário, na falta de recursos, é mais interessante o não estabelecimento de novas chamadas, que a interrupção daquelas em curso. Essa prioridade deve ser levada em conta, principalmente na alocação de recursos em áreas de células relativamente pequenas (micro-células), onde o número de *handovers* é muito maior. Por isso, nesse capítulo são apresentados alguns resultados de monitoramento dessas duas modalidades de tráfego. O comportamento desses tráfegos foi estudado para diferentes valores de:

- alocação de canais de dados;
- alocação de canais de *handover*;
- escolha do critério para reserva de canais de *handover*;
- empilhamento de chamadas de *handover*; e,
- sub-taxação desses canais de *handover*.

A alocação de canais de dados é fundamental para a transmissão contínua de tráfego de Internet. O estudo de caso desse trabalho foi feito sobre uma tecnologia GSM/GPRS de transmissão, portanto 2½ G. A ausência de canais de dados alocados na ERB de transmissão faz a rede ter características de GSM convencional, ou seja, 2 G, apenas.

A alocação de canais de *handover*, já discutida no capítulo 3 e estudada também por [78], torna a rede sensível às necessidades de realização de mobilidade dos dispositivos móveis. Por ser extremamente desconfortável uma interrupção brusca desses serviços, a rede os prioriza em realização ao estabelecimento das novas chamadas. Os critérios para a alocação de canais de *handover* podem ser: pré-reserva, ou pós-reserva de canais. A pré-reserva faz com que a simulação aloque transmissões com características de *handover*, nos canais previamente reservados. Outra opção é a pós-reserva, nela os canais do conjunto reservado para esse grupo são alocados apenas se todos os canais compartilhados estiverem momentaneamente ocupados.

O trabalho [79] propõe um esquema de priorização auto-adaptativo para canais reservados e canais dinamicamente alocados; úteis para as distribuições de tráfego uniforme e não-uniforme.

Em [80] é estudada a reserva de canais para *handover* e para o estabelecimento de novas chamadas também. Assim como [81], ele também estuda o empilhamento de ambos os tráfegos.

Diferentemente de [80], [81] usa uma única pilha para compartilhar todos os tipos de tráfegos. Quando a ocupação da pilha excede certo limite para um determinado tipo de tráfego, novas chamadas para aquele tipo são bloqueadas.

A simulação apresentada nesse capítulo estuda o empilhamento sobre o tráfego de *handover*.

Tradicionalmente, um dentre dois tipos de sistemas de *handover* pode ser empregado [7]:

- *Soft handover*: assegura que o link está configurado para a estação base na nova célula antes que a anterior seja perdida.

- *Hard handover*: exige que o telefone interrompa a conexão com uma estação base antes de se conectar à outra. Isso é menos confiável que o *soft handover* pois o telefone nem sempre será capaz de estabelecer nova conexão.

Estudos sobre a modelagem de *hard handover* são encontrados em [81, 82, 80, 78, 79, 83, 84]. Enquanto isso, [85, 86, 87] estudam o efeito do *soft handover* sobre o desempenho da rede. O simulador usado neste trabalho não entra nesses detalhes de sinalização de chamadas, atendo-se apenas ao uso de um modelo matemático de distribuição normal para definir o tempo de permanência de uma chamada na região de *handover* e a análise dinâmica destes tráfegos sobre o comportamento da carga da rede.

O empilhamento de *handovers* só é vantajoso se houver uma região de sobreposição (*overlapping*) entre as células adjacentes. Uma chamada na região de *handovers* é forçosamente terminada se não for possível obter os canais desejados antes de deixar esta região [82, 87]. Essa sobreposição é chamada: região de *handovers*. Este capítulo apresenta estudos com diferentes tipos de empilhamentos, aos quais são estudados: empilhamento usando a técnica FIFO e empilhamento MBP que usa a mensuração do sinal transmitido, como forma de decisão sobre a priorização das chamadas em espera. Empilhamento MBP é estudado com mais detalhes em [84] que propõe um esquema dinâmico para a mensuração do sinal.

Assim como neste trabalho, [82] também usa empilhamento de *handover* para reduzir a probabilidade de perdas (do inglês, *forced termination probability*).

O artigo [83] estuda diferentes aspectos do gerenciamento de mobilidade, considerando diferentes velocidades de pedestres e de veículos. A simulação apresentada nesse capítulo, não entra em detalhes de velocidade dos dispositivos móveis, a velocidade do dispositivo em questão é definida por uma distribuição normal de probabilidade, onde a média de velocidade foi definida em 30 km/h com desvio padrão de 20 km/h.

Os trabalhos [85, 86, 87] estudam apenas tráfego simétrico. Neste trabalho, foi analisado o número de canais de dados que devem ser alocados conforme uma variação da rede. Estas informações são tipicamente assimétricas.

Enquanto este trabalho aborda as redes GSM/GPRS, [86] propõe modelos de tráfego para redes DS-CDMA com *soft handovers*. E [70, 71, 63, 72, 73, 74, 75, 76] conduzem estudos em vários aspectos dos sistemas CDMA/TDD. Ex: controle de potência e controle de acesso ao médio. Eles consideram o efeito do tráfego assimétrico, mas não da mobilidade. Também não consideram o efeito do *soft handover* e do empilhamento.

Como citado anteriormente, a demanda crescente de usuários na malha telefônica exige a redução de desperdícios na utilização do espectro de frequência. Esta preocupação é um problema de otimização dinâmica com restrições em grande escala em um ambiente estocástico.

Para solução desses problemas de alocação de canais, os métodos mais adequados são: técnicas como a aprendizagem por esforço [88, 89], algoritmos evolucionários [90] e busca por redes neurais artificiais [91].

Abaixo seguem os projetos de construção das duas técnicas de inteligência, abordadas no escopo desse trabalho: lógica *fuzzy* e redes neurais artificiais.

6.1 PROJETO DO SISTEMA FUZZY

Sistemas *fuzzy* são ferramentas que usam procedimentos de conhecimento e inferência para resolver problemas que são difíceis de resolver sem o auxílio de um especialista. Esses sistemas podem ser pensados como modelos de especificação das melhores práticas de solução [92].

Neles, são usados conjuntos *fuzzy* [93] que constituem uma fonte de abstração matemática bastante adequada para a representação de conceitos, cuja fonte de imprecisão advém da ausência de limites de abrangência bem definidos, ao invés, por exemplo, de fenômenos estatísticos. Esses conjuntos viabilizam a representação e manipulação de conceitos que, como neste trabalho, não apresentam contornos bem definidos, e cuja aplicabilidade a um

dado estado é sempre uma questão de gradação. Desta forma, é possível criar-se modelos computáveis para um modo de raciocínio aproximado, semelhante ao dos seres humanos; tornando acessível á máquina formas de conhecimentos que, embora sejam amplamente aplicados no cotidiano das pessoas, apresentam dificuldades para uma formalização matemática exata.

No projeto do sistema *fuzzy*, existem alguns pontos importantes que foram seguidos na realização deste trabalho. A metodologia adotada foi extraída de [94] e [57].

1. Seleção das variáveis de entrada e saída;
2. Quantificação do universo de discurso das variáveis de entrada e saída;
3. Definição das funções de pertinência;
4. Composição das regras de inferência;
5. Discretização do universo de entrada e saída;
6. Composição dos vetores da função de pertinência;
7. Seleção do tipo da inferência ($A \square B$);
8. Implementação do motor de inferência *fuzzy*.

Para seleção das variáveis de entrada e saída do sistema *fuzzy*, foram utilizadas as variáveis (atributos) de tráfego que modelam o sistema de simulação do tráfego GSM/GPRS. Essa simulação foi feita a partir do gerador de tráfego disponibilizado por [12]. Os atributos usados são:

- tráfego ofertado (em Erlangs);
- quantidade de canais disponíveis para dados;
- quantidade de canais disponíveis para *handover*;
- quantidade de canais sub-taxados;
- tráfego simulado de *handover* (em percentagem);
- esquema de reserva de canais: pré-reserva ou pós-reserva; e,
- esquema de empilhamento de canais: FIFO ou MBP.

A quantificação do universo de discurso desses atributos foi feita através de processos de simulação. Dois cenários foram simulados: uma micro-célula isoladamente e uma micro-

célula sobreposta sobre uma macro-célula. Com base nessas simulações verificou-se o intervalo de valoração de cada um desses atributos.

A partir de então, foram construídas as variáveis lingüísticas que representam cada um desses atributos de rede. Essas variáveis e a definição de suas funções de pertinências são encontradas no apêndice D.

Em seguida, essas variáveis foram compostas em regras de inferência adaptadas a partir do conhecimento prévio do especialista. A composição dessas regras foi codificada através de proposições condicionais na forma: Se... Então. Assim, foi definido um conjunto de oito regras compostas na forma de um controlador, conforme visto no apêndice.

Para a definição das regras de composição, foram aplicadas operações de AND sobre os termos lingüísticos das variáveis lingüísticas de entrada: percentual de *handover* e tráfego ofertado. Como elementos de saída do sistema, os valores dos seguintes elementos são retornados e formarão a próxima configuração otimizada do modelo.

- quantidade de canais de dados alocados;
- quantidade de canais de *handover* alocados;
- esquema de reserva de canais adotado;
- esquema de empilhamento adotado; e,
- quantidade de canais sub-taxados alocados.

O próprio processo de composição se dá sobre a manipulação de valores lingüísticos discretizados no intervalo [0, 1].

A composição dos vetores de pertinência está armazenada nos elementos da MIB (apêndice F) e são carregados para o módulo de inferência, obtidos em tempo de execução para participar da construção dos termos e variáveis lingüísticas, das regras e do controlador manipulados no processo de avaliação da rede.

O tipo de implicação utilizado no processo de inferência *fuzzy* foi o Método de Implicação de Mandani, através da inferência de *Modus Ponens* por ser de implementação simples e utilizado na maioria dos trabalhos pesquisados. Pela mesma razão, o Método de Defuzifi-

cação das funções de pertinência usado foi o Método do Centro de Gravidade. Independentemente dos métodos utilizados, o nível de interação entre as regras de inferência depende do grau de sobreposição das funções definidas [57].

A última etapa do Projeto de um Sistema *Fuzzy* é a implementação de um motor de inferência. A técnica usada é explicada a seguir.

6.1.1 Motor de Inferência

O Motor de Inferência *Fuzzy* é formado pela agregação de um conjunto de relações lingüísticas estruturadas, como regras de inferência na forma da equação (6.1). Nela, os valores $x_1, x_2, y_1, y_2, y_3, y_4$ e y_5 são valores numéricos do universo de discurso das variáveis lingüísticas representadas pelos termos lingüísticos expressos por $A_1, A_2, B_1, B_2, B_3, B_4$ e B_5 , respectivamente.

$$\text{Se } (x_1 \text{ é } A_1) \text{ e } (x_2 \text{ é } A_2) \text{ então } (y_1 \text{ é } B_1) \text{ e } (y_2 \text{ é } B_2) \text{ e } (y_3 \text{ é } B_3) \text{ e } (y_4 \text{ é } B_4) \text{ e } (y_5 \text{ é } B_5) \quad (6.1)$$

Este sistema *fuzzy* mapeia regiões no espaço das entradas $U_1 \times U_2$ para regiões no espaço das saídas $V_1 \times V_2 \times V_3 \times V_4 \times V_5$, através da aplicação de regras de inferência [95].

O sistema da figura 6.1 mapeia regiões do espaço R^2 para o espaço R^5 . As entradas A' e B' interagem com as 5 regras, representadas por (A_i, B_i) , produzindo 5 conjuntos *fuzzy* C_i' , devido ao grau de interação da entrada com as respectivas regras. O conjunto *fuzzy* de saída C' é resultado da regra composicional de inferência (equação 6.2) que, ao aplicar o método do centro de gravidade (equação 6.3), resulta no valor defuzificado $(w^0, w^1, w^2, w^3, w^4)$.

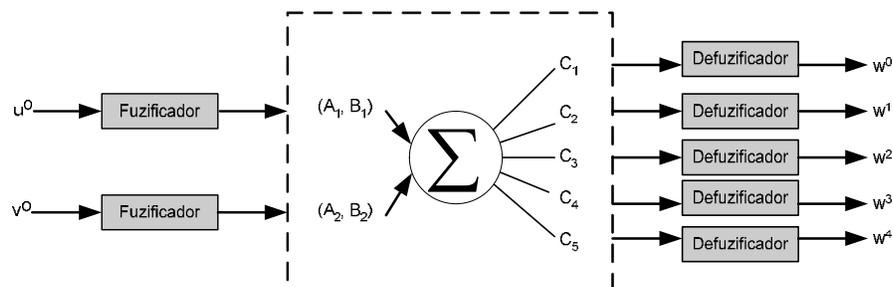


Figura 6.1 Modelo do Motor de Inferência Fuzzy utilizado

$$C' = \sum_1^n (A' \wedge B') \circ [(A_i \wedge B_i) \rightarrow (C_i \wedge D_i \wedge E_i \wedge F_i \wedge G_i)] \quad (6.2)$$

Pelo Método do Centro de Gravidade, w_i representa o centro local de gravidade e $\mu_c(w)$ representa a área local, enquanto n representa o número de pontos ou segmentos locais.

$$w^0 = \frac{\sum_{i=1}^n w_i \mu_c(w)}{\sum_{i=1}^n \mu_c(w)} \quad (6.3)$$

Esse motor de inferência, foi implementado para a validação desse trabalho, conforme descrito no apêndice D.

6.2 PROJETO DO SISTEMA NEURAL

Uma rede neural é uma máquina projetada para modelar a maneira de como o cérebro realiza uma tarefa particular ou função de interesse; aqui, a rede é implementada usando-se simulação por programação em um computador digital. A plataforma de simulação [96] serviu para modelar a rede, adaptando os pesos sinápticos a modificações no meio ambiente.

Uma das fases mais relevantes do projeto de uma rede neural é a definição das variáveis dos vetores de entrada e de saída da rede neural. Para cada problema, deve existir um conjunto de valores que represente, o mais fielmente possível, o estado real que se deseja classificar ou inferior. Portanto, para que uma rede neural possa interpolar uma função, o mais confiavelmente possível, deve-se definir suas variáveis e representá-las no vetor de entrada da rede neural, com os respectivos valores desejados de saída [97].

O tipo de rede neural escolhida para este experimento é a *feed-forward*, utilizando para o processo de treinamento o algoritmo *Backpropagation Perceptron* [98]. A topologia escolhida consiste em 3 camadas de neurônios (entrada, intermediária e saída) e é mostrada na figura 6.2.

Conforme pode ser visto, são utilizados os mesmos atributos de tráfego que foram utilizados na inferência *fuzzy*.

Um processo de normalização foi aplicado aos valores brutos, tendo como objetivo colocar os valores que compõem o banco de exemplos para treinamento e validação da rede neural no intervalo $[-1; 1]$. Esta medida tem como objetivo possibilitar e/ou facilitar a convergência no processo de treinamento da rede neural.

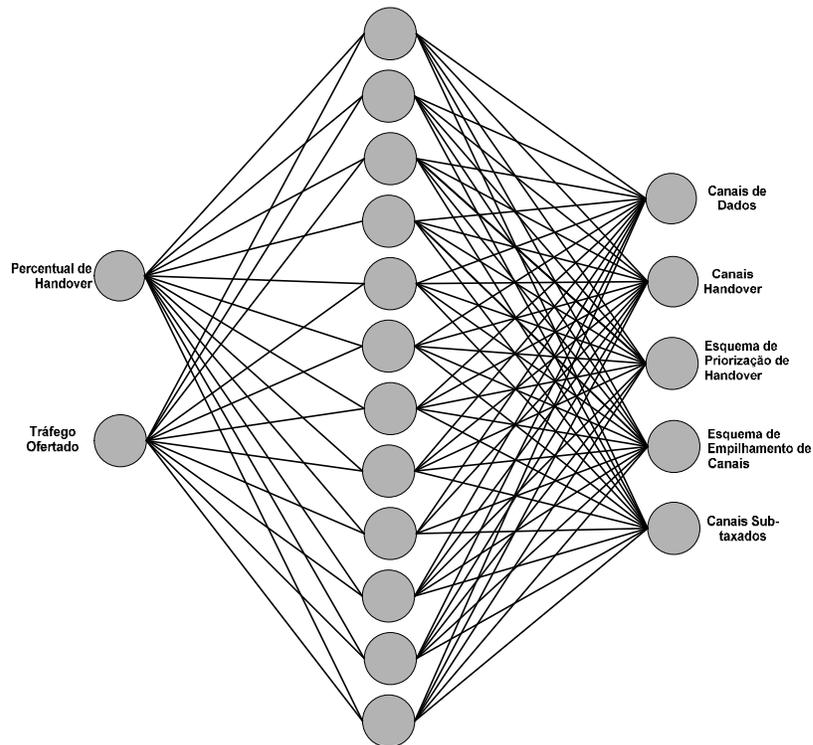


Figura 6.2 Topologia da Rede Neural Artificial de 3 Camadas Utilizada

No contexto dessa tese, essas duas ferramentas de inteligência anteriormente citadas foram aplicadas sobre valores simulados de tráfego GSM/GPRS.

6.3 MODELAGEM DO TRÁFEGO TELEFÔNICO

O desempenho de uma rede celular pode ser analisado por meio de simulação ou modelos analíticos (ou ambos). Os modelos analíticos dão resultados exatos e mais gerais, mas usualmente são formulados com comportamentos restritivos, especialmente para sistemas complexos.

Contrariamente, modelos de simulação são preferidos em estudos mais específicos de sistemas celulares cobrindo uma determinada área. Isso acontece porque não são colocadas restrições sobre o modelo: o modelo é direto. Nesta tese nenhum estudo analítico é realizado devido à complexidade inerente nos Esquemas de Alocação de Canais. Todos os parâmetros são avaliados usando simulação.

Nesta tese, um simulador baseado em eventos e objetos é construído em Linguagem Java acoplado a um simulador para tráfego telefônico em linguagem C++. Eles simplificam o estudo das redes GSM/GPRS, simulando comportamento da interface aérea U_m ao nível do modelo de tráfego, referente ao processo de alocação de canais, e na interface G_n sobre o plano de gerência distribuída [99]. Nada impede que o mesmo venha, posteriormente, a ser estendido para a simulação de outros serviços disponíveis em uma rede GSM/GPRS. Esse simulador adota o modelo de tráfego apresentado em [12] tendo por base, o uso da biblioteca em C++ CNCL desenvolvido na Universidade de *Aachen* na Alemanha.

O simulador compõe-se dos seguintes módulos:

- Números Randômicos: dispõe de geradores de números randômicos usando congruência linear e congruência linear múltipla. Esses geradores são cruciais nos processos de simulação para geração das distribuições desejadas. Eles estão presentes no módulo Gerador de Tráfego da figura 6.3.
- Avaliações Estatísticas: provê vários métodos de avaliação estatística de resultados da simulação. Essas funcionalidades são importantes para apresentação dos resultados no Gerador de Relatório da figura 6.3.
- Classes *Container*: provê estruturas de dados genéricas que podem conter quaisquer elementos, como: pilhas, listas simples e listas duplamente ligadas.
- Simulação Orientada a Eventos: formado por um conjunto de classes que realizam a simulação orientada a eventos. Podem ser encontrados no Módulo Escalonador de Eventos da figura 6.3.

- Simulação Orientada a Serviços: formado por um conjunto de classes que realizam a simulação orientada a objetos em WEB Service/EJB/Java e é representada pelos módulos superiores no modelo da figura 6.3.

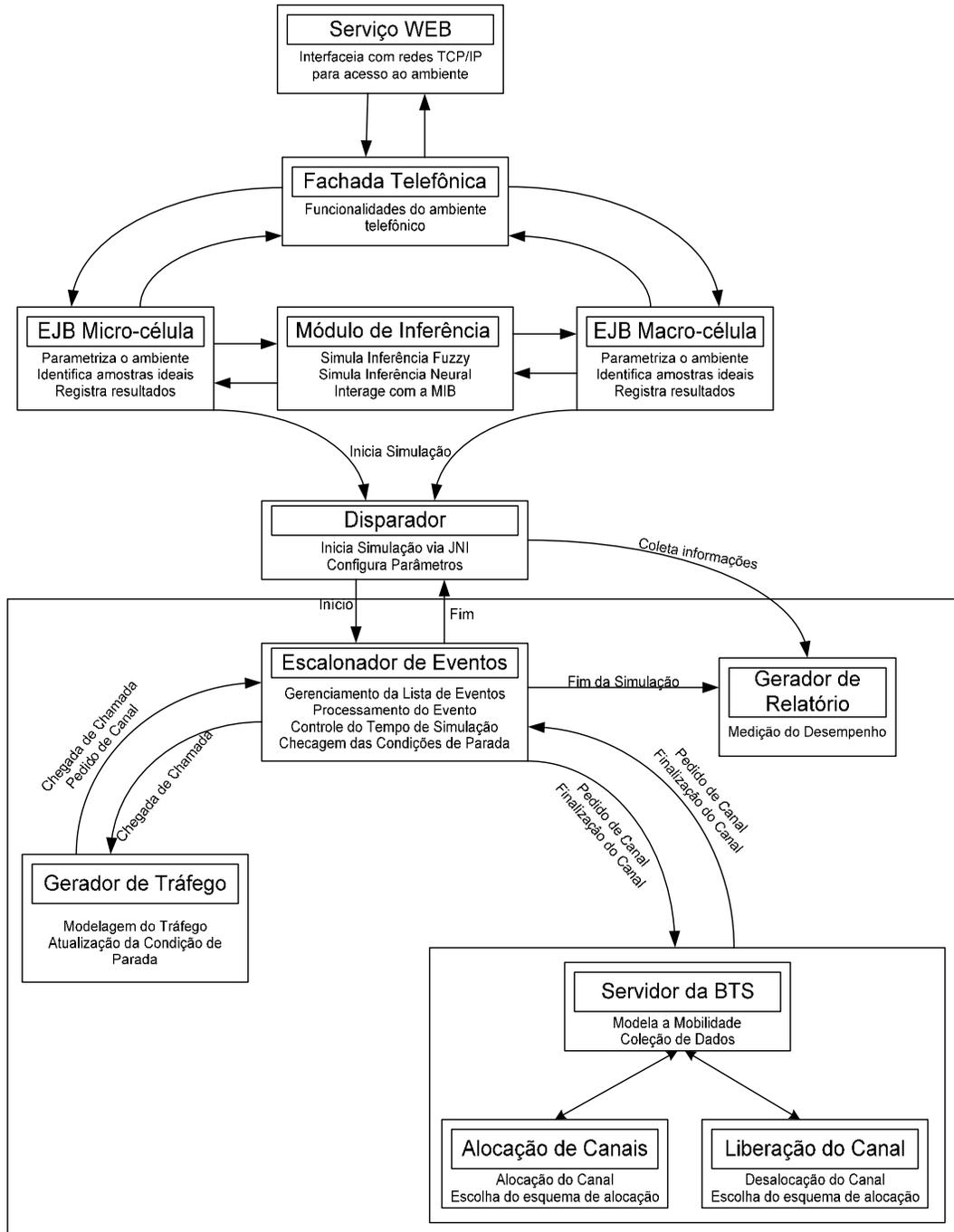


Figura 6.3 Elementos Estruturais do Modelo de Simulação

O controle de eventos é realizado pelo conjunto de classes de tratamento de eventos (Escalonador de Eventos) que são ativados pela chegada desses eventos. Um escalonador determina a ordem de processamento dos eventos. Cada evento tem uma prioridade e um tempo de processamento definido. O tempo de simulação avança em passos discretos e o evento é considerado processado quando o tempo total para processamento deste coincide com o tempo de simulação do mesmo.

A simulação se encerra quando todos os eventos são completamente simulados.

6.4 CENÁRIOS DE SIMULAÇÃO ESTUDADOS

Dois diferentes cenários foram simulados: uma única micro-célula, e uma macro-célula sobreposição a outras micro-células.

O primeiro corresponde a uma micro-célula; este cenário foi escolhido porque células de tamanho pequeno são mais comuns em regiões urbanas e o número de *handover* por chamadas é maior que em macro-células, pois o aparelho móvel terá que mudar de uma célula para outra com mais frequência. Esta célula é isolada das demais, que são descritas apenas pelos pedidos de *handover* ao longo da célula simulada.

Nas micro-células é mais importante utilizarem-se Políticas de Priorização de *Handover* para a manutenção da probabilidade de falhas em níveis aceitáveis. Comumente, o raio das micro-células encontra-se entre 200m e 1km. O simulador está assumindo raio de 800m. Além disso, usuários na periferia da célula (região de *handover*) já são considerados na área da BTS, isso evita efeitos *street-corner*.

Tipicamente, um sistema GSM tem 3 ou 4 portadoras (TRXs) por micro-célula, cada portadora com 8 *time-slots*. Dessa forma, nesta simulação são disponibilizados 32 canais físicos distribuídos em 4 portadoras. Aqui, 2 canais serão reservados para sinalização e os demais para informações do usuário, conforme demonstra a figura 6.4. Em se tratando de rede GPRS, alguns canais são reservados para dados.

Outro cenário simulado foi o de uma macro-célula com outras 7 micro-células sobrepostas, similar à região sombreada da figura 6.5(c). Em topologias como estas, dispositivos com

mobilidade superior a certa velocidade são gerenciados pela macro-célula, ficando apenas os dispositivos de menores velocidades nas dependências da micro-célula. Sendo assim, procedimentos de *handover* podem ocorrer entre micro-células adjacentes ou entre uma micro-célula e sua macro-célula sobreposta.

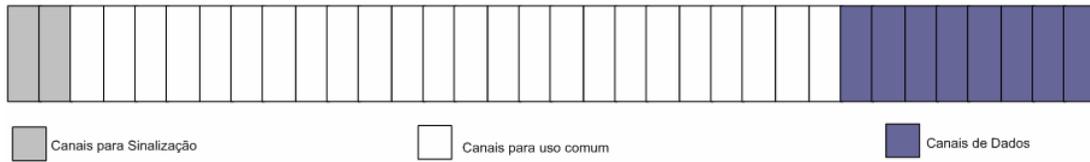


Figura 6.4 Cenário de classificação de Canais em uma Estação Rádio-Base.

A simulação foca nos procedimentos de *downlink*, onde o controle das informações inseridas nos *time slots* fica a cargo das estações rádio-base.

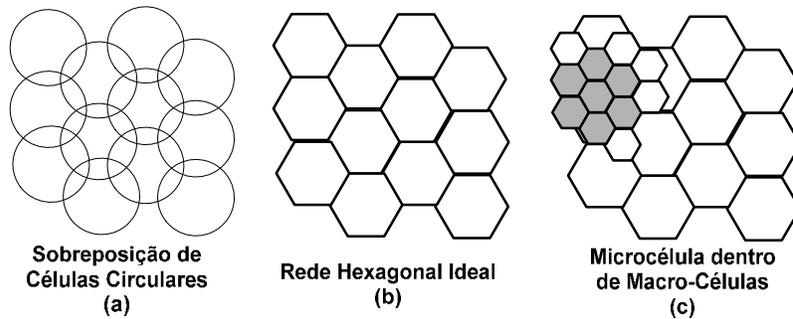


Figura 6.5 Disposição de Células Telefônicas

Conforme seção 3.1, esta simulação adota a técnica PS para alocação de canais. Dessa forma, canais são reservados unicamente para dados, conforme figura 6.4 e 6.6. Na figura 6.6 foi usado o esquema de priorização de *handover*, em algumas das simulações. Logo alguns canais são alocados para realização de *handover*.

Para o restante dessa tese, C_{gprs} representa o número de canais exclusivamente dedicados ao tráfego de dados; C_{HO} são canais dedicados exclusivamente para tráfego de *handover*; e o restante é compartilhado por tráfegos de voz e de dados, ou seja, $C_{comum} = C_{total} - C_{gprs} - C_{HO} - C_{SIG}$.

Para otimização da alocação de canais, uma das principais contribuições deste trabalho, visa variar a quantidade de canais alocados para cada um desses tipos, exceto canais de sinalização. Essa variação resulta numa maior ou menor perda no estabelecimento de novas chamadas ou na quantidade de *handover* falhos.

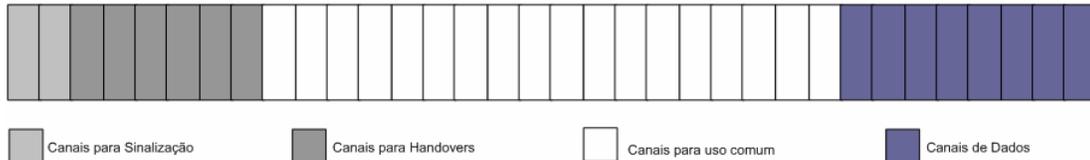


Figura 6.6 Classificação de Canais em uma Estação Rádio-Base com priorização de canais.

A próxima seção apresenta os modelos de tráfego usados para simular tanto o estabelecimento de novas chamadas quanto às realizações de *handover*.

6.5 CRITÉRIOS DE MODELAGEM DO TRÁFEGO

Considerando as novas chamadas (n) e os pedidos de *handover* (h) gerados de acordo com uma distribuição de *Poisson* com taxas de chegada de λ_n e λ_h respectivamente; o tempo entre as novas chamadas e os pedidos de *handover* são ambos exponencialmente distribuídos com taxas de $1/\lambda_n$ e $1/\lambda_h$, respectivamente.

As chamadas iniciadas na célula podem ser divididas entre aquelas finalizadas na própria célula e as que continuam em outras células. O mesmo acontece com os *handovers* que podem terminar nessa célula ou migrar para uma outra. Portanto um canal deve ser ocupado por uma nova chamada ou um pedido de *handover* e ser liberado por causa da conclusão da chamada ou da migração da mesma para outra célula.

O tempo que um usuário particular ocupa um canal é chamado de Tempo de Ocupação de um Canal. Em geral, o tempo de ocupação de um canal é uma variável randômica que é função dos parâmetros do sistema tais como, tamanho da célula, localização do usuário, mobilidade do usuário e tempo de duração da chamada [100]. A Função Densidade de Probabilidade do Tempo de Ocupação do Canal utiliza o tempo de permanência da célula para novas chamadas e *handovers*. A Função de Distribuição do Tempo de Ocupação de um Canal segue uma distribuição exponencial negativa, isto é, a probabilidade de que

qualquer tempo de permanência do canal selecionado randomicamente será igual ou menor que o tempo de duração t , onde,

$$F_{T_{ch}}(t) = P(T_{ch} \leq t) = 1 - \exp^{-\mu_{ch} \cdot t} \quad (6.4)$$

onde $T_{ch} = 1/\mu_{ch}$ é o Tempo Médio de Ocupação do Canal, ao qual depende dos parâmetros de mobilidade, tamanho da célula e tempo médio de duração da chamada.

Os problemas de *handover* ocorrem em altas e médias velocidades móveis. Portanto, considerou-se apenas velocidades médias da ordem de $V = 30$ km/h e desvio padrão σ de 20 km/h, para uma célula com 800 metros de raio, tempo médio de duração de uma chamada de 120 segundos, e o tempo médio de permanência de um canal de 60 segundos [101].

A taxa de chegada de chamadas na célula é a soma da taxa de chegadas de novas chamadas e de *handovers*, ou seja,

$$\lambda_i(\text{chamadas/s}) = \lambda_n + \lambda_h \quad (6.5)$$

então o total de tráfego ofertado para a célula será

$$A_{off}(\text{Erlangs}) = \lambda_i / \mu_{ch} = A_n + A_h \quad (6.6)$$

O total de tráfego ofertado (A_{off}) é variado a fim de mostrar as novas condições sobre o qual a célula está submetida. Então:

$$A_h = f_{ho} \times A_{off} \quad , \quad A_n = (1 - f_{ho}) \times A_{off} \quad (6.7)$$

Onde f_{ho} corresponde ao percentual de *handover* sobre o tráfego total.

Em suma, dois principais parâmetros de desempenho considerados nos serviços são:

- Probabilidade de bloqueio de novas chamadas (P_{nb}): é a probabilidade que uma nova chamada não possa ser estabelecida devido à falta de canais.

- Probabilidade de falha de *handover* (P_{hr}): é a probabilidade de que um novo pedido de *handover* não seja atendido por falta de canal disponível.

Outros dois parâmetros utilizados aqui são:

- Capacidade da Rede: é o tráfego realmente suportado pela rede. Ou seja, é o montante de tráfego admitido para a rede celular submetida a um tráfego ofertado A_{off} . Essa capacidade é determinada por:

$$A_{sup}(\text{Erlangs}) = A_n \times (1 - P_{nb}) + A_h \times (1 - P_{hf}) \quad (6.8)$$

Considerando a matemática analítica apresentada acima, a seção seguinte apresentará como se processou a simulação e quais as informações efetivamente importantes para a inferência sobre os modelos de inteligência.

6.6 O PROCESSO DE EXECUÇÃO

Toda a simulação se baseia na análise do comportamento do tráfego telefônico. Dessa forma, o processo de execução do modelo proposto nessa tese tem início na geração do tráfego telefônico que acontece sob o Módulo de Coleta.

6.6.1 A simulação do Tráfego Telefônico

Os procedimentos abaixo são necessários para a execução da geração do tráfego telefônico:

- escolha dos atributos de entrada do processo de simulação;
- simulação do tráfego;

- coleta dos atributos de saída do gerador de tráfego simulado.

Alguns atributos de tráfego foram dinamicamente configurados e alterados a cada execução do processo, são eles:

- percentual de chamadas de voz, dentro do volume total de chamadas (P_{voz});
- percentual de chamadas de *handover*, dentro do volume total de chamadas ($P_{handover}$);
- Número de canais para dados (N_{dados});
- Número de canais de *handover* (N_{ho});
- Número de canais sub-taxados (N_{sub});
- Esquema de priorização de *handover*: pré ($H_{pré}$) ou pós-reserva (H_{pos});
- O tipo de empilhamento de chamadas (Q), caso esteja sendo usado; e,
- Tráfego ofertado à rede ($A_{ofertado}$).

Além dos atributos acima citados, outros também devem ser configurados; todos estão apresentados na tabela 6.1 e são configurados com valores que são mantidos constantes ao longo de todas as simulações efetuadas.

Para cada simulação, um conjunto de (P_{voz} , $P_{handover}$, N_{dados} , N_{ho} , N_{sub} , $\langle H_{pré} | H_{pos} \rangle$, Q , $A_{ofertado}$) é submetido ao processo de execução. O processo de simulação do tráfego telefônico é feito segundo as etapas apresentadas na parte inferior da figura 6.3. O diagrama de atividades da figura 6.7 apresenta essas mesmas etapas de execução numa ordem de passos sequenciais.

Esse processo é detalhado, ainda mais, nos diagramas das figuras 6.8 a 6.13 que especificam os sub-módulos de escalonamento de eventos, geração de tráfego e alocação de canais nas ERBs.

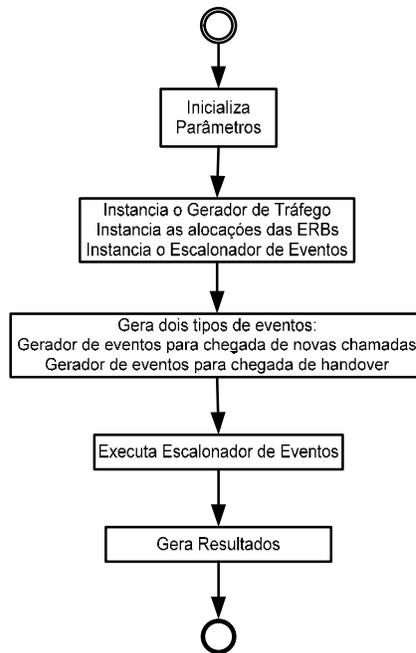


Figura 6.7 Diagrama de Atividades (parte inferior da figura 6.3) (extraído de [12])

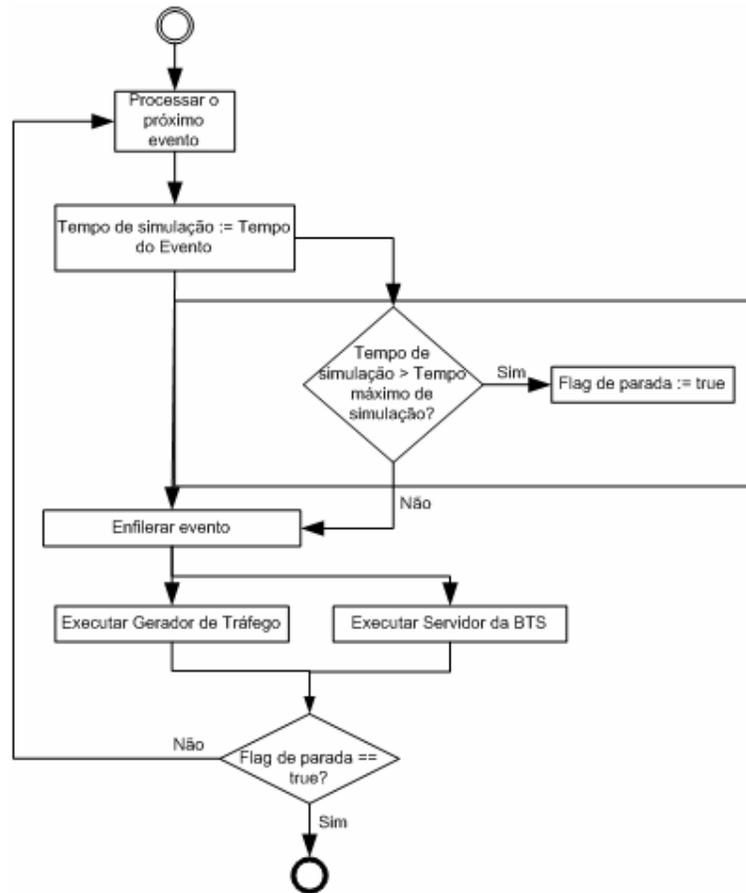


Figura 6.8 Escalonador de Eventos (extraído de [12])

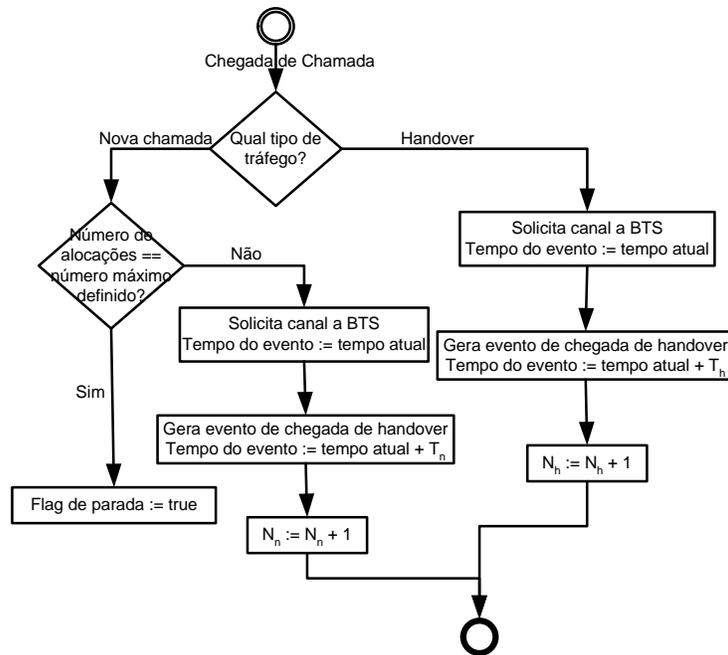


Figura 6.9 Gerador de Tráfego (extraído de [12])

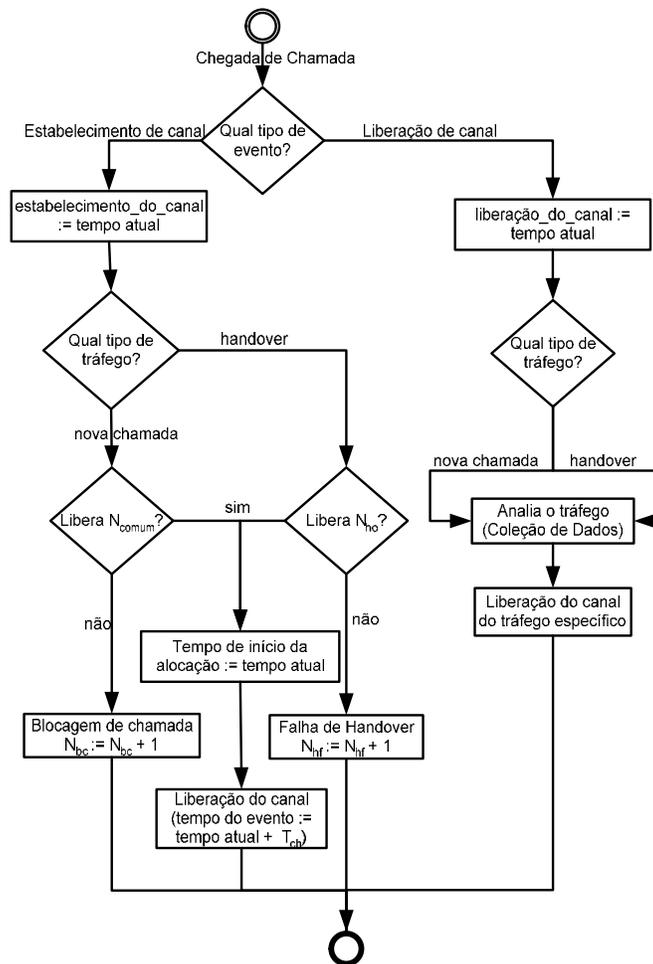


Figura 6.10 Servidor da BTS (Priorização de *Handover*) (extraído de [12])

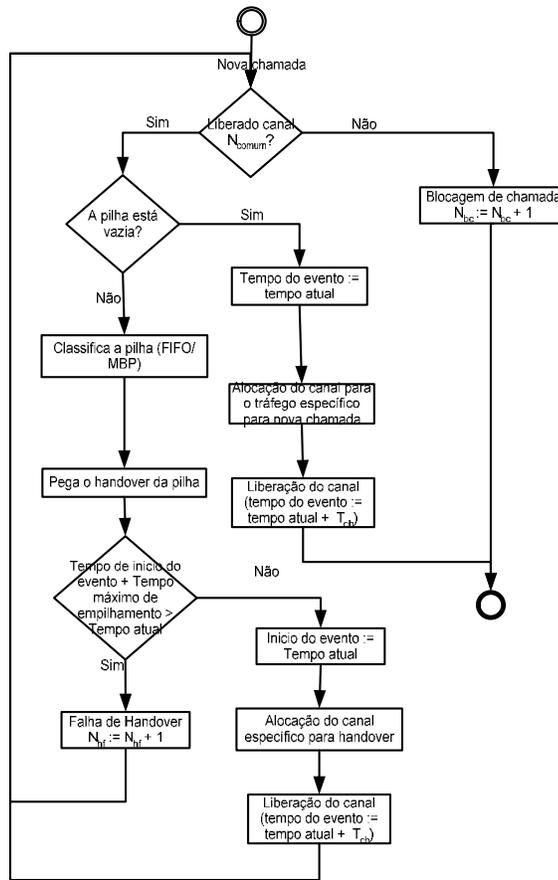


Figura 6.11 Servidor da BTS (Empilhamento de Novas Chamadas) (extraído de [12])

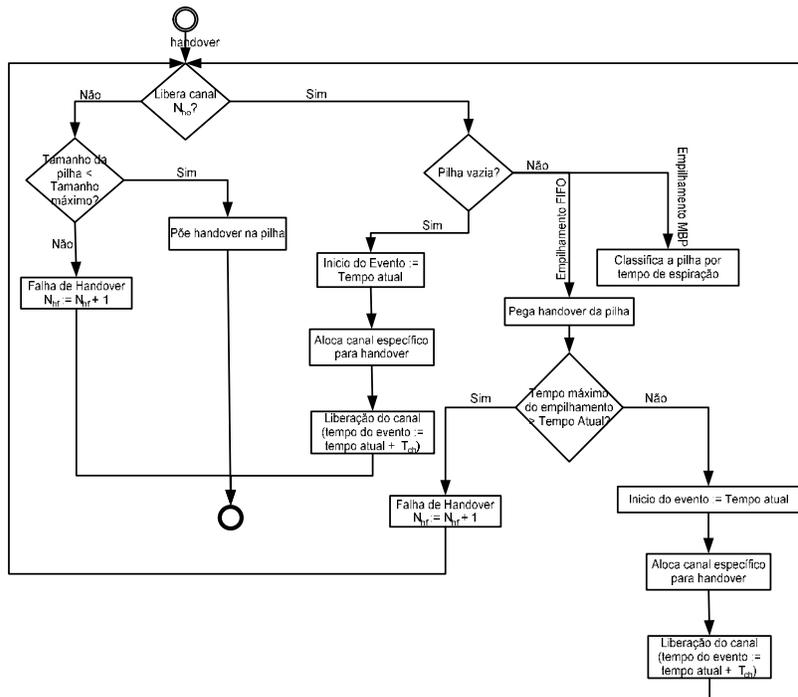


Figura 6.12 Servidor da BTS (Empilhamento de Handover) (extraído de [12])

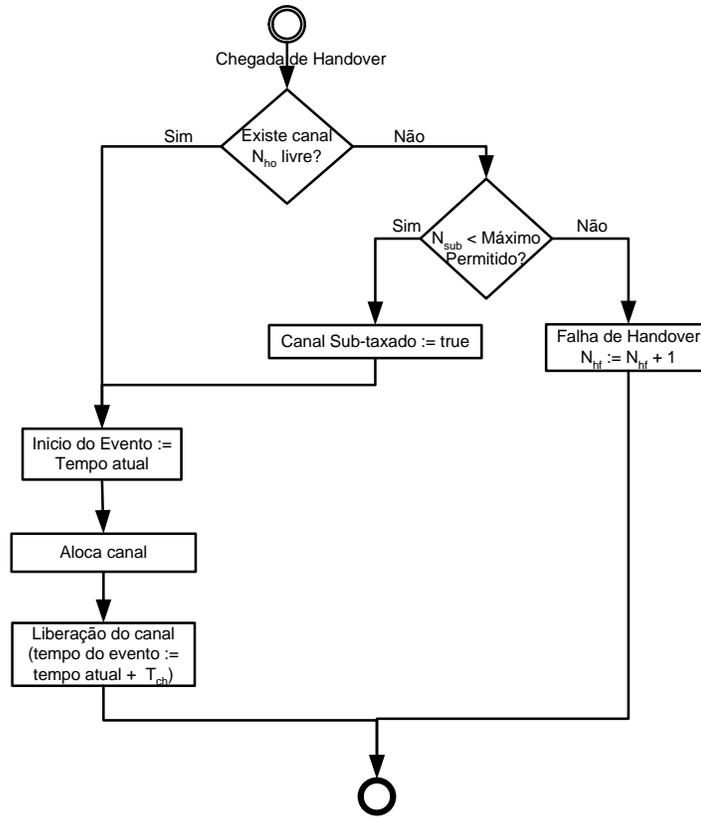


Figura 6.13 Servidor da BTS (Sub-taxação de Chamadas) (extraído de [12])

Após a realização de uma simulação, três informações são de fundamental importância para a análise do modelo:

- percentual de chamadas bloqueadas (P_{bc});
- percentual de *handovers* falhos (P_{ho}), ou seja, não realizados com sucesso; e,
- carga efetivamente suportada pela rede ($A_{suportada}$).

Dessa forma, para uma dada entrada (P_{voz} , $P_{handover}$, N_{dados} , N_{ho} , N_{sub} , $\langle H_{pré} | H_{pos} \rangle$, Q , $A_{ofertado}$), é obtida uma configuração de saída (P_{bc} , P_{ho} , $A_{suportada}$), ou seja:

$$(P_{voz}, P_{handover}, N_{dados}, N_{ho}, N_{sub}, \langle H_{pré} | H_{pos} \rangle, Q, A_{ofertado}) \square (P_{bc}, P_{ho}, A_{suportada}) \quad (6.10)$$

Esse processo é repetido para os intervalos de valores de cada um desses atributos, gerando uma base de conhecimento do tráfego da rede que é armazenada em arquivos textos, ou seja, arquivos de informações (ou *logs*).

Na etapa seguinte, esses arquivos são lidos e suas informações utilizadas na realização da inferência nos dois sistemas de inteligência artificial.

6.6.2 Inferência nos Sistemas Inteligentes

6.6.2.1 Inferência *Fuzzy*

O sistema de controle *fuzzy* e o sistema de rede neural trabalham de forma dissociada, um do outro; ou seja, suas execuções são separadas. Em momento algum, os resultados gerados por um processo de inferência interferem na análise feita pelo outro processo.

Conforme comentado anteriormente na seção 6.1, um controle *fuzzy* é usado para selecionar a configuração ideal conforme a carga a que o sistema telefônico é submetido ($A_{ofertado}$). Dessa forma, o motor de inferência da figura 6.1 é aplicado sobre as regras de controle apresentadas no apêndice D.

As informações de $P_{handover}$ e $A_{ofertado}$ são valores usados como entrada para o processo de inferência *fuzzy*, cuja saída corresponde aos valores de $(N_{dados}, N_{ho}, N_{sub}, \langle H_{pré} | H_{pos} \rangle, Q)$. Dessa forma, fica caracterizado o processo de inferência *fuzzy*.

A próxima etapa, no processo de execução do modelo, é a execução continuada da simulação usando como parâmetros para re-configuração dos canais da ERB, esses valores resultantes do processamento *fuzzy*. Tais resultados estão apresentados na seção 6.7 para dois cenários distintos.

6.6.2.2 - Inferência Neural

A estrutura da rede neural, apresentada na figura 6.2, foi usada para inferência dos valores (P_{handover} , A_{ofertado}) resultando nos valores de configuração ideal da ERB, (N_{dados} , N_{ho} , N_{sub} , $\langle H_{\text{pré}} | H_{\text{pos}} \rangle$, Q).

Inicialmente, para a execução da inferência neural, todos os valores (atributos) precisaram ser normalizados para o intervalo $[0, 1]$, constituindo assim, uma etapa de pré-processamento neural. Para isso, $[0, 100]$ foi usado como intervalo real de P_{handover} , enquanto $[0, 30]$ Erlangs foi o intervalo de valores considerado para A_{ofertado} .

Para o processo de pré-processamento, foram escolhidas tuplas de valores que obedecem à equação 6.10, onde $A_{\text{suportado}}$ é maximizado, dentre todas as soluções possíveis que obedecem à dada equação.

Para o processo de treinamento neural foram usados valores reais e ruidosos na proporção de 1 para 2, ou seja, para cada tupla com valores reais e otimizado segundo $A_{\text{suportado}}$, duas outras tuplas apresentavam variadas formas de ruídos. O apêndice E apresenta todo o processamento de um conjunto de épocas utilizado. Testes adicionais foram feitos sobre a rede neural treinada. A seção seguinte apresenta resultados obtidos a partir da configuração dessa rede neural.

6.7 RESULTADOS DE SIMULAÇÕES

Diversas simulações, com diferentes parametrizações, foram feitas sobre diferentes cenários da rede telefônica GPRS. Tais simulações focam dois cenários distintos: uma micro-célula e uma macro-célula sobreposta à primeira.

Em ambos os cenários, várias diferentes simulações foram feitas. A maioria delas usa alocação fixa FCA. Nessas alocações, são feitas simulações usando as várias técnicas para alocação de canais, discutidas no capítulo 3.

Todos os gráficos de alocação FCA são comparados com a simulação otimizada *fuzzy* e neural. Essa comparação é apresentada tanto graficamente, quanto matematicamente, por meio de tabelas comparativas.

6.7.1 Primeiro Cenário – Micro-célula GSM/GPRS

O primeiro cenário consiste em uma Micro-célula com $N_{trx} = 4$ em que os parâmetros usados para a configuração deste cenário são mostrados na tabela 6.1 e as demonstrações gráficas, a partir da figura 6.14.

Tabela 6.1 - Configurações Básicas da Micro-Célula

Parâmetros	Valores
Raio da Célula, R_{micro}	800 m
Número de TRXs, N_{trx}	4
Número total de canais disponíveis na célula, N_{total}	32
Número de canais de sinalização, N_{sig}	2
Número de canais de tráfego, N_{ch}	30
Número de canais para dados, N_{dados}	[0, 6]
Número de canais para handover, N_{ho}	[0, 6]
Número de canais subtaxados, N_{sub}	[0, 5]
Tempo de duração média de chamada, $1/\mu$	2 minutos
Velocidade Média	30 Km/h
Desvio Padrão da Velocidade	20 Km/h
Tempo médio de posse do canal	60 segundos
Tempo médio do intervalo de degradação (QPS)	6 a 8 segundos
Desvio Padrão do intervalo de degradação (QPS)	4 a 5,3 segundos
Carga total ofertada, A_{off}	Variável
Taxa de chegada de novas chamadas, λ_n	Variável
Taxa de chegada de <i>handover</i> , λ_h	Variável
Percentual de tráfego de <i>handover</i> sobre A_{off}	Variável

Dois tipos de gráficos foram gerados. O primeiro, apresenta o tráfego oferecido à rede com suas taxas de perda de comunicação por meio dos percentuais de chamadas bloqueadas e de falhas de *handovers*. O segundo tipo de gráfico mostra o tráfego efetivamente suportado pela rede e gerado a partir da equação 6.8.

Conforme a tabela 6.1, o tráfego ofertado e o tráfego suportado, desses gráficos, foi simulado variando valores de 0 a 30 Erlangs. Isso porque de um total de 32 canais, 2 foram disponibilizados para sinalização e um total de 30 canais foram variados entre dados e voz.

Como resultado final de validação da técnica otimizada para o cenário da micro-célula, a figura 6.14 apresenta a otimização do gráfico *fuzzy* e neural por sobre as 4 simulações FCA, a saber:

- alocação de 2 canais para dados, apenas;
- alocação de 6 canais para dados, apenas;
- alocação de 4 canais para dados, 4 para a realização de *handovers* usando a técnica de Pré-Reserva de Canais de *Handover* e empilhamento por meio da técnica FIFO; e,
- alocação de 4 canais para dados, 4 para a realização de *handovers* usando a técnica de Pré-Reserva de Canais de *Handover*, empilhamento por meio da técnica FIFO e a sub-taxação de 5 canais.

Várias outras configurações foram testadas, mas estas foram escolhidas para facilitar a visualização. Todas as configurações geradas, resultaram da variação dos seguintes parâmetros de entrada da simulação:

- Número de canais para dados, N_{dados} ;
- Número de canais para handover, N_{ho} ;
- Número de canais sub-taxados, N_{sub} ;
- Tipo de priorização de *handover* usado; e,
- Tipo de empilhamento usado.

Devido à natureza em rajada das aplicações GPRS, será uma perda de recursos de rádio se o tráfego de dados não puder usar quaisquer dos canais de tráfego ociosos. Esta é a principal razão pela qual alocar um novo espectro para serviços de dados não é considerado. Daí, este trabalho considerar a técnicas PS para alocação de canais, conforme discutido na seção 3.1.

Para o gráfico da figura 6.14, quanto maior o tráfego suportado em relação ao tráfego ofertado, melhor o desempenho da rede. Como se pode ver, a inferência por meio da rede neural apresentou resultados consideravelmente melhores para tráfego 'meio alto'; enquanto isso, tráfegos verdadeiramente altos demonstram bons resultados por meio da inferência *fuzzy*.

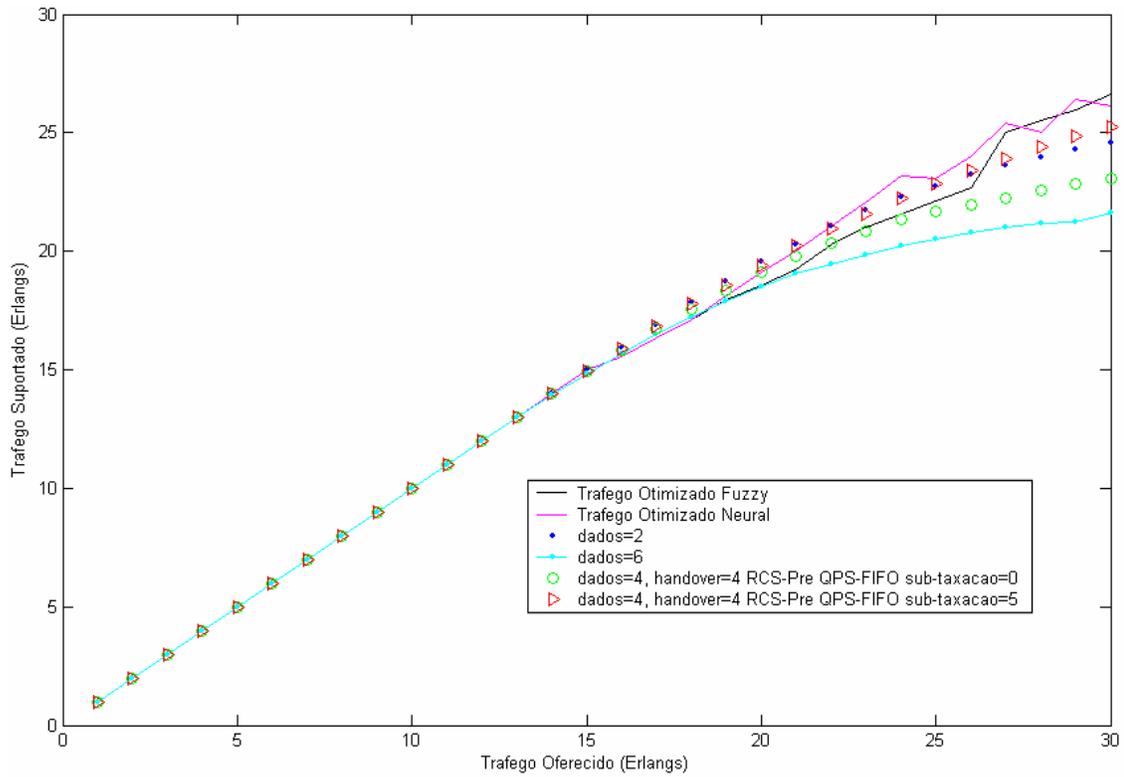


Figura 6.14 - Comparação dos desempenhos dos tráfegos em uma Microcélula.

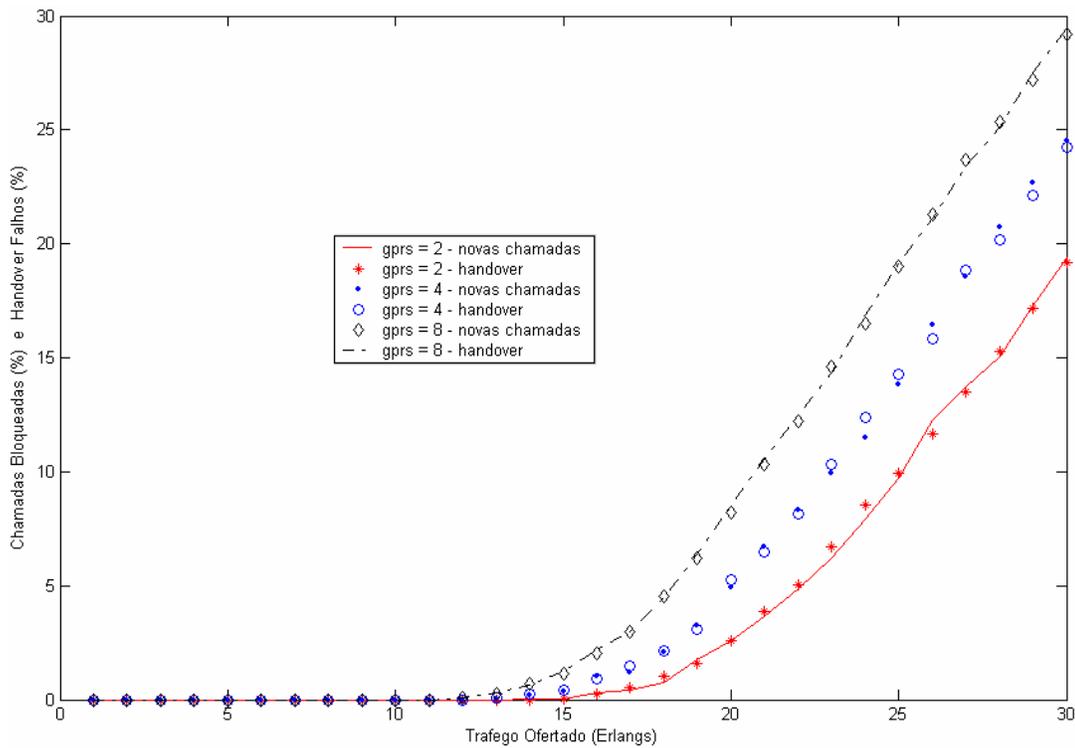


Figura 6.15 – Taxa de Perda quando alocados canais de dados

Essa conclusão deriva da análise de alguns outros gráficos simulados e apresentados logo abaixo.

A figura 6.15 mostra o efeito do aumento no número de canais GPRS sobre a capacidade de tráfego dos serviços GSM, quando não é usado qualquer esquema de priorização de *handover*. Uma vez que o número total de canais é limitado, a figura demonstra que quanto maior a quantidade de canais alocados para dados, maior tende a ser a perda de chamadas de voz, ou seja, P_{nb} e P_{hf} . Por causa disso, o tráfego suportado tende a diminuir, conforme demonstra a figura 6.16.

Na figura 6.15, tanto P_{hf} quanto P_{nb} são aproximadamente iguais por não se usarem esquemas de priorização. Além do mais, conforme o número de canais GPRS aumenta, menor é a capacidade para transmissão de voz por meio dos canais compartilhados. Neste caso, o uso de esquemas de priorização de *handover* permite a redução de P_{hf} .

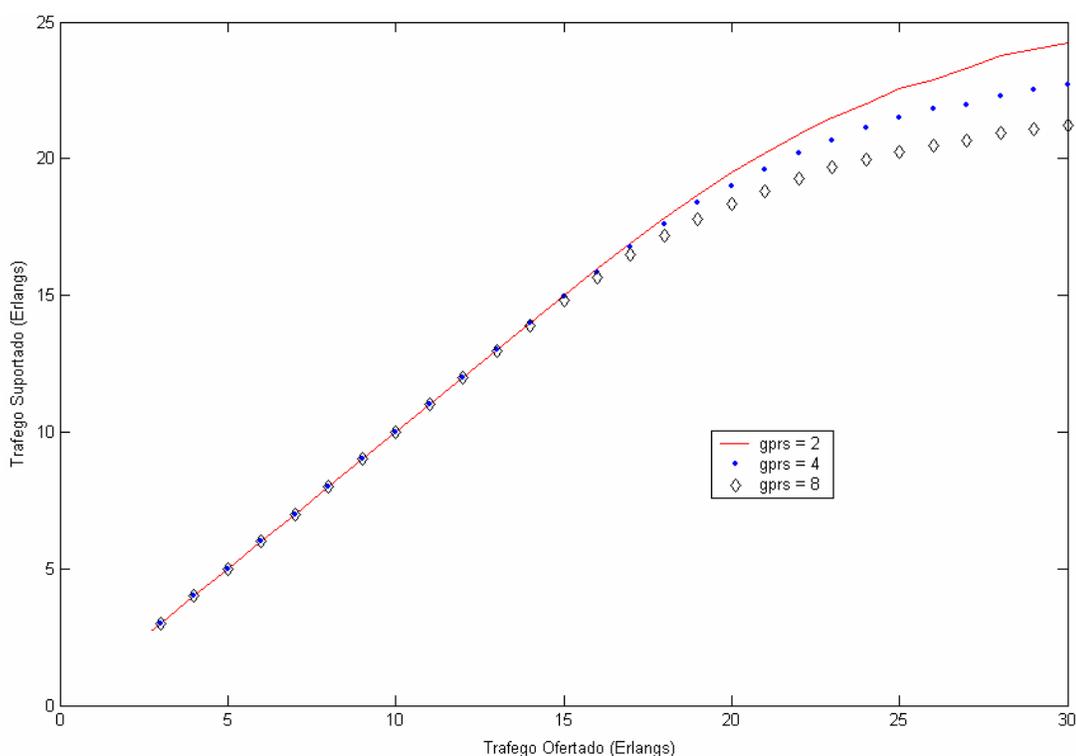


Figura 6.16 – Carga da Rede

Uma das técnicas de priorização de *handover*, conforme comentada no capítulo 3, é o uso de canais reservados para este fim. Esse tipo de situação é analisada no gráfico da figura 6.17. Ele demonstra que quanto mais canais são reservados para a realização de *handover*, menor a probabilidade de falhas decorrentes desse fim; mas, em contrapartida, aumenta o número de chamadas bloqueadas, pois o número de canais para novas chamadas se reduz.

A figura 6.18 mostra que a carga suportada tende a diminuir, conforme aumenta o número de canais reservados para *handover*. Esse resultado demonstra que para se alcançar um bom desempenho, possibilitando a alocação de canais para dados e para *handover*, a solução é a adoção de um resultado que busque conciliar essas duas situações.

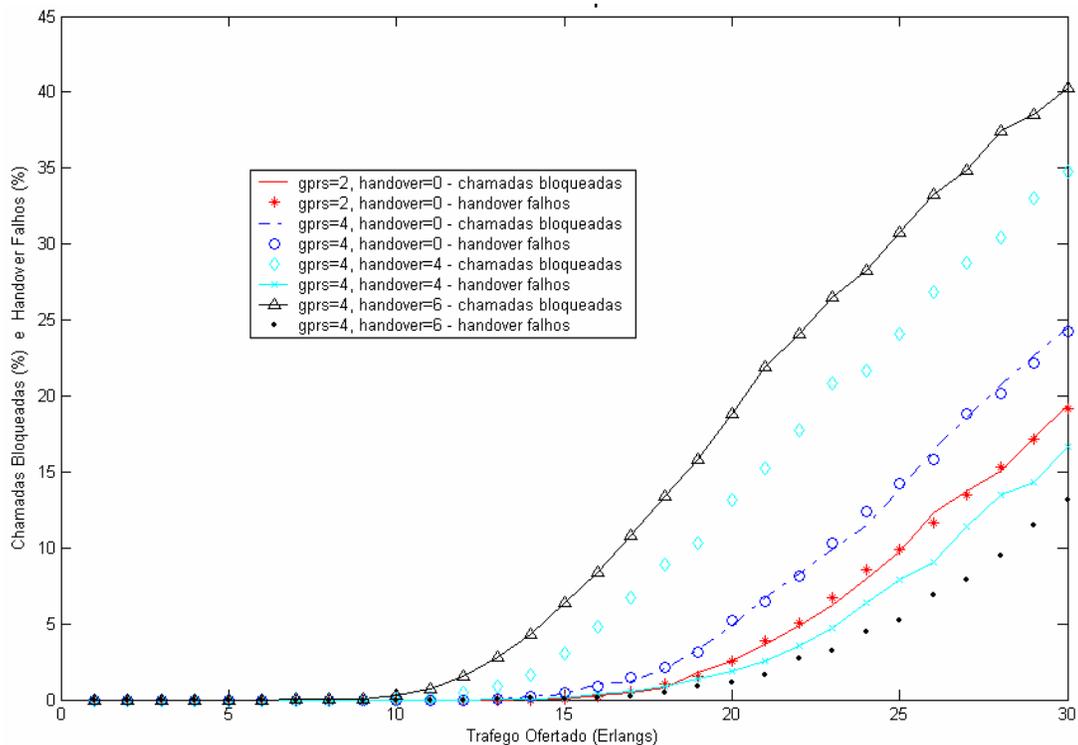


Figura 6.17 – Taxa de Perda quando alocados canais para priorização de *handover*

As figuras 6.19 e 6.20 mostram um comparativo dos tipos de reservas de canais de *handover*: pré-reserva e pós-reserva. Essas figuras mostram que a diminuição de P_{hf} e o aumento de P_{nb} é maior na pré-reserva. Além do mais, a figura 6.20 indica que o tráfego suportado é claramente menor no uso da pós-reserva.

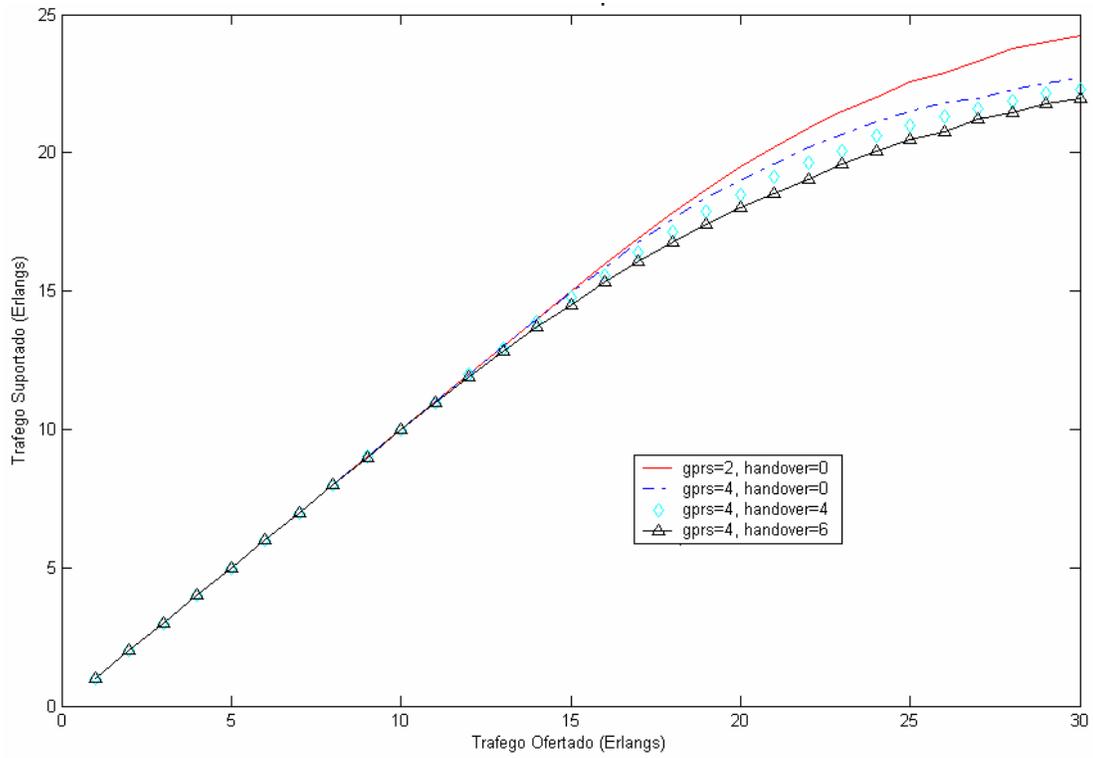


Figura 6.18 – Carga da Rede

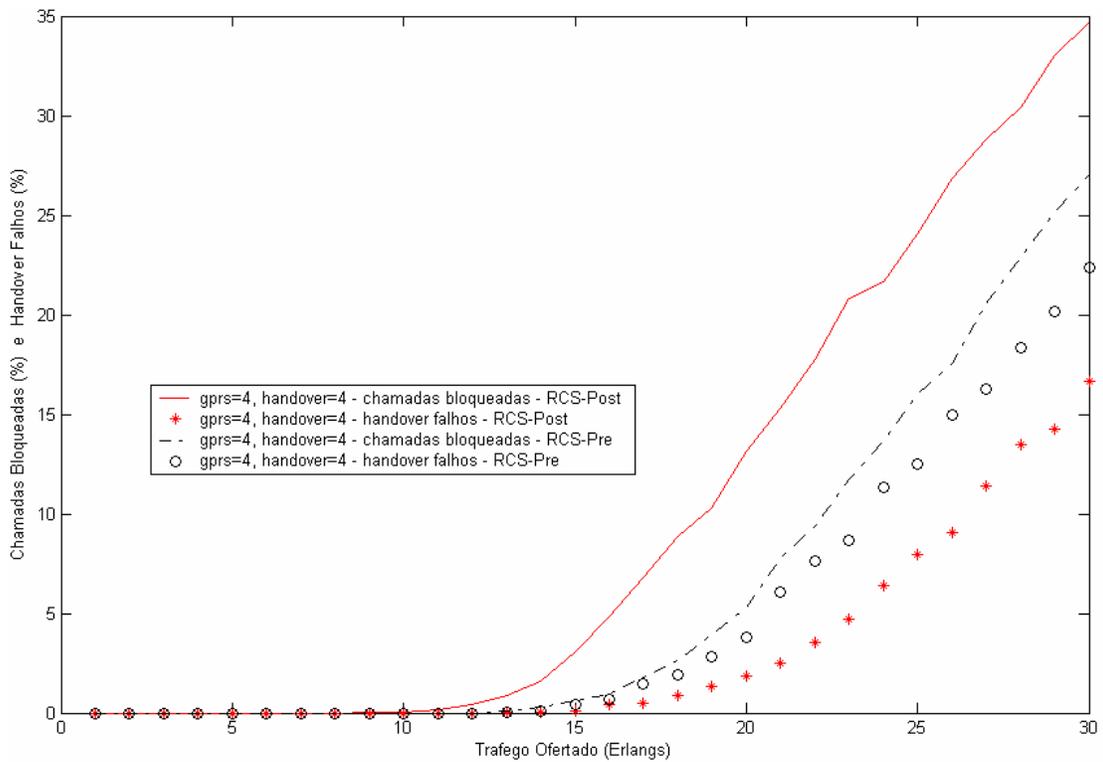


Figura 6.19 – Taxa de Perda de chamadas com tipo de reserva de canais: pré/pós-reserva

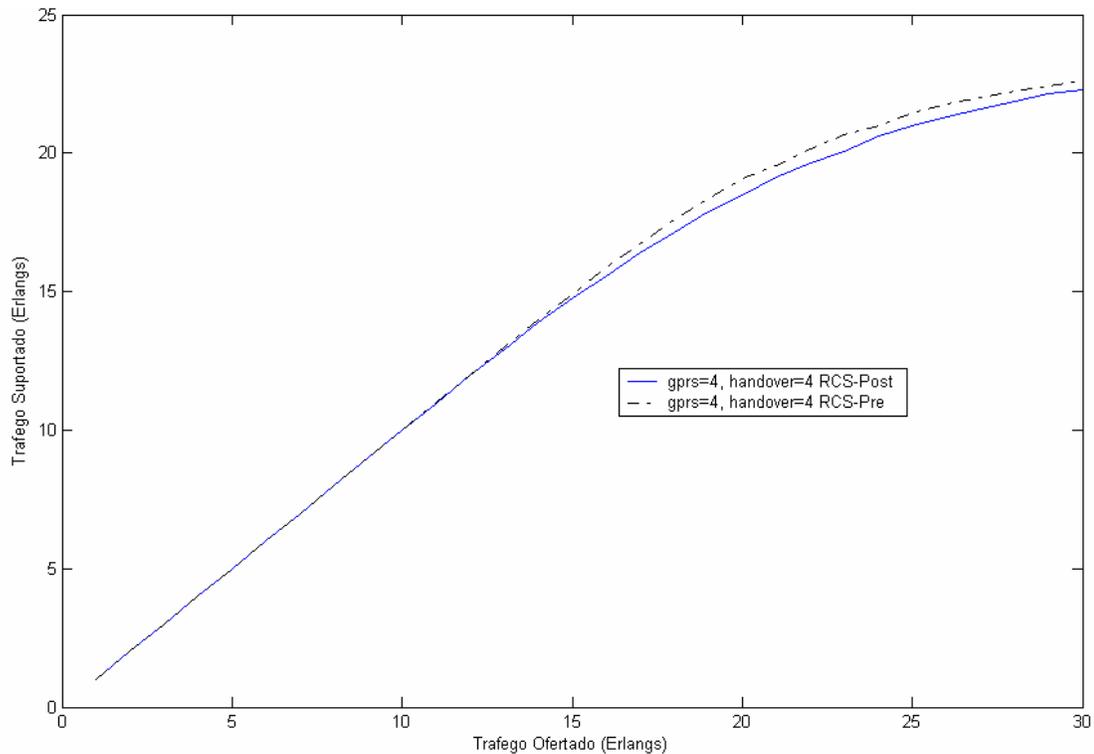


Figura 6.20 – Carga da Rede

Observa-se que a pós-RCS para P_{hf} é menor que a pré-RCS, embora ocorra o contrário com a P_{nb} . Isto é esperado, uma vez que os pedidos de *handover* para o pós-RCS ocupam primeiro os canais de alocação comuns; muitos pedidos de *handover* ocuparão os canais comuns, mesmo havendo alguns canais reservados desocupados. Isso significa que a utilização dos canais para o grupo de canais reservados no caso do pós-RCS será menor que para o pré-RCS.

Duas diferentes técnicas de empilhamento de chamadas de *handover* são comparadas nas figuras 6.21 e 6.22: empilhamento FIFO e empilhamento MBP.

Conforme a figura 6.21, o uso de quaisquer das técnicas de empilhamento reduz consideravelmente o desperdício de falhas de *handover*, em contrapartida, isso pode causar um ligeiro acréscimo no número de chamadas bloqueadas. Dessa forma, o uso de empilhamento deve ser usado apenas quando estritamente necessário.

A figura 6.21 indica que P_{hf} para o esquema MBP é menor que para o esquema FIFO; entretanto, P_{nb} para o esquema FIFO é menor que para MBP. Vale ressaltar que quase não há

diferença no tráfego suportado por quaisquer das duas técnicas, conforme pode ser visto no gráfico da figura 6.22. Nessa figura, a curva da técnica FIFO e MBP, praticamente, se sobrepõem.

Além do mais, a figura 6.21 mostra que o valor P_{hf} de ambos os esquemas é menor que os valores obtidos no gráfico 6.15 que não apresenta priorização de *handover*. Também quanto maior o intervalo de degradação, melhor o desempenho do empilhamento de chamadas.

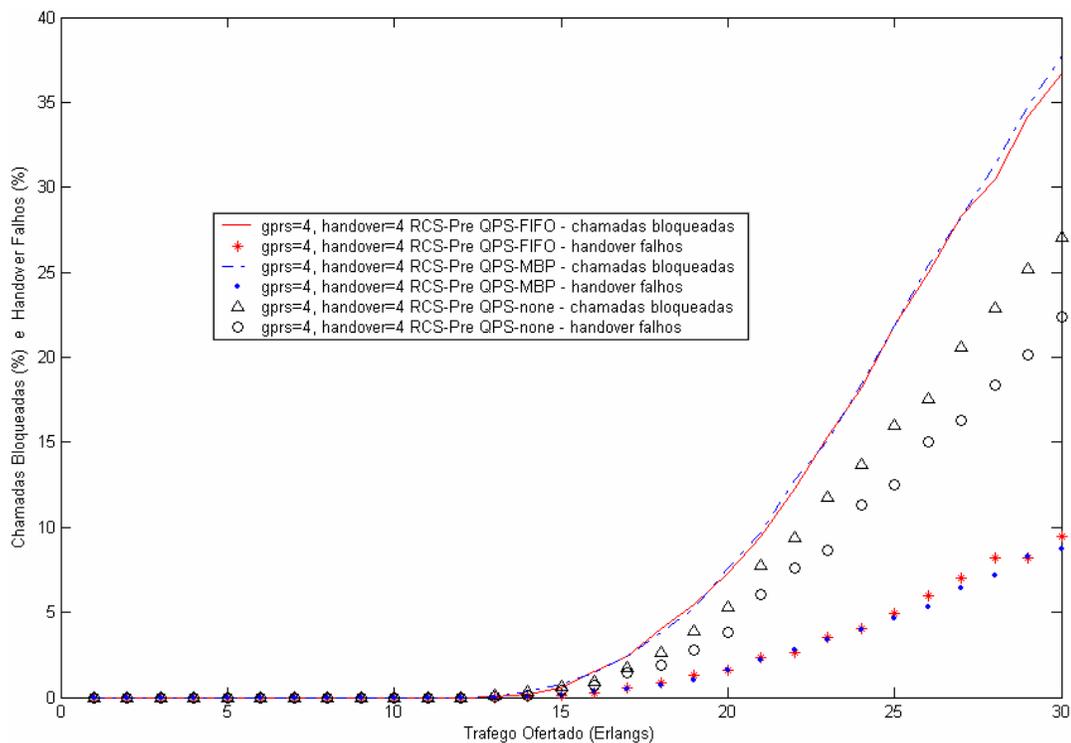


Figura 6.21 – Taxa de Perda quando utilizada uma Técnica de Empilhamento de Chamadas

Os gráficos das figuras 6.23 e 6.24 apresentam o comportamento da rede telefônica quando do uso da técnica de sub-taxação de canais. Por essas figuras, pode-se observar que o uso de sub-taxação beneficia tanto chamadas originárias na própria célula quando chamadas advindas da realização de *handover*. Isso demonstra o benefício de se utilizar canais sub-taxados quando a rede está submetida a uma carga elevada. Essas figuras demonstram que a sub-taxação efetivamente aumenta o tráfego suportado, quando comparado ao caso sem priorização; porque aqui, a P_{hf} diminui sem aumenta a P_{nb} . Mesmo com carga ofertada alta, ainda assim, os percentuais P_{nb} e P_{hf} são menores.

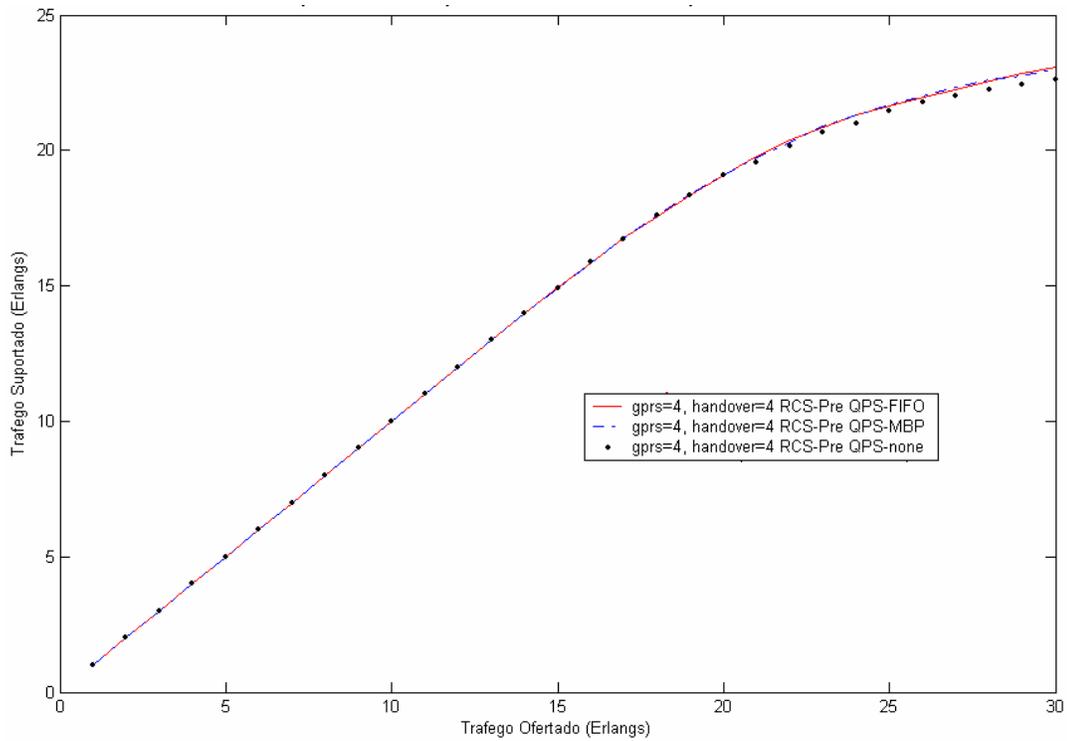


Figura 6.22 – Carga da Rede

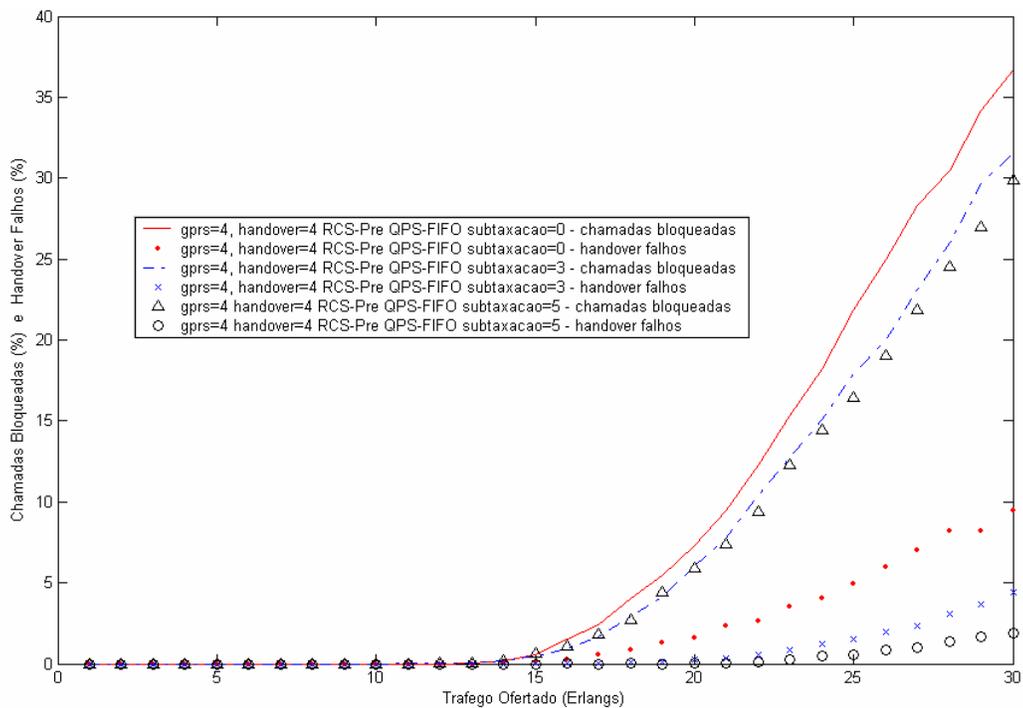


Figura 6.23 – Taxa de Perda quando utilizada uma Técnica de Sub-taxação de Canais

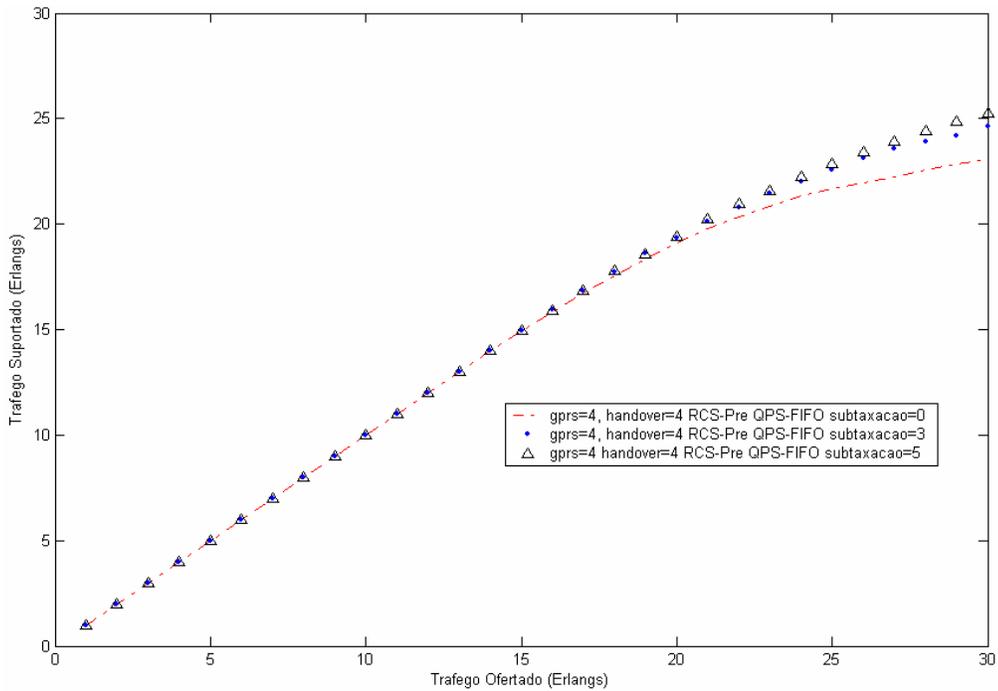


Figura 6.24 – Carga da Rede

O artigo [66] assume que quaisquer canais, do conjunto de canais comuns, pode ser sub-taxado, mas os experimentos aqui realizados só buscam sub-taxar um máximo de 10 canais.

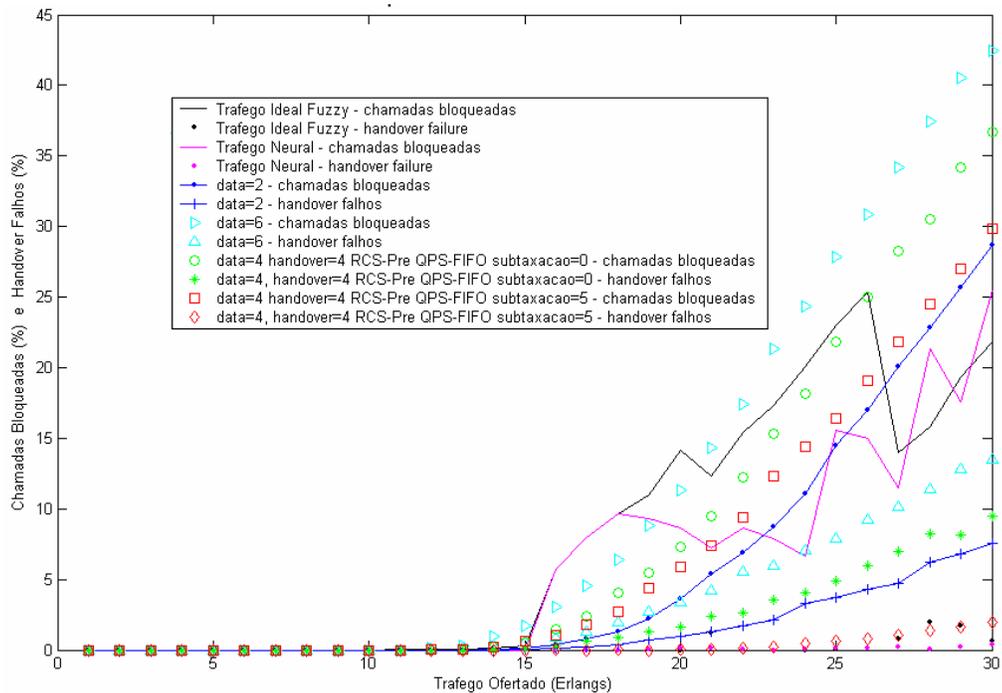


Figura 6.25 – Comparação entre várias das taxas de perdas

Como a qualidade da voz para canais sub-taxados pode não ser boa em relação aos demais, é importante que venham a ser desenvolvidos estudos sobre como os canais sub-taxados afetam a qualidade da voz. Custos referentes à redução da qualidade da voz são largamente estudados em [102].

A figura 6.25 apresenta a taxa de chamadas bloqueadas e de falhas de *handover* referentes à figura 6.14.

6.7.2 Segundo Cenário – Micro-célula sobreposta a uma Macro-célula GSM/GPRS

Na seção anterior o desempenho das realizações de *handover* foram melhorados por meio do uso de técnicas de priorização de *handover*. Em contrapartida, esse esquema de priorização causa aumento na probabilidade de bloqueio de novas chamadas. Para atenuar tal situação, se faz necessário a sobreposição de uma macro-célula sobre essas micro-células.

Dessa forma, é possível a desalocação de chamadas trafegando sobre uma micro-célula para uma macro-célula, e vice-versa. Onde cada micro-célula simulada apresenta os mesmos parâmetros de configuração na tabela 6.1. Segundo [103], as redes GSM, em uso, consistem em múltiplas camadas de células com diferentes tamanhos de células, oferecendo cobertura completa e capacidade suficiente para o volume de tráfego real.

A fim de reduzir o número de *handover* nas micro-células, portanto diminuindo a probabilidade de falhas destes, pode-se fazer com que os dispositivos móveis mais velozes sejam conduzidos através dos canais de alocação da macro-célula. Conforme dito anteriormente, o simulador utiliza uma distribuição normal para definição da velocidade do dispositivo móvel, com média de 30 km/h e desvio padrão de 20 km/h, limitando o intervalo de velocidade em [0, 100] km/h.

Para a realização dessa simulação, foi definido um limiar de 50 km/h, onde qualquer *handover* entre micro-células, em cujo dispositivo móvel tenha velocidade igual ou superior a esta, os próximos canais de tráfego serão alocados dentre os canais disponíveis na macro-célula.

Este cenário foi configurado de modo a que uma macro-célula sobrepusesse outras 7 micro-células, conforme a figura 6.26. Neste cenário, o monitoramento de valores foi realizado sobre a micro-célula central. Dessa forma, o raio da macro-célula é $R_{macro} = 3 \times R_{micro} = 2400m$ e o tempo médio para a sinalização em um canal é de 90 segundos, conforme [101]. Nesse cenário, a micro-célula apresenta então uma probabilidade de bloqueio e de falhas menores que as da seção anterior.

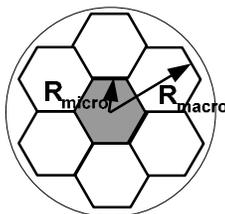


Figura 6.26 Cenário de Sobre-posição Macro-célula/Micro-célula

Considerando que $P(V > 50km/h)$, em uma distribuição normal, é aproximadamente 17%, então o tráfego ofertado da micro-célula para a macro-célula será de aproximadamente 1,25 Erlangs. A fim de suportar uma baixa taxa de bloqueio, a macro-célula foi configurada com 3 TRXs. Totalizando 24 canais físicos, dos quais 2 são reservados para sinalização e os demais 22 são disponibilizados para servir ao *cluster* de alocação. Usando a equação (3.1) e considerando o procedimento de simulação, onde $C=22$ e supondo $A_{off}=13.25$ Erlangs, a probabilidade de bloqueio ficou em 0.72% na macro-célula, de forma que ela terá todas as condições de recepcionar o tráfego de *handover* advindo das micro-células.

O gráfico da figura 6.27 demonstra melhorias no resultado obtido pela inferência *fuzzy* e neural em relação às simulações FCAs. Essa conclusão é derivada de resultados obtidos por meio dos demais gráficos apresentados adiante nessa seção.

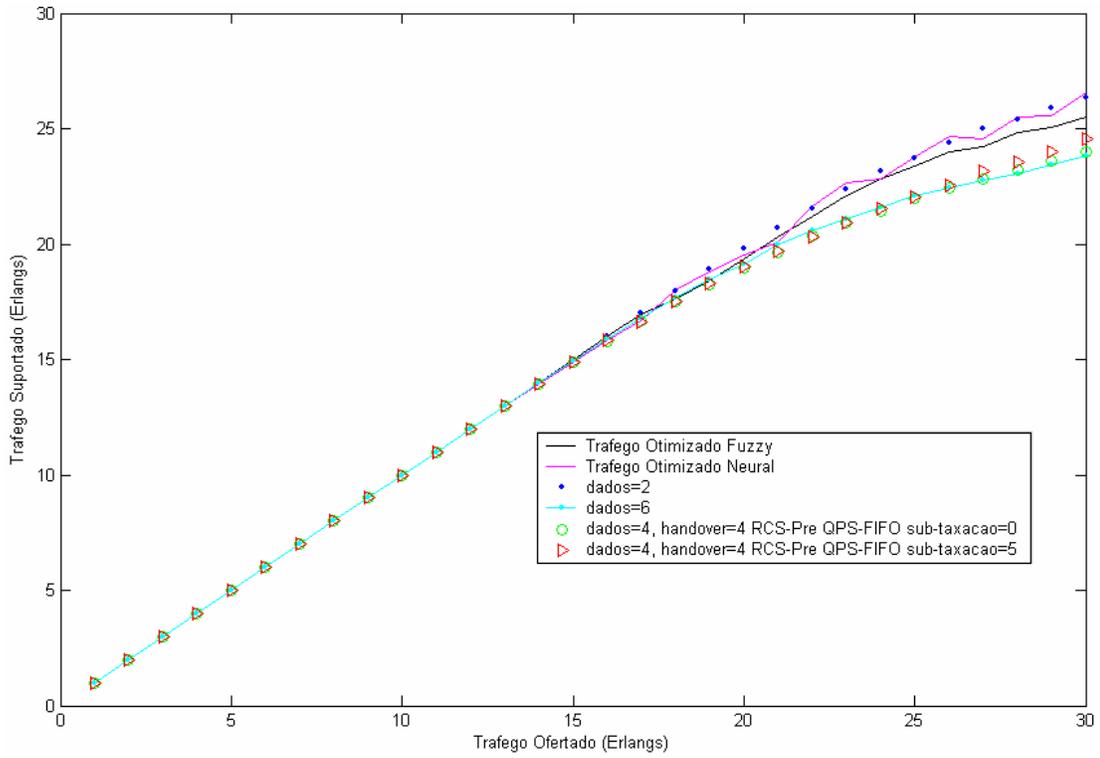


Figura 6.27 – Comparação dos desempenhos dos tráfegos em Micro-célula/Macro-célula

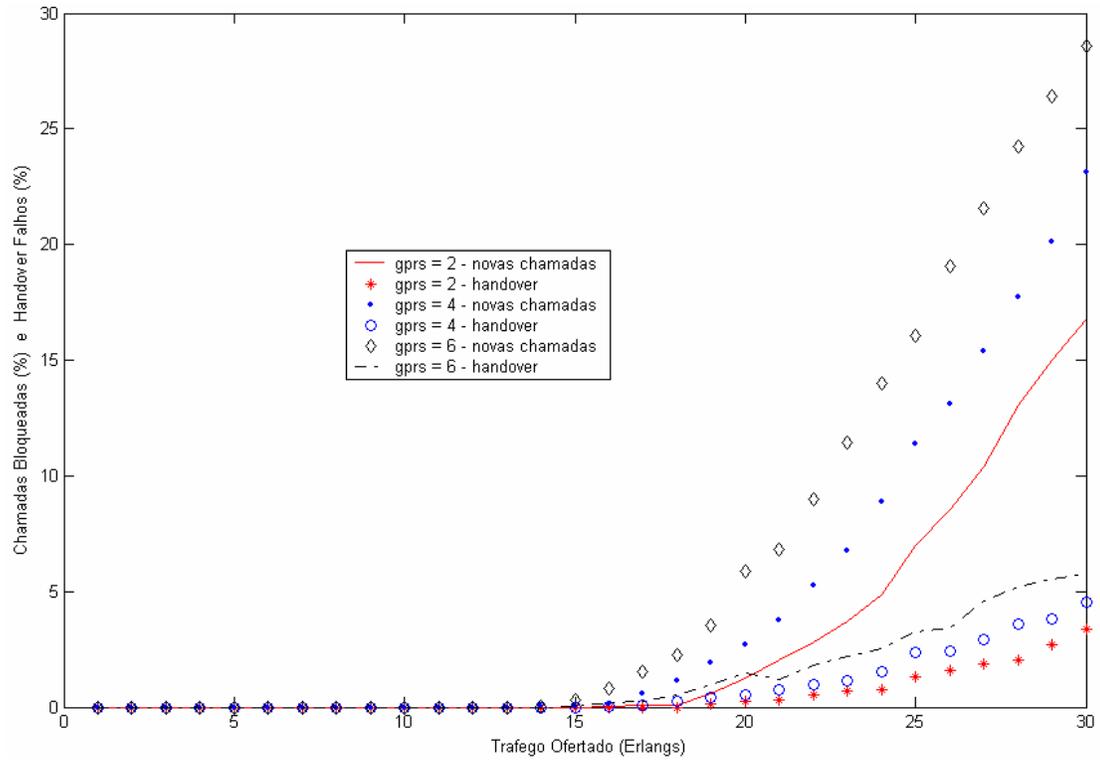


Figura 6.28 – Taxa de perda quando alocados canais de dados.

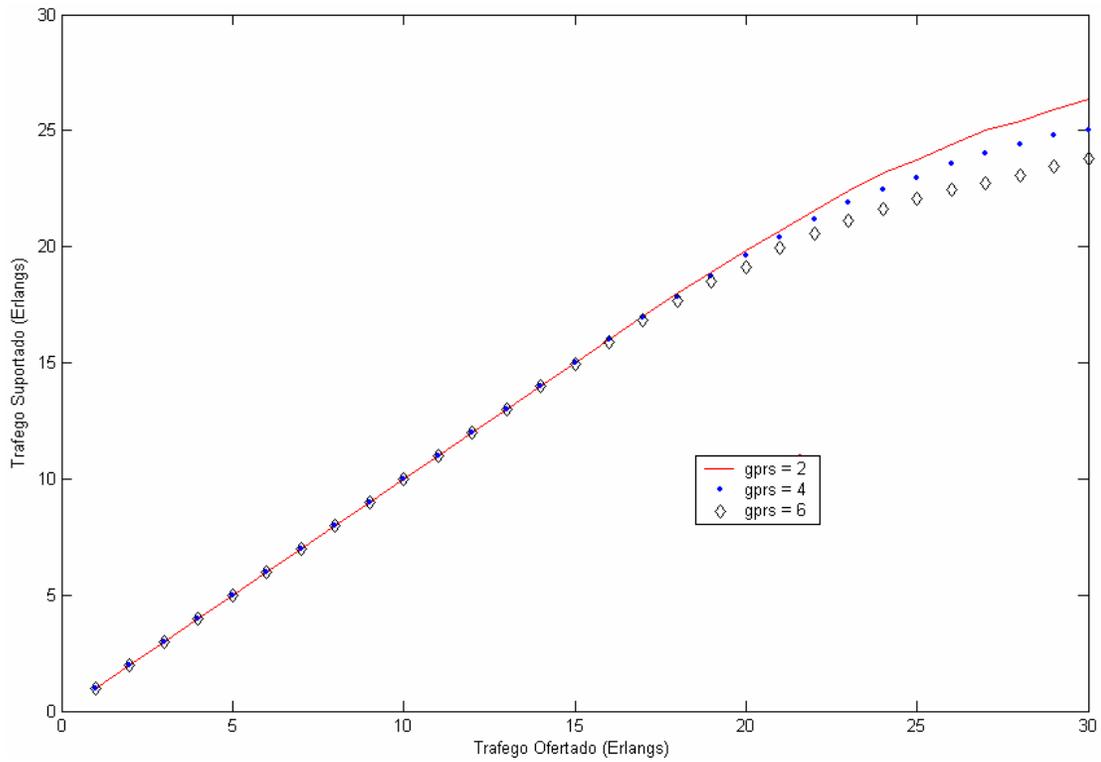


Figura 6.29 – Carga da Rede.

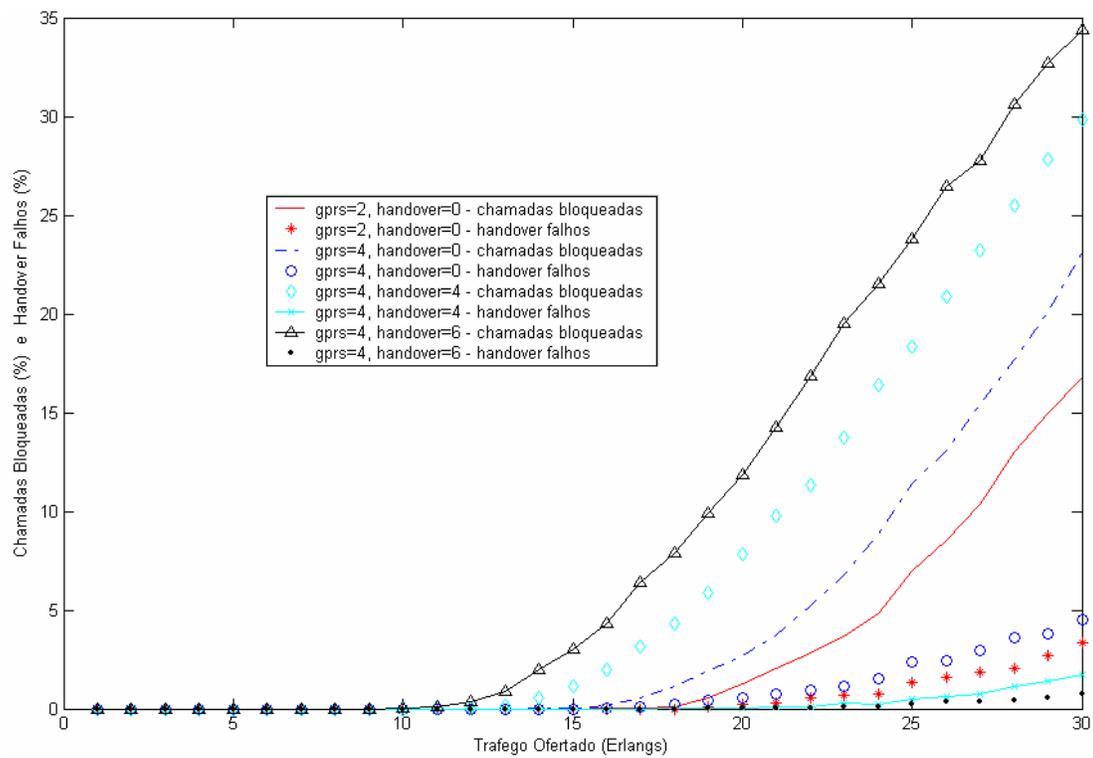


Figura 6.30 – Taxa de Perda quando alocados canais para priorização de *handover*.

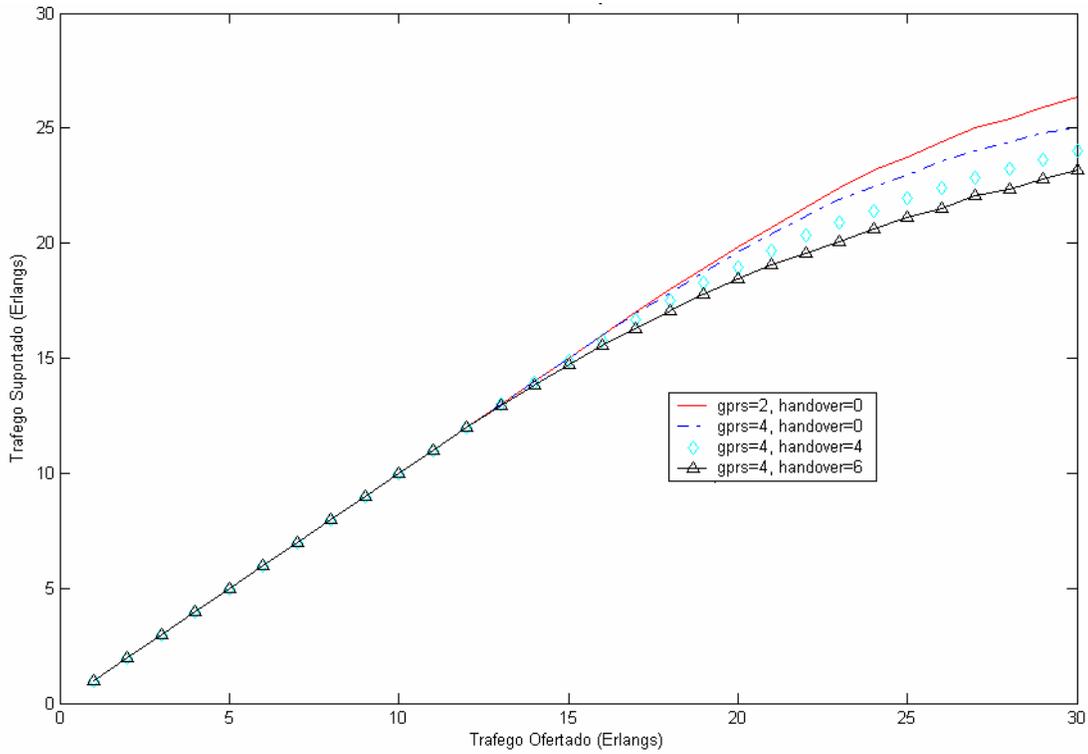


Figura 6.31 – Carga da Rede

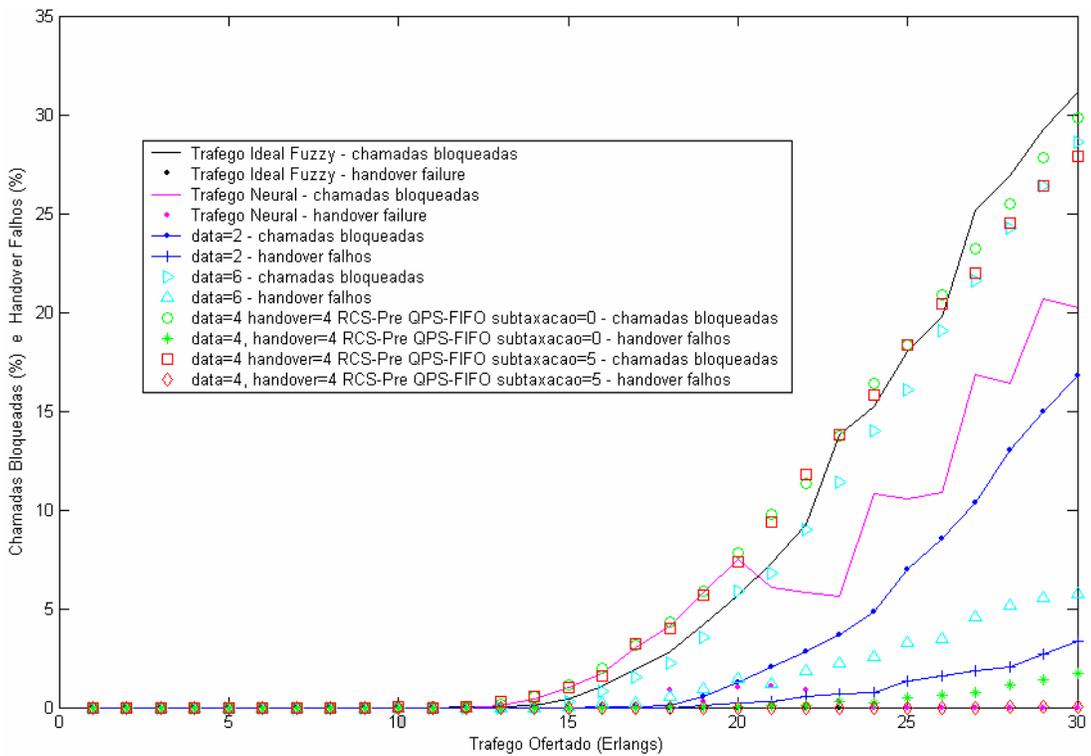


Figura 6.32 – Comparação entre as várias taxas de perdas

6.7.3 Análise dos Resultados

As tabelas 6.2 e 6.3 comparam as técnicas de inteligência com os FCAs. Na maioria dos casos, as técnicas de inteligência atingiram, aproximadamente, 3% de melhoria em relação às técnicas FCAs comparadas. Percentuais negativos demonstram situações pontuais em que as técnicas FCAs superaram as inteligentes. Entretanto, se o intervalo de análise for discretizado em sub-intervalos menores, observa-se que o melhor seria garantir a alternância entre diferentes metodologias, conforme o perfil do tráfego no instante de observação, predominando, em alguns casos, o uso das técnicas inteligentes no processo de inferência.

Tabela 6.2 - Avaliação dos Resultados do Gráfico 6.14 (Micro-célula)

	<i>Fuzzy</i>	Neural
Data channels = 2	-0.5340 %	1.1989 %
Data channels = 6	5.6190 %	7.4590 %
Data channels = 4 Handover = 4 RCS-Pre QPS-FIFO Sub-rated channels = 0	2.3126 %	4.0950 %
Data channels = 4 Handover = 4 PCS-Pre QPS-FIFO Sub-rated channels = 5	-0.7578 %	0.9712 %

Tabela 6.3 - Avaliação de Resultados do Gráfico 6.27 (Macro-Célula)

	<i>Fuzzy</i>	Neural
Data channels = 2	-0.2405 %	-0.3061 %
Data channels = 6	4.1169 %	4.0484 %
Data channels = 4 Handover = 4 RCS-Pre QPS-FIFO Sub-rated channels = 0	4.5033 %	4.4346 %
Data channels = 4 Handover = 4 PCS-Pre QPS-FIFO Sub-rated channels = 5	4.0751%	4.0066 %

Os resultados também demonstram que a inferência neural na alocação dinâmica de canais obteve melhores resultados que a inferência *fuzzy*. Isto ocorreu porque há um pré-

processamento neural que seleciona os melhores valores de alocação de recursos disponibilizados em cada conjunto de pontos de carga suportada (em Erlangs). Enquanto isso, a lógica *fuzzy* faz uso de um controlador ajustado de acordo com o conhecimento do engenheiro de rede, que traz junto suas imperfeições sobre o problema.

Uma comparação entre os dois cenários é apresentada na tabela 6.4, onde todos os valores são obtidos para uma carga de célula ofertada de 21 Erlangs. Para $N_{\text{gprs}} = 2$, o aumento em P_{nb} é desconsiderado por causa do benefício de se reservar mais 2 canais para dados do usuário, principalmente na Macro-célula, conforme [104]. Para $N_{\text{gprs}} = 4$ e 6, observa-se um aumento mais considerável no P_{nb} para o cenário da Micro-célula, especificamente no cenário pós-RCS. Como solução, o uso de uma Macro-célula sobreposta deve vir a ser considerada. Além do mais, quando $N_{\text{gprs}} = 6$ é necessário, isto é, quando o fator de penetração do GPRS é muito alto, os valores de P_{hf} e P_{nb} aumentam para níveis inaceitáveis, não importando o esquema de *handover* usado, nem a presença de uma macro-célula. Portanto, nesse caso, faz-se necessário aumentar o número de TRX ou uma nova ERB para manter a qualidade na transmissão GSM/GPRS.

A tabela 6.4, a seguir, apresenta alguns valores coletados do simulador por meio do processo de simulação FCA, de forma similar ao extraído em [12]. Todas as amostras de valores obedecem à equação 6.8, quando o percentual de tráfego de handover é de 35% e o de novas chamadas é de 65%, durante a janela de monitoramento de 100.000 time-slots atualizados.

A tabela 6.6 apresenta a otimização obtida, também obedecendo aos mesmos princípios da equação 6.8.

Tabela 6.4 – Parâmetros de Desempenho FCA para os cenários, com $A_{\text{off}} = 21$ Erlangs.

N_{gprs}	Cenário Micro-célula				Cenário Macro-célula			
	Esquema de Alocação de Canais	P_{hf} (%)	P_{nb} (%)	A_{carr} (Erlangs)	Esquema de Alocação de Canais	P_{hf} (%)	P_{nb} (%)	A_{carr} (Erlangs)
0	NPS	2,00	2,00	20,58	NPS	1,10	1,11	20,77
	NPS	6,68	6,68	19,59	NPS	3,99	4,73	20,06
	$N_{\text{ho}}=1$	6,33	6,88	19,59	$N_{\text{ho}}=1$	3,90	4,85	20,05

4	pré-RCS	N _{ho} =2	6,00	7,18	19,57	pré-RCS	N _{ho} =2	3,60	5,12	20,04
		N _{ho} =3	5,81	7,54	19,54		N _{ho} =3	3,42	5,47	20,00
		N _{ho} =4	5,48	8,10	19,49		N _{ho} =4	3,15	5,85	19,97
	pós-RCS	N _{ho} =1	4,74	8,48	19,49	pós-RCS	N _{ho} =1	2,62	6,20	19,96
		N _{ho} =2	3,32	10,64	19,30		N _{ho} =2	1,72	8,02	19,78
		N _{ho} =3	2,25	13,06	19,05		N _{ho} =3	1,07	10,24	19,52
		N _{ho} =4	1,53	15,61	18,75		N _{ho} =4	0,68	12,62	19,23
	QPS	FIFO	2,11	8,44	19,69	QPS	FIFO	1,26	5,68	20,13
		MBP	1,94	8,46	19,70		MBP	1,20	5,74	20,13
	SRS	N _{sub} =1	3,47	6,68	19,83	SRS	N _{sub} =1	1,94	4,73	20,22
		N _{sub} =2	1,72	6,68	19,96		N _{sub} =2	0,88	4,73	20,29
		N _{sub} =3	0,77	6,68	20,03		N _{sub} =3	0,43	4,73	20,32
		N _{sub} =4	0,33	6,68	20,06		N _{sub} =4	0,22	4,73	20,35
	NPS		3,82	3,82	20,19	NPS		2,14	2,42	20,51
	pré-RCS	N _{ho} =2	3,58	4,11	20,17	pré-RCS	N _{ho} =2	1,95	2,68	20,49
		N _{ho} =4	3,17	4,58	20,14		N _{ho} =4	1,72	3,07	20,45
pré-RCS	N _{ho} =2	1,72	6,67	19,96	pós-RCS	N _{ho} =2	0,88	4,72	20,29	
	N _{ho} =3	1,17	8,50	19,75		N _{ho} =3	0,54	6,20	20,11	
QPS	FIFO	1,10	4,75	20,26	QPS	FIFO	0,67	2,94	20,55	
	MBP	1,08	4,82	20,26		MBP	0,65	2,96	20,55	
SRS	N _{sub} =2	0,85	3,82	20,41	SRS	N _{sub} =2	0,44	2,42	20,64	
	N _{sub} =4	0,13	3,82	20,46		N _{sub} =4	0,14	2,42	20,67	
NPS		10,56	10,56	18,78	NPS		6,72	8,04	19,41	
pré-RCS	N _{ho} =2	9,60	11,51	18,72	pré-RCS	N _{ho} =2	6,00	8,90	19,34	
	N _{ho} =4	8,51	12,83	18,62		N _{ho} =4	5,13	10,10	19,24	
pré-RCS	N _{ho} =2	5,78	15,65	18,43	pré-RCS	N _{ho} =2	3,14	12,67	19,04	
	N _{ho} =3	2,87	18,43	18,18		N _{ho} =3	2,06	15,24	18,77	
QPS	FIFO	3,54	13,47	18,89	QPS	FIFO	2,23	9,81	19,50	
	MBP	3,39	13,51	18,90		MBP	2,14	9,85	19,50	
SRS	N _{sub} =2	3,32	10,56	19,30	SRS	N _{sub} =2	1,72	8,04	19,78	
	N _{sub} =4	0,71	10,56	19,50		N _{sub} =4	0,38	8,04	19,88	

Tabela 6.5 – Parâmetros de Desempenho usando Inteligência Artificial para ambos os cenários, com $A_{off} = 21$ Erlangs.

Método	Cenário Micro-célula			Cenário Macro-célula				
	Esquemas de Alocação de Canais	Phf(%)	Pnb(%)	Acarr (Erlangs)	Esquema de Alocação de Canais	Phf (%)	Pnb (%)	Acarr (Erlangs)
<i>Fuzzy</i>	4 _{GPRS} /5 _{Handover} /Pós-RCS/FIFO	1,2	12,43	19,21	4 _{GPRS} /5 _{Handover} /Pós-RCS/FIFO	0,98	7,28	19,99
Neural	3 _{GPRS} /3 _{Handover} /Pós-RCS/FIFO	1,63	7,22	20,00	3 _{GPRS} /3 _{Handover} /Pós-RCS/FIFO	1,10	6,08	20,10

6.8 CONSIDERAÇÕES

A redução da capacidade dos canais de voz é desprezível quando a quantidade de canais de dados não supera duas unidades. Isso, devido ao baixo fator de penetração dos dados, nessas circunstâncias, em relação à voz.

O uso de técnicas de alocação de canais ganha importância quando o fator de penetração dos dados é médio, ou seja, aproximadamente quatro canais nas circunstâncias do modelo de simulação.

Para um alto fator de penetração do número de canais de dados, a capacidade da redução do tráfego de voz é demasiadamente alta. Dessa forma, a ocupação de um alto número de canais de dados deve ocorrer apenas quando o tráfego de voz é pequeno.

7 - CONCLUSÃO

Por conta da evolução tecnológica e do aumento do número de novos dispositivos móveis vendidos e consumidos pelo mercado, é necessária uma melhor eficiência das redes telefônicas atuais. O estudo de caso desta tese analisou o comportamento do tráfego conduzido pelos canais telefônicos presentes nas estações rádio-base. A análise foi realizada sobre o ponto de vista da alocação dos canais telefônicos para os diferentes tipos de tráfego de dados e voz, levando-se em conta a priorização do tráfego de *handover*.

Esta avaliação está focada, principalmente, nas malhas telefônicas de 2½G e 3G, devido à necessidade atual em garantir o compartilhamento dos recursos para os diferentes perfis de tráfego.

A transmissão de tráfego digital nesses meios, e as demandas por serviços novos e melhores nessas malhas, obrigam a uma melhoria em todo o sistema de gerenciamento telefônico, permitindo a adequação de novos modelos e técnicas.

No contexto das metas propostas como objetivos deste trabalho estavam o desenvolvimento de uma plataforma integrada para a simulação de redes móveis de comunicação, a simulação e o estudo de desempenho de tráfego ofertado e suportado, o desenvolvimento de técnicas de otimização do tráfego suportado por meio da alocação dinâmica de canais de voz e de dados e, a comparação de desempenho entre o modelo estático (atual) e o dinâmico (proposto). Neste sentido, foi formulado um ambiente de gerenciamento adaptativo baseado na integração da malha telefônica com a rede de Internet. O modelo de comunicação proposto foi validado sobre a pilha de protocolos de Internet TCP/IP e SOAP.

No escopo da proposta foi apresentado o projeto de um sistema de gerenciamento distribuído (Capítulo 5), na área do gerenciamento de desempenho em redes telefônicas, com suporte à distribuição através de tecnologias portáteis, distribuídas, escalares e heterogêneas.

Na alocação dinâmica de canais de comunicação foram utilizadas técnicas de inteligência artificial como redes neurais artificiais e lógica *fuzzy*. A análise inteligente da alocação dos canais telefônicos nas ERBs apresenta uma abordagem que permitirá a antecipação no pro-

cesso de tomada de decisões sobre um provável comportamento danoso do tráfego com possibilidade de avaliação de figuras de mérito associadas ao desempenho em “tempo real”, bem como, pode orientar uma expansão eficiente de uma rede de comunicação móvel.

Esta abordagem pró-ativa caracteriza-se pelo fato do agente de gerenciamento poder prever um problema, utilizando técnicas de inteligência artificial do tipo lógica *Fuzzy* ou mesmo através de redes neurais, durante a fase de análise dos dados sobre os parâmetros mensurados. O modelo apresentado tem a capacidade de solucionar os problemas de desempenho encontrados; permitindo, dessa forma, a reconfiguração da alocação de canais, adaptando-os dinamicamente ao novo contexto temporal.

A Lógica *Fuzzy* e a Rede Neural Artificial são usadas separadamente durante o processo de análise dos dados, levando a adoção dos resultados do método que resultar na melhor configuração para os recursos utilizados.

A determinação das funções de pertinência dos conjuntos *Fuzzy* é de natureza subjetiva e, geralmente, dependente do contexto no qual está inserido. Contudo, essa determinação não é arbitrária, visto que a função de pertinência tem que ser consistente com o perfil de comportamento dos valores monitorados. O emprego da lógica *Fuzzy* neste trabalho, por meio de controladores, facilita o mapeamento do conhecimento do administrador para inferência dos recursos telefônicos. Por outro lado, a Rede Neural adota uma topologia baseada no conexionismo dos vários elementos processados.

Neste trabalho, a forma de treinamento das redes neurais permitiu que ela apresentasse resultados um pouco melhores que os obtidos por meio da lógica *Fuzzy*, dentre os testes efetuados. Além disso, seu treinamento foi idealizado através de perturbações (ruídos) nas configurações de alocações de canais dos vários pontos de carga ofertada da rede. Nestes pontos, foram usados como base de treinamento e teste as melhores configurações de tráfego em cada ponto de carga ofertada.

Diferentemente da lógica *Fuzzy*, em que a inferência é feita sobre toda a base de conhecimento simulada, o treinamento neural é baseado nesses melhores pontos de tráfego; isso permitiu maior robustez à rede neural. Daí os resultados neurais tenderem a melhores resultados que os obtidos pela lógica *Fuzzy*.

Infelizmente, é virtualmente impossível obter-se as cadeias de inferência lógica das redes neurais, ou seja, não é possível a introspecção sobre as redes neurais. Diferentemente, o conhecimento *Fuzzy* está textualmente apresentado no apêndice D deste trabalho.

Por ser o tráfego telefônico um problema de otimização dinâmica com restrições em grande escala em um ambiente estocástico, os valores coletados e analisados no Capítulo 6 servem para que o administrador ou gerente da malha telefônica possa estabelecer os melhores critérios para automatização do processo de análise comportamental da rede.

Este trabalho contribuiu para o gerenciamento pró-ativo telefônico nas seguintes dimensões:

- na definição, formulação, modelagem e implementação de uma plataforma distribuída (ambiente distribuído), que proporcione a heterogeneidade de máquinas e a modularização dos processos em pontos distribuídos, onde tais processos são associados a protocolos de interface WEB, conforme demonstrado nos Capítulos 4 e 5;
- na proposição de um módulo de inteligência com características de modularização, permitindo seu acoplamento a outras arquiteturas telefônicas. Este módulo foi descrito no Capítulo 5 e sua análise perfaz todos os demais capítulos e alguns dos apêndices;
- na utilização de ferramentas computacionais usando técnicas de inteligência artificial (Lógica *Fuzzy* e Rede Neurais Artificiais) como metodologia para o gerenciamento do tráfego telefônico;
- na adaptação do simulador de tráfego telefônico para redes móveis, escrito em linguagem de programação C++, com interfaces de distribuição implementadas em linguagem de programação Java;
- na geração de arquivos de atributos da rede com relação ao cenário simulado;
- na modelagem de uma MIB proprietária, para permitir o acesso a valores de inferência do módulo de inteligência. Como esta MIB foi modelada usando a linguagem ASN.1, suas informações podem ser facilmente utilizadas por outras plataformas de gerenciamento de redes de comunicação.

7.1 - TRABALHOS FUTUROS

Este trabalho de modelagem (conceitual e físico) de uma plataforma para a monitoração, avaliação e controle de sistemas de comunicação móveis pode ser precursor de várias outras plataformas para as áreas de redes de computadores, de telefonia fixa e móvel.

Simulações e testes não provam a correte e completude matemática de um sistema. Para garantir a validação deste modelo para casos além daqueles aplicados nos testes simulados, técnicas de especificação formal de protocolos devem ser estudadas e aplicadas ao modelo proposto.

O banco de dados desenvolvido nesse modelo e feito em linguagem ASN.1 é uma forma de especificar os dados do modelo. Entretanto, é importante o uso de técnicas que especifiquem formalmente todo o modelo telefônico. A ferramenta de especificação *Estelle* pode ser uma forma viável de garantir a modelagem em ambiente telefônico, conforme feito em [105].

Outra forma de validação, que também pode ser abordada, é a implantação deste modelo em um sistema de telefonia real. Em cada operadora telefônica, podem surgir necessidades bastante específicas que obriguem a adequação da plataforma distribuída ao contexto em particular. Assim, novas formas de modularização, novos geradores de tráfego, outros fluxos de condução das informações e uma infinidade de outros atributos de monitoramento da rede podem surgir para dar maior consistência, modularização e portabilidade ao modelo. Tudo isso pode tornar o modelo verdadeiramente adaptável ao rápido processo evolutivo que permeia o mercado de telefonia móvel mundial.

O resultado dessa pesquisa poderá ser usado também como pressuposto para geração de outros algoritmos meta-heurísticos para gerenciamento de recursos de rádio mais eficientes, além de auxiliar em problemas de redimensionamento da malha vigente por meio de automação da simulação feita em tempo real de processamento.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Oliberal. http://www.oliberal.com.br/plantão/noticia/default.asp?id_noticia=98203. URL visitada pela última vez em setembro, 2005.
- [2] GSM World <http://www.gsmworld.com>. URL visitada pela última vez em fevereiro, 2004.
- [3] Colcher, S., Soares, L. F. G. *Um Meta Modelo para a Engenharia de Serviços de Telecomunicações*. Em XVIII Simpósio Brasileiro de Redes de Comunicações – SBRC'2000, Belo Horizonte, Minas Gerais, pp. 441-456, 2000.
- [4] Erl, T. *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Prentice Hall. ISBN 0131858580, 2005.
- [5] Object Management Group (OMG). *The Common Object Request Broker, Architecture and Specification*. John Wiley & Sons. Revision 2.2. New York, 1998.
- [6] ISO/IEC JTC1/SC2/WG7. ISO/IEC DIS 10746-1/ITU-T X.901. *Reference Model of Open Distributed Processing*, Part 1: Overview, May, 1995.
- [7] Dornan, A. *Wireless Communication: O Guia Essencial de Comunicação Sem Fio*, Editora Campus, 2001.
- [8] Lin, Y. B., Chlamtac, I. *Wireless and Mobile Network Architectures*, John Wiley & Sons INC, 2001.
- [9] EIA/TIA. *Cellular System Dual-Mode Mobile Station-Base Station Compatibility Standard*, Technical Report IS-54, 1992.
- [10] GSM03.34 *Digital cellular telecommunications system (Phase 2+); High Speed Circuit Switched Data (HSCSD); Service Description, Stage 2*, ETSI TC-SMG, v.7.0.0, 1999.

- [11] Stuchmann, P. *The GSM Evolution: Mobile Packet Data Services*, Aachen University, Germany, Ltd John Wiley & Sons, 2003.
- [12] Ventura, J. *Impact of GPRS on existing GSM services*, Networking Laboratory, Helsinki University of Technology, Thesis, 2002.
- [13] Hoff, S., Meyer, M., Sachs, J. *Analysis of the General Packet Radio Service (GPRS) of GSM as Access to the Internet*, Ericsson Eurolab Deutschland, Germany, IEEE, 1998.
- [14] Dziong, Z. *ATM Network Resource Management*, McGraw-Hill, 1997.
- [15] Guerin, R., Ahmadi, H., Naghshineh, M. *Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks*, IEEE Journal on Selected Areas on Communications, 9(7), pp. 868-881, 1991.
- [16] SOA Vision SOA Systems: Service-Oriented solutions for the real world. The SOA Vision: Service-Oriented Computing. Disponível em: <http://www.soasystems.com/soa1.asp>, em maio de 2006.
- [17] Bioinformatics. *Web Services for Bioinformatics*. O'Reilly Webservices.com.br. Disponível em <http://www.xml.com/pub/a/ws/2002/05/14/biows.html>, 2006.
- [18] Obf (2006). *Open Bioinformatics Foundation*. Disponível em http://www.open-bio.org/wiki/Main_Page.
- [19] Stein, L. *Creating a bioinformatics nation*. *Nature* **417**, 119 – 120. 09 May, 2002.
- [20] Telecom Web Services Toolkit Preview. *An integrated development environment for Telecom-enabled applications through Web Services for WebSphere Studio Application Developer 5.1.2*, Disponível em <http://www.alphaworks.ibm.com/tech/twst>, 2006.
- [21] Ancelmo, M. *Web Services*. Material de curso ministrado na DRACO Consultoria, 2006.

- [22] Pree, W. *Framework Patterns. SIGS Management Briefings*, SIGS Books & Multimedia, 1996.
- [23] Christensen, G., Florack, P. G., Duncan, R. *Wireless Intelligent Networking. Mobile Communications Series*, Artech House Publishers, 2001.
- [24] Lazar, A. A. *Programming Telecommunication Networks*. IEEE Network. pp. 8-18. Setembro, 1997.
- [25] Yemini, Y., Silva, S. D. *Towards Programmable Networks*, In proceedings of the IFIP/IEEE Workshop on Distributed Systems: Operations and Management. L'Aquila. October, Italy, 1996.
- [26] Psounis, K. *Active Networks: Applications, Security, Safety and Architectures*. IEEE Communications Surveys. (First Quarter):2-16, 1999.
- [27] Tennenhouse, D. L., Smith, J. M., Sincoskie, W. D., Wetherall, D. J., Minden, G. A. *Survey of Active Network Research*, IEEE Communications, January, 1997.
- [28] Campbell, A. T., Kounavis, M. E., Meer, H. D., Miki, K., Vicente, J. *A Survey of Programming Networks*. ACM Computer Communications Review. Abril, 1999.
- [29] Gamma, E., Helm, R., Johnson, R. Vlissides, J. *Design Patterns – Elements of Reusable Object-Oriented Software*, Addison-Wesley, July, 2001.
- [30] Colcher, S., Soares, L. F. G. *Modelo de Referência Unificado para Arquitetura de Protocolos e Programação de Aplicações Multimídia*. Em XVI Simpósio Brasileiro de Redes de Comunicações – SBRC'98, editado por J. Leite e E. S. Silva, Rio de Janeiro, RJ, pp. 631-650, maio, 1998.
- [31] Gomes, A. T. A., Colcher, S., Soares, L. F. G. *Uma Plataforma para Provisão de QoS em ambientes genéricos de processamento e comunicação*. Em XVII Simpósio Brasileiro de Redes de Computadores – SBRC'99, pp. 307-322, Salvador, BA, Brasil, junho, 1999.

- [32] Rodrigues, M. A. A., Colcher, S., Soares, L. F. G. *Um Framework para a Provisão de Serviços de Multicast em Ambientes Genéricos de Comunicação de Dados*, Em XVIII Simpósio Brasileiro de Redes de Computadores – SBRC’99, pp. 206-221, Salvador, BA, Brasil, junho, 1999.
- [33] Blair, G. S. Coulson, G. *The Case for Reflective Middleware*, relatório técnico MPG-98-38, distributed multimedia research group, 1998.
- [34] Costa, F. M., Blair, G. S., Coulson, G. Experiments with Reflective Middleware. In Proceedings of the ECOOP’98 Workshop on Reflective Object Oriented Programming and Systems. Springer Verlag, 1998.
- [35] Coulson, G., Waddington, D. G. *A CORBA Compliant Real-Time Multimedia Platform for broadband Networks*. In Proceedings of TRENDS’96. Germany. September, 1996.
- [36] Busmman, L. A. S. *Uma Arquitetura para Gerência de Sistemas Corporativos*. Dissertação de Mestrado. Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial do Centro Federal de Educação Tecnológica do Paraná, Março, 1998.
- [37] Case, J. D., Fedor, M. Schoffstal, M. L. *RFC 1157: Simple Network Management Protocol – SNMP*. IETF - Network Working Group, 1990.
- [38] *DMTF - Desktop Management Task Force*. Desktop Management Interface Specification – Version 2.0, Hillsboro – OR, 1996.
- [39] ISO 7498-4. *Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management Framework*, Geneva, 1989.
- [40] ISO 10040. *Information Processing Systems – Open Systems Interconnection – Systems Management Overview*, Geneva, 1992.
- [41] *JMAP - Java Management API Architecture*. <http://java.sun.com/jmapi-arch.html>. Sun Microsystems, 2006.

- [42] SUN Portifólio de Gerenciamento SOLARIS / JAVA. Disponível em:
[HTTP://www.s2.com.br/scripts/release_print.asp?clienteID=37&releaseID=18532](http://www.s2.com.br/scripts/release_print.asp?clienteID=37&releaseID=18532), 2005.
- [43] Wade, V., Lewis, D. Malbon, C., Richardson, T., Sorensen, L., Stathopoulos, C.,
Component Integration Technologies for Telecoms Management Systems,
<http://64.233.161.104/search?q=cache:XucOwpXqeukJ:https://www.cs.tcd.ie/publications/tech-reports/reports.99/TCD-CS-1999-27.pdf+Component+Integration+Technologies+for+Telecoms+Management+Systems&hl=pt-BR>, URL visitada pela última vez em junho, 2005.
- [44] Booch, G., Rumbaugh, J., Jacobson, I. *The Unified Modeling Language User Guide*, Addison Wesley, ISBN 0-201-57168-4, 1999.
- [45] Cavalcanti, J. W. A. *Uma Plataforma de Suporte à Execução de Aplicações em Ambientes de Computação Móvel*, Dissertação de Mestrado, Centro de Ciências Exatas e da Natureza, Departamento de Informática, Universidade Federal de Pernambuco, Recife, PE, 1997.
- [46] Soares, L. F. G., Lemos, G., Colcher, S. *Redes de Computadores – Das LANs, MANs e WANs às Redes ATM*. Editora Campus, Segunda Edição, 1995.
- [47] Da Rocha, C. A. C. *SNPI: Uma Abordagem para Implantação de Nós Programáveis em Redes de Computadores*, Dissertação de Mestrado, Departamento de Ciência da Computação, Universidade Federal do Ceará, Fortaleza, Ceará, 11 de Março, 2002.
- [48] WEB Service. <http://java.sun.com/webservices/index.jsp>, consulta feita em 20 de outubro de 2005.
- [49] Feit, S. *SNMP: A Guide to Network Management*, McGraw-Hill, 1995.
- [50] McCloghrie, K., Rose, M. *Management Information Base for Network Management of TCP/IP based Internets*, RFC 1213, May, 1990.

- [51] Perkins, D. *Understanding SNMP MIBs*, September, 1993.
- [52] Rose, M. T., McCloghrie, K. *Structure and Identification of Management Information for TCP/IP based Internets*, RFC 1155, May, 1990.
- [53] Rose, M. T., McCloghrie, K. *Concise MIB definitions*, RFC 1212, March, 1991.
- [54] Sloman, M. *Network and Distributed Systems Management*, Addison-Wesley, 1994.
- [55] Tanenbaum, A. S. *Redes de Computadores*. Editora Campus, 5ª edição, ISBN 85-352-01572, 1997.
- [56] ISO/IEC 10165-4. *Information technology - Open Systems Interconnection - Structure of management information - Part 4: Guidelines for the definition of managed objects*, 1992.
- [57] Carvalho, E. V. *Gerenciamento Pró-ativo Distribuído baseado em Lógica Fuzzy*. Dissertação de Mestrado. Departamento de Ciência da Computação. Universidade Federal do Ceará. Novembro, 1999.
- [58] Carvalho, E. V., Belchior, A. D., Souza, J. N. *Gerenciamento Pró-ativo Distribuído baseado em Lógica Fuzzy*, SBRC – Simpósio Brasileiro de Redes de Computadores, Salvador, 1999.
- [59] Carvalho, E. V., Belchior, A. D., Souza, J. N. *Distributed Proactive Network Management based on Fuzzy Logic*. ICT'99, vol. 2, ISBN 89-86328-45-3/99, pp. 308-312, Cheju, Korea, 1999.
- [60] Nascimento, A. S. *Desenvolvendo Agentes Inteligentes para a Gerencia Pro-Ativa de Redes ATM*, Dissertação de Mestrado, Departamento de Ciência da Computação, Universidade Federal do Ceará, Março, 1999.
- [61] Rappaport, T. S. *Wireless Communications: Principles and Practice*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

- [62] Jeon, W. S., Jeong, D. G., Choi, C. H. *An Integrated Services MAC Protocol for Local Wireless Communications*, IEEE Trans, Veh. Technol. 47, pp. 352-264, 1998.
- [63] Jeong, D. G., Kim, J. G., Kim, D. *Capacity Analysis of Spectrally Overlaid Multiband CDMA Mobile Networks*, IEEE Trans. Veh. Technol. 47, pp. 798-807, 1998.
- [64] Schwartz, M. *Network Management and Control Issues in Multimedia Wireless Networks*, IEEE Personal Commun. 2, pp. 8-16, 1995.
- [65] Cao, Y., Zhou, B., Li, C. *Admission Control of Integrated Voice and Data CDMA/TDD System considering Assymmetric Traffic and Power Limit*, IEEE PIMRC, pp. 896-900, 2003.
- [66] Cooper, W., Zeidler, J. R., Bitmead, R. R. *Modeling Dynamic Channel Allocation Algorithms in Multi-BS TDD Wireless Networks with Internet-Based Traffic*, IEEE Trans. Veh. Technol. 53(3), pp. 783-804, 2004.
- [67] Holma, H., Toskala, A. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, John Wiley & Sons, New York, pp. 313, 2001.
- [68] Yang, N. *The Third Generation Wireless Network using CDMA air interface*, WCNC 2, pp. 649-653, 1999.
- [69] Baier, A., Fiebig, U. C., Granzow, W. Koch, W. Toder, P., Thielecke, J. *Design Study for a CDMA-Based Third-Generation Mobile Radio System*. IEEE JSAC 12. pp. 733-743, 1994.
- [70] Esmailzadeh, R., Nakagawa, M., Kajiwara, A. *Power Control in Packet Switched Time Division Duplex Direct Sequence Spread Spectrum Communications*. VTC'92. pp. 989-992, 1992.
- [71] Futakata, T., Itoh, M. S., Moon, U. K., Uebayashi, S. *Experimental Performance Results of IMT-2000 CDMA TDD System*, VTC'01 2, pp. 1153-1157, 2001.

- [72] Jeong, D. G., Jeon, W. S. *CDMA/TDD System for Wireless Multimedia Services with Traffic Unbalance between Uplink and Downlink*, IEEE JSAC 17(5), pp. 939-946, 1999.
- [73] Jeong, D. G., Jeon, W. S. *Time Slot Allocation in CDMA/TDD System for Mobile Multimedia Services*. IEEE Commun. Lett. 4(2), 2000.
- [74] Jeong, D. G., Jeon, W. S. *Call Admission Control for CDMA Mobile Communications Systems supporting Multimedia Services*, IEEE Trans. Wireless Commun. 1(4), pp. 649-659, 2002.
- [75] Lin, C., Gitlin, R. D. *Multi-code CDMA Wireless Personal Communications Networks*. ICC'95. Seattle. WA. pp. 1060-1064, 1995.
- [76] Ojanpera, T., Skold, J., Castro, J., Girard, L., Klein, A. *Comparison of Multiple Access Schemes for UMTS*. VTC'97. Phoenix. AZ. pp. 490-494, 1997.
- [77] Chung, Shun-Ping, Chien, Shih-Chin. *Analysis of Multiservice Cellular Networks with Asymmetrical Traffic and Handoff Queue*. Computer Communications. 28. pp. 864-879, 2005.
- [78] Li, W., Alfa, A. S. *Channel Reservation for Handoff Calls in a PCS Network*. IEEE Trans. Veh. Technol. 49(1). pp. 95-104, 2000.
- [79] Ortigoza-Guerrero, L., Aghvami, A. H. *A Prioritized Handoff Dynamic Channel Allocation Strategy for PCS*. IEEE Trans. Veh. Technol. 48 (4). pp. 1203-1215, 1999.
- [80] Koo, I., Kim, K. *Erlangs Capacity of Voice/Data DS-CDMA System with Prioritized Services*. IEICE Trans. Commun. E84-B. pp. 716-726, 2001.
- [81] Bang, J., Tekinay, S., Ansari, N. *Performance Analysis of Cell Switching Management Scheme in Wireless Packet Communication*. GLOBECOM'01 6. pp. 3639-3643, 2001.

- [82] Hong, D., Rappaport, S. S. *Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures*, IEEE Transactions on Vehicular Technology, vol. 35, pp. 77-92, August, 1986.
- [83] Rappaport, S. S. *Blocking, Hand-off and Traffic Performance for Cellular Communication Systems with Mixed Platforms*, IEE Proc.-1 140(5), pp. 389-401, 1993.
- [84] Sivalingam, K. M., Agrawal, P., Kishore, S. *Dynamic Resource Allocation Schemes during Handoff for Mobile Multimedia Wireless Networks*, IEEE JSAC 17, pp. 1270-1283, 1999.
- [85] Chang, J. W., Chung, J. H., Sung, D. K. *Admission Control Scheme for Soft Handoff in DS-CDMA Cellular Systems Supporting Voice and Stream-Type Data Services*. IEEE Trans. Veh. Technol. 51(6). pp. 1445-1459, 2002.
- [86] Pillay, R., Takawira, F. *Performance Analysis of Soft Handoff in CDMA Cellular Networks*, IEEE Trans. Veh. Technol. 50(6), pp. 1507-1517, 2001.
- [87] Su, S. L., Chen, J. Y., Huang, J. H. *Performance Analysis of Soft Handoff in CDMA Cellular Networks*, IEEE JSAC 14(9), pp. 1762-1769, 1996.
- [88] Nie, J., Haykin, S. *A Q-Learning Dynamic Channel Assignment Technique for Mobile Communications Systems*. IEEE Transactions on Vehicular Technology, vol. 48, no. 5, pp. 1676-1687, September, 1999.
- [89] Nie, J., Haykin, S. *A Dynamic Channel Assignment Policy Through Q-Learning*, IEEE Transactions on Neural Networks, Vol. 10, n° 6, pp. 1443-1455, November, 1999.
- [90] Lai, W. K., Coghill, G. G. *Channel Assignment Through Evolutionary Optimization*. IEEE Transactions on Vehicular Technology, Vol. 45, n° 1, pp. 91-96, February, 1996.
- [91] Skapura, David M. *Building Neural Networks*. New York. N. Y. ACM Press. Reading. Mass: Addison-Wesley. pp. 286, 1996.

- [92] Schneider, M., Kandel, A., Langholz, G., Chew, G. *Fuzzy Expert System Tools*, John Wiley & Sons. ISBN 0 471 95867 0, 1996.
- [93] Zadeh, L. A. *Fuzzy Sets, Information and Control*, Vol. 8, pp. 338-353, 1965.
- [94] De Sousa, C. P. *Introdução à Lógica Nebulosa*. Apostila de Aula. Universidade Federal do Ceará. Outubro, 1995.
- [95] Gadelha, J. C. *Controlador Fuzzy-Expert para Fluxo Viário*. Dissertação de Mestrado. Departamento de Engenharia Elétrica. Universidade Federal do Ceará. Julho, 1999.
- [96] Zell, A., Mamier, G., Vogt, M., Mache, N. *SNNS – Stuttgart Neural Network Simulator*. User Manual. Version 4.1. University of Stuttgart. Institute for Parallel and Distributed High Performance. Disponível em <http://www-ra.informatik.uni-tuebingen.de/SNNS/>, 2005.
- [97] Hiramatsu, A. *Handbook of Neural Computing*. Institute of Physics Publishing and Oxford University Publishing. Taylor & Francis. ISBN 0750303123, 1996.
- [98] Haykin, Simon. *Redes Neurais – Princípios e Prática*, Bookman, 2ª edição, 2001.
- [99] GSM 03.64 *Digital cellular telecommunications system (Phase 2+): General Packet Radio Service (GPRS)*. Overall description of the GPRS radio interface. Stage2, Publicação: ETSI TS 101 350, V.8.5.0, 59 pp., august, 2000.
- [100] Zonoozi, M. M., Dassanayake, P. *User mobility modeling and characterization of mobility patterns*, IEEE Journal on Selected Areas in Communications, vol.15, no.7, September, pp.1239-1252, 1997.
- [101] Zonoozi, M. M., Dassanayake, P., Faulkner, M. *Mobility modeling and channel holding time distribution in cellular mobile communication systems*, IEEE Global Telecommunications Conference, vol.1, no.7, November, pp.12-16, 1995.

- [102] Lin, Y.B., Noerpes, A., Harasty, D.J. *The sub-rating channel assignment strategy for PCS hand-offs*, IEEE Transactions on Vehicular Technology, vol.45, iss.1, February, pp. 122-130, 1996.
- [103] Cardona, N. *Digital M3bile Communication Systems: GSM, GPRS, HSCSD, EDGE, UMTS*, course material, Polytechnic University of Valencia, July, 2000.
- [104] Lindemann, C., Thümmeler, A. *Performance analysis of the General Packet Radio Service*, IEEE International Conference on Distribute Computing Systems, pp. 673-680, 2001.
- [105] Souza, W. L. de, Andrade, R. M. C., Araújo, A. J. P. de. *Especificação Formal, em Estelle, de Sistemas de Telefonia*, IX SBT (Simp3sio Brasileiro de Telecomunica33es), S3o Paulo, SP, 02 a 05 de Setembro, pp. 17.3.1 a 17.3.5, 1991.
- [106] Taffener, M., Bonek, E. *Wireless Internet Access over GSM and UMTS*, Springer, Berlin, 2002.
- [107] Ni, S. *Network Capacity and Quality of Service Management in F/TDMA Cellular Systems*, Acta Polytechnica Scandinavica, Electrical Engineering Series No. 105, Espoo, Published by the Finnish Academies of Technology. ISBN 951-666-5666-7, ISSN 0001-6845, pp 69, 2001.
- [108] Lin, Y. B., Mohan, S., Noerpel, A. *Analyzing the trade off between implementation costs and performance: PCS channel assignment strategies for handover and initial access*, IEEE Personal Communications, vol. 1, iss.3, Third Quarter 1994, pp.47, 1994.
- [109] Ho, H., Zhu, Y., Madhavapeddy, S. *Throughput and Buffer Analysis for GSM General Packet Radio Service (GPRS)*, Nortel Networks, IEEE, 1999.
- [110] Begain, K., R3zsa, G. I., Pfening, A., Telek, M. *Performance Analysis of GSM Networks with Intelligent Underlay-Overlay*, proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02), IEEE, Publica33o 1530-1346/02, 2002.

[111] Nokia Telecommunications Web Site. www.nokia.com. URL visitada pela última vez em março, 2004.

[112] Kulavaratharajah, M. D., Aghvami, H. *Teletraffic performance evaluation of micro-cellular personal communication networks (PCNs) with prioritized handoff procedures*, IEEE Transactions on Vehicular Technology, vol. 48, iss.1, January, pp.137-152, 1999.

[113] Lima, G., Ulisses, T. Implementação de Stateful Session Beans, J2EE Brasil – O site da comunidade J2EE no Brasil, disponível em 01/maio em www.j2eebrasil.com.br, 2006.

[114] JBoss.ORG. The Professional Open Source Company. Disponível em 26 de maio em www.jboss.org, 2006.

APÊNDICES

A - LINGUAGEM DE DESCRIÇÃO DE SERVIÇOS WEB - WSDL

A WSDL (Web Service Description Language – Linguagem de Descrição de Serviços Web) é um vocabulário XML usado para descrever serviços WEB. Ela define operações, tipos de dados e informações de ligação. A especificação WSDL pode ser encontrada no endereço <http://www.w3.org/TR/wsdl>.

A WSDL gerada para o protótipo dessa aplicação telefônica encontra-se abaixo. Nela podemos encontrar o prólogo XML e o elemento-raiz do início do arquivo até (/wsdl:types). As declarações de espaço de nomes no elemento-raiz (wsdl:definitions) mostram que todos os elementos não qualificados e os atributos vêm do esquema WSDL. O prefixo *soap* (em xmlns:soapenc) denota tipos do esquema SOAP, enquanto o prefixo *xsd* (em xmlns:xsd) denota tipos da definição W3C XML Schema.

Nas definições de mensagem WSDL (wsdl:message), encontramos mensagens correspondentes a requisições (com sufixo *Request*) e respostas (com sufixo *Response*).

Nas definições WSDL *portType*, uma definição *portType* (wsdl:portType) é equivalente a uma definição de interface. Ela contém uma ou mais definições de operações (wsdl:operation), as quais por sua vez são construídas com base nas definições de mensagem (wsdl:message) do documento.

A interface wsdl:portType define os protocolos por meio dos quais essa interface pode ser acessada. O elemento de ligação cria uma ligação chamada *Binding*, entre *PortType* e SOAP. Dentro dessa ligação WSDL, é definida uma ligação SOAP (wsdlsoap:binding). Como o SOAP pode trabalhar com uma variedade de transportes subjacentes e pode trabalhar de maneira centrada em RPC ou centrada em documento, os atributos em (wsdlsoap:binding) indicam que se trata de uma ligação estilo RPC que usa HTTP.

A operação WSDL é então mapeada em uma operação SOAP, com elementos de entrada e saída (wsdlsoap:body) definidos para mapear a requisição e a resposta.

Por fim, uma instância do serviço é definida no elemento WSDL service (wsdl:service). Um serviço WSDL contém uma lista de elementos (wsdl:port). Cada elemento (wsdl:port) define uma instância específica de um servidor que obedece a uma das ligações (*binding*) WSDL definidas anteriormente.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://sti.doutorado.dee.fat.unb.br"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://sti.doutorado.dee.fat.unb.br"
  xmlns:intf="http://sti.doutorado.dee.fat.unb.br"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns1="http://gerente.TELCOM.sti.doutorado.dee.fat.unb.br"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <schema targetNamespace="http://gerente.TELCOM.sti.doutorado.dee.fat.unb.br"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="Amostra">
        <sequence>
          <element name="blockedCalls" type="xsd:double"/>
          <element name="carriedTraffic" type="xsd:double"/>
          <element name="gprs" type="xsd:double"/>
          <element name="handover" type="xsd:double"/>
          <element name="handoverFailure" type="xsd:double"/>
          <element name="handoverUtilization" type="xsd:double"/>
          <element name="offeredTraffic" type="xsd:double"/>
          <element name="percHO" type="xsd:double"/>
          <element name="percVO" type="xsd:double"/>
          <element name="qps" nillable="true" type="xsd:string"/>
          <element name="rcs" nillable="true" type="xsd:string"/>
          <element name="signaling" type="xsd:double"/>
          <element name="subrated" type="xsd:double"/>
          <element name="totalCanais" type="xsd:double"/>
          <element name="voiceUtilization" type="xsd:double"/>
        </sequence>
      </complexType>
      <complexType name="ArrayOfAmostra">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="tns1:Amostra[]"/>
          </restriction>
        </complexContent>
      </complexType>
    </schema>
  </wsdl:types>
</wsdl:definitions>
```

```

    <schema targetNamespace="http://sti.doutorado.dee.fat.unb.br" x-
        mlns="http://www.w3.org/2001/XMLSchema">
        <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
        <complexType name="FachadaTelefonica">
            <sequence/>
        </complexType>
    </schema>
</wsdl:types>
<wsdl:message name="executaInferenciaMacrocelulaFuzzyResponse">
    <wsdl:part name="executaInferenciaMacrocelulaFuzzyReturn"
        type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMicrocelulaNeuralResponse">
    <wsdl:part name="reconfiguraInferenciaMicrocelulaNeuralReturn"
        type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="executaInferenciaMicrocelulaNeuralRequest">
</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMacrocelulaNeuralRequest">
    <wsdl:part name="amostra" type="tns1:Amostra"/>
</wsdl:message>
<wsdl:message name="executaInferenciaMacrocelulaNeuralRequest">
</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMacrocelulaFuzzyRequest">
    <wsdl:part name="amostra" type="tns1:Amostra"/>
</wsdl:message>
<wsdl:message name="obtemAmostraIdealMicrocelulaFuzzyResponse">
    <wsdl:part name="obtemAmostraIdealMicrocelulaFuzzyReturn"
        type="tns1:Amostra"/>
</wsdl:message>
<wsdl:message name="executaInferenciaMacrocelulaFuzzyRequest">
</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMicrocelulaFuzzyResponse">
    <wsdl:part name="reconfiguraInferenciaMicrocelulaFuzzyReturn"
        type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="obtemAmostraIdealMacrocelulaNeuralRequest">
    <wsdl:part name="trafegoOfertado" type="xsd:double"/>
    <wsdl:part name="percentualHandover" type="xsd:double"/>
</wsdl:message>
<wsdl:message name="getInstanciaResponse">
    <wsdl:part name="getInstanciaReturn" type="impl:FachadaTelefonica"/>
</wsdl:message>
<wsdl:message name="obtemTodosParametrosMicrocelulaFuzzyRequest">
</wsdl:message>
<wsdl:message name="obtemAmostraIdealMacrocelulaFuzzyResponse">
    <wsdl:part name="obtemAmostraIdealMacrocelulaFuzzyReturn"
        type="tns1:Amostra"/>
</wsdl:message>
<wsdl:message name="obtemTodosParametrosMicrocelulaNeuralRequest">

```

```

</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMacrocelulaNeuralResponse">
  <wsdl:part name="reconfiguraInferenciaMacrocelulaNeuralReturn"
    type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="executaInferenciaMicrocelulaFuzzyResponse">
  <wsdl:part name="executaInferenciaMicrocelulaFuzzyReturn"
    type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="executaInferenciaMicrocelulaFuzzyRequest">
</wsdl:message>
<wsdl:message name="obtemTodosParametrosMacrocelulaNeuralRequest">
</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMicrocelulaFuzzyRequest">
  <wsdl:part name="amostra" type="tns1:Amostra"/>
</wsdl:message>
<wsdl:message name="obtemAmostraIdealMicrocelulaNeuralRequest">
  <wsdl:part name="trafegoOfertado" type="xsd:double"/>
  <wsdl:part name="percentualHandover" type="xsd:double"/>
</wsdl:message>
<wsdl:message name="obtemAmostraIdealMicrocelulaFuzzyRequest">
  <wsdl:part name="trafegoOfertado" type="xsd:double"/>
  <wsdl:part name="percentualHandover" type="xsd:double"/>
</wsdl:message>
<wsdl:message name="obtemAmostraIdealMacrocelulaFuzzyRequest">
  <wsdl:part name="trafegoOfertado" type="xsd:double"/>
  <wsdl:part name="percentualHandover" type="xsd:double"/>
</wsdl:message>
<wsdl:message name="obtemAmostraIdealMicrocelulaNeuralResponse">
  <wsdl:part name="obtemAmostraIdealMicrocelulaNeuralReturn"
    type="tns1:Amostra"/>
</wsdl:message>
<wsdl:message name="obtemTodosParametrosMacrocelulaFuzzyRequest">
</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMacrocelulaFuzzyResponse">
  <wsdl:part name="reconfiguraInferenciaMacrocelulaFuzzyReturn"
    type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="reconfiguraInferenciaMicrocelulaNeuralRequest">
  <wsdl:part name="amostra" type="tns1:Amostra"/>
</wsdl:message>
<wsdl:message name="obtemTodosParametrosMicrocelulaNeuralResponse">
  <wsdl:part name="obtemTodosParametrosMicrocelulaNeuralReturn"
    type="tns1:ArrayOfAmostra"/>
</wsdl:message>
<wsdl:message name="executaInferenciaMicrocelulaNeuralResponse">
  <wsdl:part name="executaInferenciaMicrocelulaNeuralReturn"
    type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="obtemTodosParametrosMicrocelulaFuzzyResponse">

```

```

    <wsdl:part name="obtemTodosParametrosMicrocelulaFuzzyReturn"
              type="tns1:ArrayOfAmostra"/>
</wsdl:message>
<wsdl:message name="obtemAmostraIdealMacrocelulaNeuralResponse">
  <wsdl:part name="obtemAmostraIdealMacrocelulaNeuralReturn"
                type="tns1:Amostra"/>
</wsdl:message>
<wsdl:message name="obtemTodosParametrosMacrocelulaNeuralResponse">
  <wsdl:part name="obtemTodosParametrosMacrocelulaNeuralReturn"
                type="tns1:ArrayOfAmostra"/>
</wsdl:message>
<wsdl:message name="executaInferenciaMacrocelulaNeuralResponse">
  <wsdl:part name="executaInferenciaMacrocelulaNeuralReturn"
                type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="getInstanciaRequest">
</wsdl:message>
<wsdl:message name="obtemTodosParametrosMacrocelulaFuzzyResponse">
  <wsdl:part name="obtemTodosParametrosMacrocelulaFuzzyReturn"
                type="tns1:ArrayOfAmostra"/>
</wsdl:message>
<wsdl:portType name="FachadaTelefonica">
  <wsdl:operation name="getInstancia">
    <wsdl:input message="impl:getInstanciaRequest"
                 name="getInstanciaRequest"/>
    <wsdl:output message="impl:getInstanciaResponse"
                       name="getInstanciaResponse"/>
  </wsdl:operation>
  <wsdl:operation name="obtemAmostraIdealMicrocelulaFuzzy"
                  parameterOrder="trafegoOfertado percentualHandover">
    <wsdl:input message="impl:obtemAmostraIdealMicrocelulaFuzzyRequest"
                 name="obtemAmostraIdealMicrocelulaFuzzyRequest"/>
    <wsdl:output message="impl:obtemAmostraIdealMicrocelulaFuzzyResponse"
                  name="obtemAmostraIdealMicrocelulaFuzzyResponse"/>
  </wsdl:operation>
  <wsdl:operation name="obtemAmostraIdealMicrocelulaNeural"
                  parameterOrder="trafegoOfertado percentualHandover">
    <wsdl:input message="impl:obtemAmostraIdealMicrocelulaNeuralRequest"
                 name="obtemAmostraIdealMicrocelulaNeuralRequest"/>
    <wsdl:output message="impl:obtemAmostraIdealMicrocelulaNeuralResponse"
                  name="obtemAmostraIdealMicrocelulaNeuralResponse"/>
  </wsdl:operation>
  <wsdl:operation name="obtemAmostraIdealMacrocelulaFuzzy"
                  parameterOrder="trafegoOfertado percentualHandover">
    <wsdl:input message="impl:obtemAmostraIdealMacrocelulaFuzzyRequest"
                 name="obtemAmostraIdealMacrocelulaFuzzyRequest"/>
    <wsdl:output message="impl:obtemAmostraIdealMacrocelulaFuzzyResponse"
                  name="obtemAmostraIdealMacrocelulaFuzzyResponse"/>
  </wsdl:operation>
  <wsdl:operation name="obtemAmostraIdealMacrocelulaNeural"

```

```

        parameterOrder="trafegoOfertado percentualHandover">
<wsdl:input message="impl:obtemAmostraIdealMacrocelulaNeuralRequest"
    name="obtemAmostraIdealMacrocelulaNeuralRequest"/>
<wsdl:output message="impl:obtemAmostraIdealMacrocelulaNeuralResponse"
    name="obtemAmostraIdealMacrocelulaNeuralResponse"/>
</wsdl:operation>
<wsdl:operation name="obtemTodosParametrosMicrocelulaFuzzy">
<wsdl:input message="impl:obtemTodosParametrosMicrocelulaFuzzyRequest"
    name="obtemTodosParametrosMicrocelulaFuzzyRequest"/>
<wsdl:output message="impl:
    obtemTodosParametrosMicrocelulaFuzzyResponse"
    name="obtemTodosParametrosMicrocelulaFuzzyResponse"/>
</wsdl:operation>
<wsdl:operation name="obtemTodosParametrosMicrocelulaNeural">
<wsdl:input message="impl:obtemTodosParametrosMicrocelulaNeuralRequest"
    name="obtemTodosParametrosMicrocelulaNeuralRequest"/>
<wsdl:output message="impl:
    obtemTodosParametrosMicrocelulaNeuralResponse"
    name="obtemTodosParametrosMicrocelulaNeuralResponse"/>
</wsdl:operation>
<wsdl:operation name="obtemTodosParametrosMacrocelulaFuzzy">
<wsdl:input message="impl:obtemTodosParametrosMacrocelulaFuzzyRequest"
    name="obtemTodosParametrosMacrocelulaFuzzyRequest"/>
<wsdl:output message="impl:
    obtemTodosParametrosMacrocelulaFuzzyResponse"
    name="obtemTodosParametrosMacrocelulaFuzzyResponse"/>
</wsdl:operation>
<wsdl:operation name="obtemTodosParametrosMacrocelulaNeural">
<wsdl:input message="impl:obtemTodosParametrosMacrocelulaNeuralRequest"
    name="obtemTodosParametrosMacrocelulaNeuralRequest"/>
<wsdl:output message="impl: obtemTodosParametrosMacrocelulaNeuralRe-
    sponse"
    name="obtemTodosParametrosMacrocelulaNeuralResponse"/>
</wsdl:operation>
<wsdl:operation name="executaInferenciaMicrocelulaFuzzy">
<wsdl:input message="impl:executaInferenciaMicrocelulaFuzzyRequest" na-
    me="executaInferenciaMicrocelulaFuzzyRequest"/>
<wsdl:output message="impl:executaInferenciaMicrocelulaFuzzyResponse"
    name="executaInferenciaMicrocelulaFuzzyResponse"/>
</wsdl:operation>
<wsdl:operation name="executaInferenciaMicrocelulaNeural">
<wsdl:input message="impl:executaInferenciaMicrocelulaNeuralRequest" na-
    me="executaInferenciaMicrocelulaNeuralRequest"/>
<wsdl:output message="impl:executaInferenciaMicrocelulaNeuralResponse"
    name="executaInferenciaMicrocelulaNeuralResponse"/>
</wsdl:operation>
<wsdl:operation name="executaInferenciaMacrocelulaFuzzy">
<wsdl:input message="impl:executaInferenciaMacrocelulaFuzzyRequest" na-
    me="executaInferenciaMacrocelulaFuzzyRequest"/>
<wsdl:output message="impl:executaInferenciaMacrocelulaFuzzyResponse"

```

```

        name="executaInferenciaMacrocelulaFuzzyResponse"/>
</wsdl:operation>
<wsdl:operation name="executaInferenciaMacrocelulaNeural">
  <wsdl:input message="impl:executaInferenciaMacrocelulaNeuralRequest"
    name="executaInferenciaMacrocelulaNeuralRequest"/>
  <wsdl:output message="impl:executaInferenciaMacrocelulaNeuralResponse"
    name="executaInferenciaMacrocelulaNeuralResponse"/>
</wsdl:operation>
<wsdl:operation name="reconfiguraInferenciaMicrocelulaFuzzy"
  parameterOrder="amostra">
  <wsdl:input message="impl:reconfiguraInferenciaMicrocelulaFuzzyRequest"
    name="reconfiguraInferenciaMicrocelulaFuzzyRequest"/>
  <wsdl:output message="impl:reconfiguraInferenciaMicrocelulaFuzzyResponse"
    name="reconfiguraInferenciaMicrocelulaFuzzyResponse"/>
</wsdl:operation>
<wsdl:operation name="reconfiguraInferenciaMicrocelulaNeural"
  parameterOrder="amostra">
  <wsdl:input message="impl:reconfiguraInferenciaMicrocelulaNeuralRequest"
    name="reconfiguraInferenciaMicrocelulaNeuralRequest"/>
  <wsdl:output message="impl:reconfiguraInferenciaMicrocelulaNeuralResponse"
    name="reconfiguraInferenciaMicrocelulaNeuralResponse"/>
</wsdl:operation>
<wsdl:operation name="reconfiguraInferenciaMacrocelulaFuzzy"
  parameterOrder="amostra">
  <wsdl:input message="impl:reconfiguraInferenciaMacrocelulaFuzzyRequest"
    name="reconfiguraInferenciaMacrocelulaFuzzyRequest"/>
  <wsdl:output message="impl:reconfiguraInferenciaMacrocelulaFuzzyResponse"
    name="reconfiguraInferenciaMacrocelulaFuzzyResponse"/>
</wsdl:operation>
<wsdl:operation name="reconfiguraInferenciaMacrocelulaNeural"
  parameterOrder="amostra">
  <wsdl:input message="impl:reconfiguraInferenciaMacrocelulaNeuralRequest"
    name="reconfiguraInferenciaMacrocelulaNeuralRequest"/>
  <wsdl:output message="impl:reconfiguraInferenciaMacrocelulaNeuralResponse"
    name="reconfiguraInferenciaMacrocelulaNeuralResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="FachadaTelefonicaSoapBinding"
  type="impl:FachadaTelefonica">
  <wsdlsoap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getInstancia">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getInstanciaRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="getInstanciaResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

```

</wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtemAmostraIdealMicrocelulaFuzzy">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="obtemAmostraIdealMicrocelulaFuzzyRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="obtemAmostraIdealMicrocelulaFuzzyResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtemAmostraIdealMicrocelulaNeural">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="obtemAmostraIdealMicrocelulaNeuralRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="obtemAmostraIdealMicrocelulaNeuralResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtemAmostraIdealMacrocelulaFuzzy">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="obtemAmostraIdealMacrocelulaFuzzyRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="obtemAmostraIdealMacrocelulaFuzzyResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtemAmostraIdealMacrocelulaNeural">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="obtemAmostraIdealMacrocelulaNeuralRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="obtemAmostraIdealMacrocelulaNeuralResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtemTodosParametrosMicrocelulaFuzzy">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="obtemTodosParametrosMicrocelulaFuzzyRequest">

```

```

    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="obtemTodosParametrosMicrocelulaFuzzyResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtemTodosParametrosMicrocelulaNeural">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="obtemTodosParametrosMicrocelulaNeuralRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="obtemTodosParametrosMicrocelulaNeuralResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtemTodosParametrosMacrocelulaFuzzy">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="obtemTodosParametrosMacrocelulaFuzzyRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="obtemTodosParametrosMacrocelulaFuzzyResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="obtemTodosParametrosMacrocelulaNeural">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="obtemTodosParametrosMacrocelulaNeuralRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="obtemTodosParametrosMacrocelulaNeuralResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="executaInferenciaMicrocelulaFuzzy">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="executaInferenciaMicrocelulaFuzzyRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="executaInferenciaMicrocelulaFuzzyResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>

```

```

</wsdl:output>
</wsdl:operation>
<wsdl:operation name="executaInferenciaMicrocelulaNeural">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="executaInferenciaMicrocelulaNeuralRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="executaInferenciaMicrocelulaNeuralResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="executaInferenciaMacrocelulaFuzzy">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="executaInferenciaMacrocelulaFuzzyRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="executaInferenciaMacrocelulaFuzzyResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="executaInferenciaMacrocelulaNeural">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="executaInferenciaMacrocelulaNeuralRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="executaInferenciaMacrocelulaNeuralResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="reconfiguraInferenciaMicrocelulaFuzzy">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="reconfiguraInferenciaMicrocelulaFuzzyRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="reconfiguraInferenciaMicrocelulaFuzzyResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="reconfiguraInferenciaMicrocelulaNeural">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="reconfiguraInferenciaMicrocelulaNeuralRequest">

```

```

    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="reconfiguraInferenciaMicrocelulaNeuralResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="reconfiguraInferenciaMacrocelulaFuzzy">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="reconfiguraInferenciaMacrocelulaFuzzyRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="reconfiguraInferenciaMacrocelulaFuzzyResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="reconfiguraInferenciaMacrocelulaNeural">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="reconfiguraInferenciaMacrocelulaNeuralRequest">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:input>
  <wsdl:output name="reconfiguraInferenciaMacrocelulaNeuralResponse">
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      namespace="http://sti.doutorado.dee.fat.unb.br" use="encoded"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="FachadaTelefonicaService">
  <wsdl:port binding="impl:FachadaTelefonicaSoapBinding"
    name="FachadaTelefonica">
    <wsdlsoap:address location="http://localhost:8080/STIServer/services
      /FachadaTelefonica"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

A.1 PARÂMETROS DO SERVIDOR DE SERVIÇOS WEB

Informações gerais da Classe de Serviço:

Nome do Serviço: FachadaTelefonica

Classe or interface referenciada: br.unb.fat.dee.doutorado.sti.FachadaTelefonica

Nome da porta do serviço: FachadaTelefonica

Opções do Servidor:

Estilo do serviço: rpc

Usa: encoded
Versão do SOAP: 1.1
Ação do SOAP: DEFAULT
URL de localização: <http://localhost:8080/STIServer/services/FachadaTelefonica>
Nome do Serviço: FachadaTelefonicaService
Nome da Amarração: FachadaTelefonicaSoapBinding
Nome do tipo de porta: FachadaTelefonica
Escopo de Distribuição: Request

Opções WSDL

Endereço alvo: <http://sti.doutorado.dee.fat.unb.br>

Opções de Saída

Interface de saída do arquivo WSDL: FachadaTelefonica.wsdl

Opções de Endereço:

Nome: http://gerente.TELCOM.sti.doutorado.dee.fat.unb.br

B - DIAGRAMATIZAÇÃO DA ARQUITETURA VIA UML

O Sistema de Telefonia Inteligente – STI foi projetado para uso nas corporações telefônicas objetivando a administração de uma rede interna de telecomunicações.

Nesse sentido a figura B.1 apresenta o modelo de caso de uso da implementação do protótipo. Nela, o administrador da rede telefônica faz uma requisição a uma das funcionalidades do Sistema de Simulação Telefônica Inteligente, recebendo a resposta da requisição, em tela.



Figura B.1 Diagrama de Caso de Uso para acesso direto ao servidor.

A figura B.2 mostra o administrador do sistema acessando a aplicação a partir da interface WEB do cliente. Dessa forma, o administrador pode acessar o sistema, remotamente; não há necessidade de acesso a partir da própria rede local.



Figura B.2 Diagrama de Caso de Uso para acesso a partir da aplicação Cliente de Interface com o serviço WEB.

Para tal, o administrador pode, por meio do protótipo, solicitar:

- ver uma amostra ideal, seja *fuzzy* ou neural;
- ver toda a base de log da amostra ideal, *fuzzy* ou neural;
- executar o gerador de tráfego;
- reconfigurar manualmente o sistema.

Para isso, foram criados os seguintes métodos na classe FachadaTelefonica (figura B.3), do servidor:

FachadaTelefonica	
obtemAmostradealMicrocelulaFuzzy(trafegoOfertado : Double, percentualHandover : Double) : Amostra	
obtemAmostradealMicrocelulaNeural(trafegoOfertado : Double, percentualHandover : Double) : Amostra	
obtemAmostradealMacrocelulaFuzzy(trafegoOfertado : Double, percentualHandover : Double) : Amostra	
obtemAmostradealMacrocelulaNeural(trafegoOfertado : Double, percentualHandover : Double) : Amostra	
obtemTodosParametrosMicrocelulaFuzzy() : Amostra[]	
obtemTodosParametrosMicrocelulaNeural() : Amostra[]	
obtemTodosParametrosMacrocelulaFuzzy() : Amostra[]	
obtemTodosParametrosMacrocelulaNeural() : Amostra[]	
executaInferenciaMicrocelulaFuzzy() : boolean	
executaInferenciaMicrocelulaNeural() : boolean	
executaInferenciaMacrocelulaFuzzy() : boolean	
executaInferenciaMacrocelulaNeural() : boolean	
reconfiguraInferenciaMicrocelulaFuzzy(Amostra : Amostra) : boolean	
reconfiguraInferenciaMicrocelulaNeural(amostra : Amostra) : boolean	
reconfiguraInferenciaMacrocelulaFuzzy(amostra : Amostra) : boolean	
reconfiguraInferenciaMacrocelulaNeural(amostra : Amostra) : boolean	

Figura B.3 Classe br.unb.fat.dee.doutorado.sti.FachadaTelefonica.java

- **obtemAmostradealMicrocelulaFuzzy**: dada uma carga ofertada e um percentual de *handover*, é retornada uma amostra obtida por meio da otimização da inferência sobre a micro-célula usando a lógica *fuzzy*;
- **obtemAmostradealMicrocelulaNeural**: dada uma carga ofertada e um percentual de *handover*, é retornada uma amostra obtida por meio da otimização da inferência sobre a micro-célula usando a rede neural artificial;
- **obtemAmostradealMacrocelulaFuzzy**: dada uma carga ofertada e um percentual de *handover*, é retornada uma amostra obtida por meio da otimização da inferência sobre a macro-célula usando a lógica *fuzzy*;
- **obtemAmostradealMacrocelulaNeural**: dada uma carga ofertada e um percentual de *handover*, é retornada uma amostra obtida por meio da otimização da inferência sobre a macro-célula usando a rede neural artificial;
- **obtemTodosParametrosMicrocelulaFuzzy**: retorna todo o log de resultados de inferência otimizado, extraído a partir da inferência *fuzzy* sobre a micro-célula;
- **obtemTodosParametrosMicrocelulaNeural**: retorna todo o log de resultados de inferência otimizado, extraído a partir da inferência neural sobre a micro-célula;

- `obtemTodosParametrosMacrocelulaFuzzy`: retorna todo o log de resultados de inferência otimizado, extraído a partir da inferência *fuzzy* sobre a macro-célula;
- `obtemTodosParametrosMacrocelulaNeural`: retorna todo o log de resultados de inferência otimizado, extraído a partir da inferência *neural* sobre a macro-célula;
- `executaInferenciaMicrocelulaFuzzy`: executa o módulo de simulação do tráfego sobre uma micro-célula usando otimização fuzzy;
- `executaInferenciaMicrocelulaNeural`: executa o módulo de simulação do tráfego sobre uma micro-célula usando otimização neural;
- `executaInferenciaMacrocelulaFuzzy`: executa o módulo de simulação do tráfego sobre uma macro-célula usando otimização *fuzzy*;
- `executaInferenciaMacrocelulaNeural`: executa o módulo de simulação do tráfego sobre uma macro-célula usando otimização neural;
- `reconfiguraInferenciaMicrocelulaFuzzy`: executa o módulo de simulação do tráfego sobre uma micro-célula usando otimização *fuzzy*;
- `reconfiguraInferenciaMicrocelulaNeural`: executa o módulo de simulação do tráfego sobre uma micro-célula usando otimização neural;
- `reconfiguraInferenciaMacrocelulaFuzzy`: executa o módulo de simulação do tráfego sobre uma macro-célula usando otimização *fuzzy*;
- `reconfiguraInferenciaMacrocelulaNeural`: executa o módulo de simulação do tráfego sobre uma macro-célula usando otimização neural;

Uma amostra (figura B.4) possui informações sobre as configurações de uma carga específica de tráfego.

Amostra (from gerente)	
totalCanais	: double
percVO	: double
percHO	: double
signaling	: double
gprs	: double
handover	: double
subrated	: double
voiceUtilization	: double
handoverUtilization	: double
rcs	: java.lang.String
qps	: java.lang.String
blockedCalls	: double
handoverFailure	: double
offeredTraffic	: double
carriedTraffic	: double

Figura B.4 Classe br.unb.fat.dee.doutorado.sti.TELCOM.gerente.Amostra

- totalCanais: representa o número total de canais de uma ERB;
- percVO: percentual de tráfego de voz;
- percHO: percentual de tráfego de *handover*;
- signaling: número de canais dedicados à sinalização;
- gprs: número de canais dedicados ao tráfego de dados gprs;
- *handover*: número de canais dedicados ao tráfego de *handover*;
- subrated: número máximo de canais sub-taxados;
- voiceUtilization: percentual de utilização de tráfego de voz;
- handoverUtilization: percentual de utilização de tráfego de *handover*;
- rcs: esquema de reserva de canais, do tipo pré-reserva ou pós-reserva;

- qps: esquema de prioridade de canais, do tipo FIFO (do inglês, *First-In First-Out*) ou MBPS (do inglês, *Measured-based Priority Scheme*);
- blockedCalls: percentual de chamadas bloqueadas;
- handoverFailure: percentual de *handover* falhos;
- offeredTraffic: tráfego ofertado (em Erlangs);
- carriedTraffic: tráfego suportado (em Erlangs).

As classes MacrocellBean e MicrocellBean (figura B.5) armazenam a lógica de negócio de cada cenário de simulação.

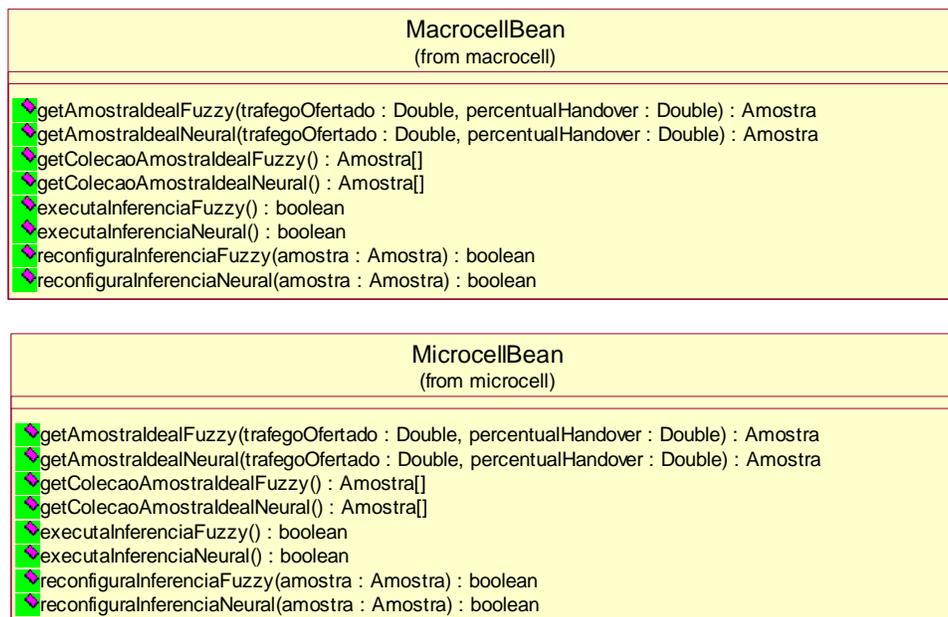


Figura B.5 Classes br.unb.fat.dee.doutorado.sti.TELCOM.gerente.macrocell.MacrocellBean e br.unb.fat.dee.doutorado.sti.TELCOM.gerente.microcell.MicrocellBean

- getAmostradealFuzzy: retorna a amostra ideal inferida sobre o módulo de inferência *fuzzy* do modelo;

- `getAmostraIdealNeural`: retorna a amostra ideal inferida sobre o módulo de inferência neural do modelo;
- `getColecaoAmostraIdealFuzzy`: retorna todas as amostras inferidas sobre o módulo de inferência *fuzzy* do modelo;
- `getColecaoAmostraIdealNeural`: retorna todas as amostras inferidas sobre o módulo de inferência neural do modelo;
- `executaInferenciaFuzzy`: executa o módulo de inferência *fuzzy* do modelo;
- `executaInferenciaNeural`: executa o módulo de inferência neural do modelo;
- `reconfiguraInferenciaFuzzy`: altera os parâmetros de configuração da inferência *fuzzy* garantindo a dinâmica na alocação dos canais;
- `reconfiguraInferenciaNeural`: altera os parâmetros de configuração da inferência neural garantindo a dinâmica na alocação dos canais.

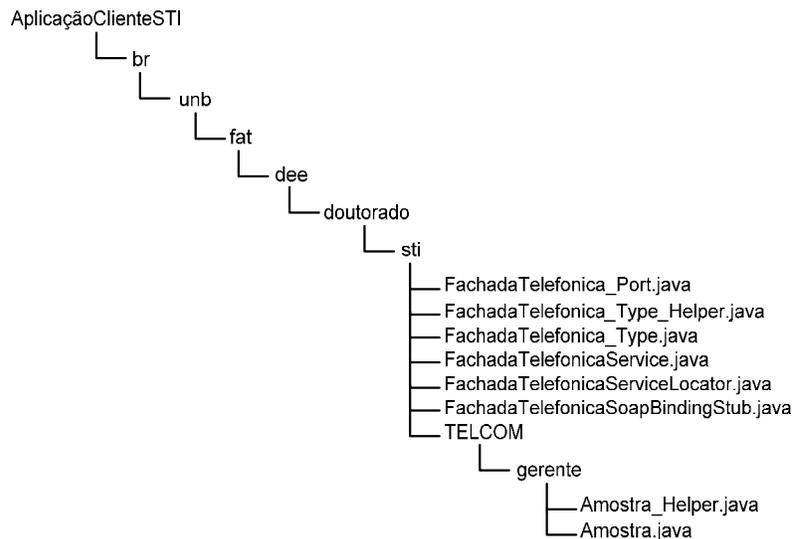


Figura B.6 Estrutura da aplicação cliente para acesso ao serviço.

Por fim, as classes que implementam o serviço WEB do lado do cliente (figura B.6) montam a interface desses serviços por meio da classe `FachadaTelefonica`, do lado do servidor,

anteriormente comentada. Assim, os serviços que são externalizados para o cliente têm a mesma assinatura que os métodos da classe FachadaTelefonica.

A figura B.7 apresenta a estrutura de empacotamento da aplicação do lado servidor. Além desses pacotes e classes há também as classes escritas em linguagem C++ que são responsáveis pela geração do tráfego telefônico simulado.

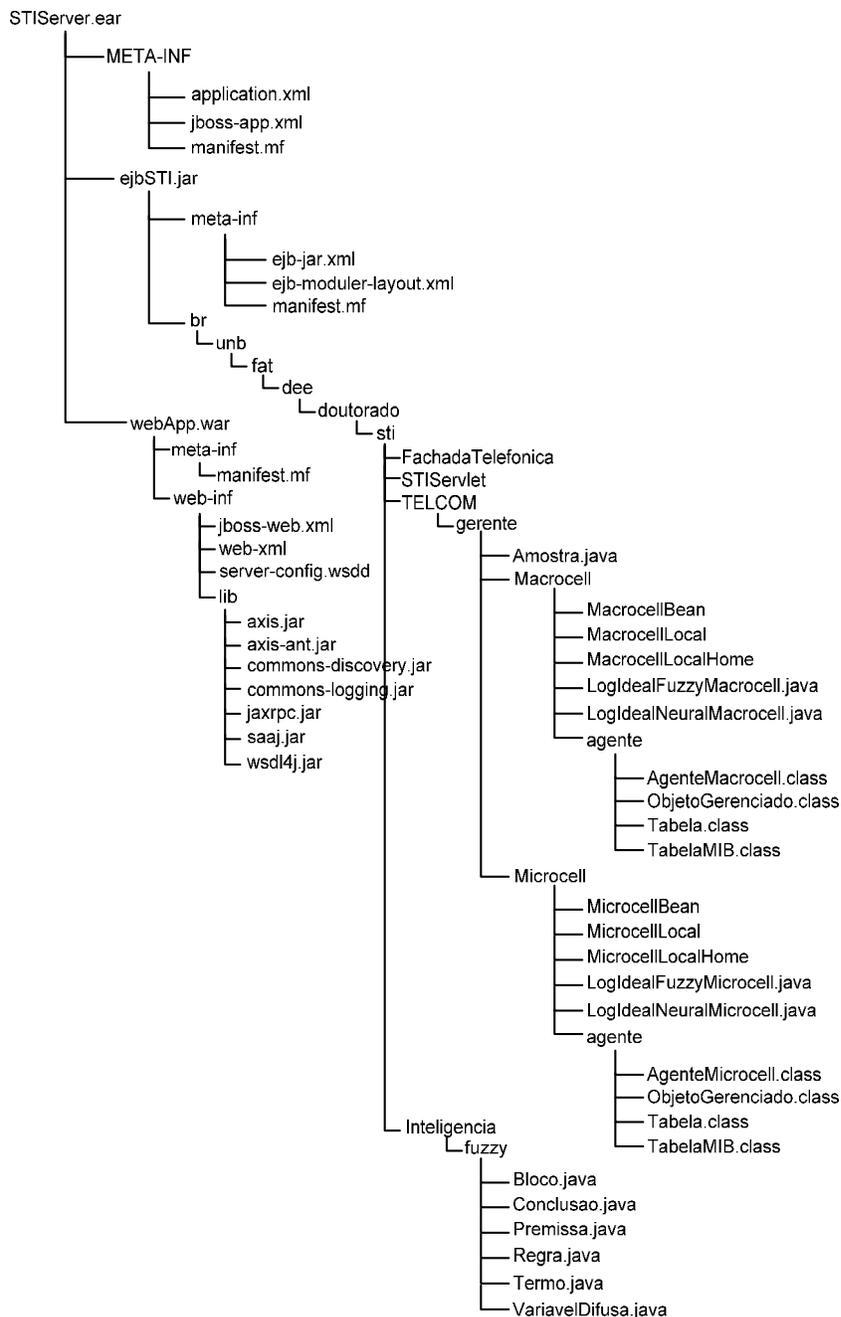


Figura B.7 Estrutura da aplicação servidora que implementa o serviço.

A estrutura de modularização da aplicação do lado servidor foi anteriormente apresentada na figura 6.3.

C – DESCRITORES XML PARA EJB

Os descritores de distribuição da aplicação WEB são documentos que descrevem e configuram um componente. É através deles que se pode configurar o comportamento transacional dos componentes, persistência, e diversos outros recursos providos automaticamente pelo contêiner de aplicação.

Existem dois tipos de descritores: os portáteis e os específicos. Os descritores portáteis (seção C.1) configuram genericamente os componentes da aplicação com comportamentos providos por todos os servidores de aplicação. A mudança de contêiner não afeta a implementação desses descritores.

Os vários aplicativos servidores diferentes possuem recursos e serviços variados. Não existe uma forma de prever quais são esses serviços, nem uma forma de configurá-los genericamente, conforme cita [122], pois essa configuração varia de servidor para servidor.

O descritor geral é armazenado em um arquivo no formato XML chamado ejb-jar.xml (seção C.1). É nesse descritor que os componentes são genericamente configurados.

C.1 - CONTEÚDO DO ARQUIVO EJB-JAR.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ejb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans
2.0//EN" "http://java.sun.com/dtd/ejb-jar_2_0.dtd">
<ejb-jar>
  <display-name>EJBModule</display-name>
  <enterprise-beans>
    <session>
      <ejb-name>Macrocell</ejb-name>
      <local-home>br.unb.fat.dee.doutorado.sti.TELCOM.gerente.macrocell. Macro-
cellLocalHome</local-home>
      <local>br.unb.fat.dee.doutorado.sti.TELCOM.gerente.macrocell. MacrocellLo-
cal</local>
```

```

    <ejb-class>br.unb.fat.dee.doutorado.sti.TELCOM.gerente.macrocell. Macro-
        cellBean</ejb-class>

    <session-type>Stateless</session-type>
    <transaction-type>Container</transaction-type>
</session>
<session>
    <ejb-name>Microcell</ejb-name>
    <local-home>br.unb.fat.dee.doutorado.sti.TELCOM.gerente.microcell. Micro-
        cellLocalHome</local-home>
    <local>br.unb.fat.dee.doutorado.sti.TELCOM.gerente.microcell. MicrocellLo-
        cal</local>
    <ejb-class>br.unb.fat.dee.doutorado.sti.TELCOM.gerente.microcell. Micro-
        cellBean</ejb-class>

    <session-type>Stateless</session-type>
    <transaction-type>Container</transaction-type>
</session>
</enterprise-beans>
<assembly-descriptor>
    <container-transaction>
        <method>
            <ejb-name>Macrocell</ejb-name>
            <method-name>*</method-name>
        </method>
        <trans-attribute>Required</trans-attribute>
    </container-transaction>
    <container-transaction>
        <method>
            <ejb-name>Microcell</ejb-name>
            <method-name>*</method-name>
        </method>
        <trans-attribute>Required</trans-attribute>
    </container-transaction>
</assembly-descriptor>
</ejb-jar>

```

Além do mais, existe também um descritor específico para cada servidor de aplicação, conforme seção seguinte.

C.2 - CONTEÚDO DO ARQUIVO JBOSS.XML

O contêiner servidor da aplicação usado na simulação deste trabalho foi o JBoss [123]. Abaixo, segue a descrição desse arquivo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE jboss (View Source for full doctype...)>
<jboss>
  <enterprise-beans>
    <session>
      <ejb-name>Macrocell</ejb-name>
      <local-jndi-name>MacrocellJNDI</local-jndi-name>
    </session>
    <session>
      <ejb-name>Microcell</ejb-name>
      <local-jndi-name>MicrocellJNDI</local-jndi-name>
    </session>
  </enterprise-beans>
</jboss>
```

O elemento raiz do descritor de aplicação JBoss é a *tag* <jboss>. Debaixo deste esquema de configuração JBoss são encontrados os elementos <enterprise-beans>. Este elemento pode conter múltiplos elementos <session>. Cada um desses elementos pode ter exatamente um elemento <ejb-name> que refere-se a uma declaração correspondente ao padrão de descritor J2EE EJB localizado no arquivo ejb-jar.xml.

C.3 - REFERÊNCIAS A EJBS

A declaração de referências a EJBs é feita para permitir a localização de uma interface EJB. Esta tese usa o seguinte trecho de código para localização das funcionalidades dos serviços da micro-célula e da macro-célula simulada:

```
InitialContext ic = new InitialContext();  
Object lookup = ic.lookup("MicrocellJNDI");  
return (MicrocellLocalHome) PortableRemoteObject.narrow(lookup, MICRO-  
CELL_HOME);
```

```
InitialContext ic = new InitialContext();  
Object lookup = ic.lookup("MacrocellJNDI");  
return (MacrocellLocalHome) PortableRemoteObject.narrow(lookup, MACRO-  
CELL_HOME);
```

D - CONTROLADOR FUZZY

Neste anexo, são apresentados todas as variáveis lingüísticas do controlador *fuzzy* usado para o processo de mapeamento do comportamento da ERB.

Dentre os objetivos propostos para a realização desta tese, um deles trata da necessidade de descobrir meios, para que o processo de gerência da malha telefônica tenha uma expressividade humana mais lingüística.

Conforme mencionado no texto do trabalho e em outras literaturas, a carga máxima submetida sobre uma ERB deve ficar em torno de 30 *Erlangs*. Isso devido ao número de transceptores (total de 4) implementado originalmente sobre o simulador de tráfego telefônico adotado.

Dessa forma, fica justificada a criação da variável lingüística ‘tráfego ofertado’ com seus termos lingüísticos: baixo, baixo_médio, médio, médio_alto e alto, enquadrados dentro do intervalo [0, 30] *Erlangs*. Além dos outros enquadramentos pertinentes às demais variáveis lingüísticas.

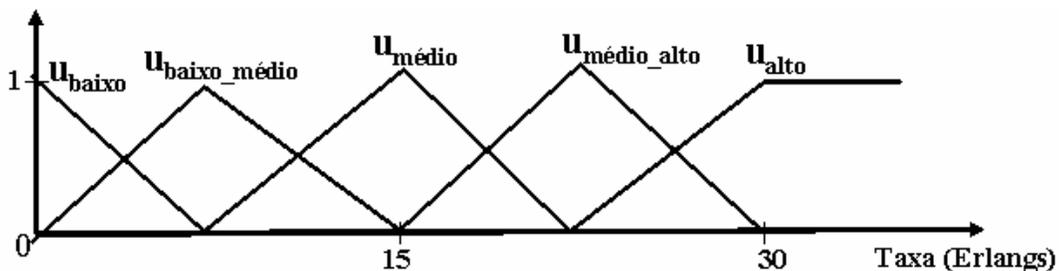


Figura D.1 Tráfego Ofertado (Erlangs)

Em relação ao número de canais GPRS permanentemente alocados, os mesmos podem variar dinamicamente dentro do intervalo [2, 8]. O limite inferior de 2 canais GPRS permanentemente alocados evidencia que sempre será possível manter um mínimo de tráfego puramente comutado.

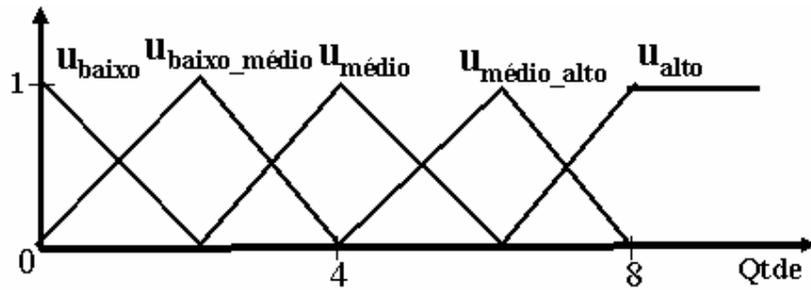


Figura D.2 Canais GPRS (quantidade de canais)

Quanto ao número de canais alocados para *handover*, os mesmos podem variar dentro do intervalo $[0, 8]$ canais. Quando nenhum canal de *handover* for alocado, nenhuma priorização desse tráfego, estará sendo feita sobre o tráfego correspondente ao estabelecimento de novas chamadas. Ao contrário, quando algum canal de *handover* for alocado, estará sendo usado um Esquema de Priorização de Canais de Handover, podendo ser do tipo: pré-reserva, ou pós-reserva de canais.

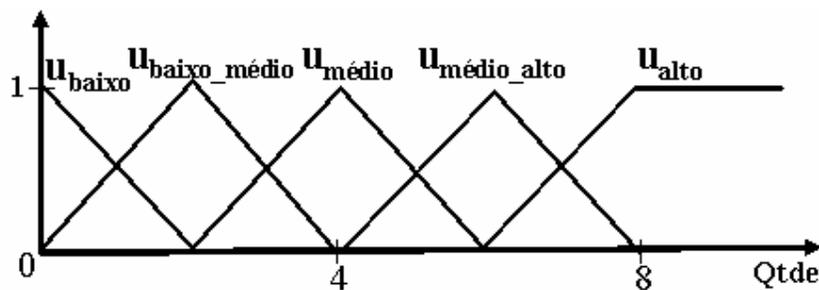


Figura D.3 Canais de Handover (quantidade de canais)

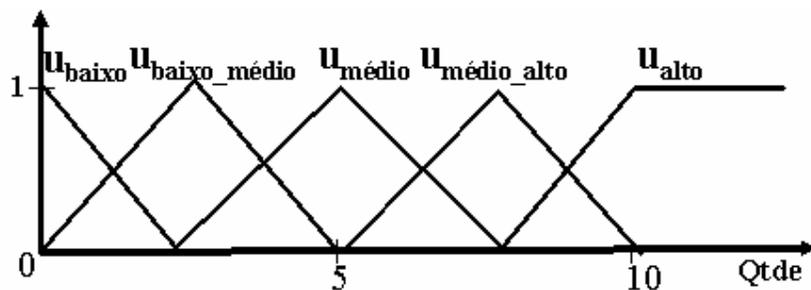


Figura D.4 Canais Sub-taxados (quantidade de canais)

Como outro recurso de otimização, os canais já em uso podem ser compartilhados com novas requisições solicitadas. Esse compartilhamento de um canal por duas chamadas em curso é chamado de sub-taxação. A variável linguística que o referencia está caracterizada

no intervalo [0, 10] canais, com seus termos lingüísticos: baixo, baixo_médio, médio, médio_alto, alto.

Uma vez que a simulação é feita sobre uma aproximação do tráfego real, aquele a ser alocado na ERB, é necessário que se evidencie um percentual de submissão, sobre o simulador, para os tráfegos de chamadas *inicializadas* e *handovers*. Esses dois tráfegos podem ser representados por meio das variáveis lingüísticas: Percentual de Voz e Percentual de *Handover*.

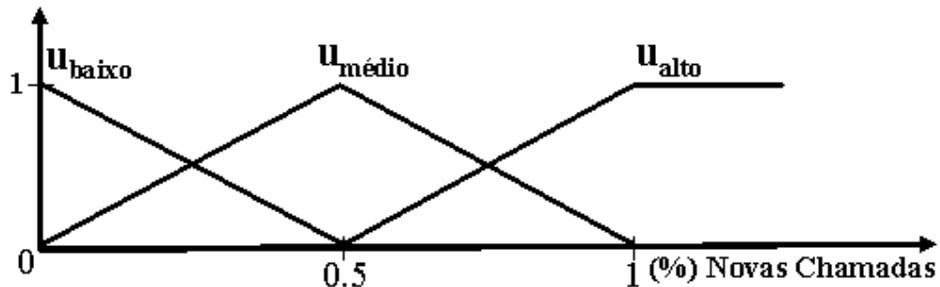


Figura D.5 Percentual de tráfego simulado de Novas Chamadas

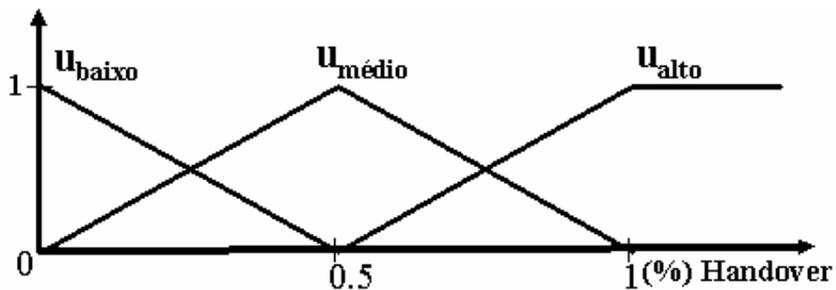


Figura D.6 Percentual de tráfego simulado de tráfego de Handover

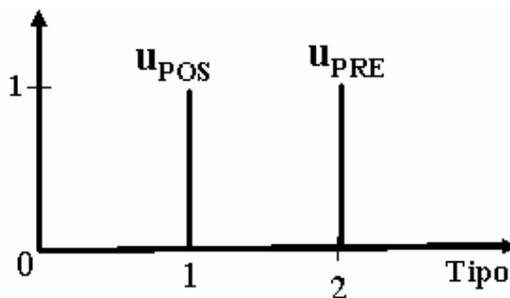


Figura D.7 Formas de Esquema de Reservas de Canais: Pré-Reserva e Pós-Reserva

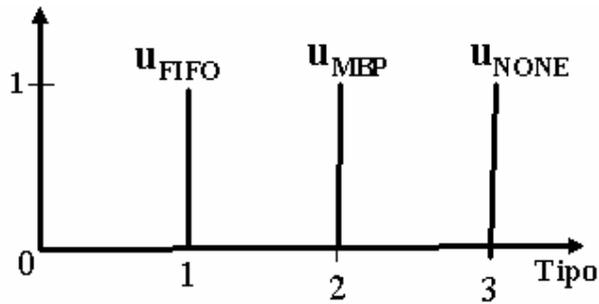


Figura D.8 Formas de Esquema de Empilhamento de Canais: FIFO (First-In First-Out), MBP (Measurable Band Priority) e NONE (ausência de empilhamento)

Por fim, outras duas variáveis lingüísticas foram criadas: a RCS e a QPS; ambas são variáveis lingüísticas pontuais, uma vez que seus termos lingüísticos são representados por pontos discretos dentro dos seus intervalos [1, 2] e [1, 2, 3] respectivamente.

D.1 - REGRAS DO CONTROLADOR FUZZY

```

Se trafego handover é baixo
  Se offeredTraffic é baixo
    gprs é alto
    handover é baixo
    pre RCS
    nonQPS
    SRS é baixo
  end
  Se offeredTraffic é baixo e médio
    gprs é alto
    handover é baixo
    pre RCS
    nonQPS
    SRS é baixo
  end
  Se offeredTraffic médio
    gprs é alto
    handover é médio
    pre RCS
    QPS é FIFO
    SRS é baixo e médio
  end
  Se offeredTraffic médio e alto
    gprs é médio
    handover é médio e alto
    pos RCS
    QPS é FIFO
    SRS é médio
  end

```

```

    Se offeredTraffic alto
        gprs é médio
        handover é médio e alto
        pos RCS
        QPS é MBP
        SRS é médio
    end
end trafego percentual

Se trafego handover é médio
    Se offeredTraffic é baixo
        gprs é alto
        handover é baixo
        pre RCS
        nonQPS
        SRS é baixo
    end
    Se offeredTraffic é baixo e médio
        gprs é alto
        handover é baixo e médio
        pre RCS
        nonQPS
        SRS é baixo
    end
    Se offeredTraffic médio
        gprs é médio
        handover é médio e alto
        pre RCS
        QPS é FIFO
        SRS é baixo e médio
    end
    Se offeredTraffic médio e alto
        gprs é baixo e médio
        handover é médio e alto
        pos RCS
        QPS é FIFO
        SRS é médio
    end
    Se offeredTraffic alto
        gprs é baixo
        handover é alto
        pos RCS
        QPS é MBP
        SRS é médio
    end
end trafego percentual

Se trafego handover é alto
    Se offeredTraffic é baixo
        gprs é alto

```

```

        handover é baixo
        pre RCS
        nonQPS
        SRS é baixo
    end
    Se offeredTraffic é baixo e médio
        gprs é alto
        handover é baixo e médio
        pre RCS
        nonQPS
        SRS é baixo e médio
    end
    Se offeredTraffic médio
        gprs é médio
        handover é médio e alto
        pre RCS
        QPS é FIFO
        SRS é médio
    end
    Se offeredTraffic médio e alto
        gprs é baixo e médio
        handover é médio e alto
        pos RCS
        QPS é FIFO
        SRS é médio
    end
    Se offeredTraffic alto
        gprs é baixo
        handover é alto
        pos RCS
        QPS é MBP
        SRS é médio
    end
end trafego percentual

```

D.2 - RESULTADOS DE PONTOS DISCRETOS DA OTIMIZAÇÃO *FUZZY* PARA O CENÁRIO DE MICRO-CÉLULA

Os valores abaixo representam uma das execuções do controlador *fuzzy*, para o intervalo [1, 30] Erlangs de tráfego ofertado e situação onde o tráfego simulado de *handover* e de novas chamadas sejam aproximadamente iguais, ou seja, 50% e 50%, respectivamente.

Tráfego Ofertado	Canais GPRS	Canais <i>Handover</i>	Esquema RCS	Esquema QPS	Canais Sub-Taxados
1	7	1	Pré-RCS	não	0

2	7	1	Pré-RCS	não	0
3	7	1	Pré-RCS	não	0
4	7	1	Pré-RCS	não	0
5	7	1	Pré-RCS	Não	0
6	7	1	Pré-RCS	Não	0
7	7	1	Pré-RCS	Não	0
8	7	2	Pré-RCS	Não	1
9	6	3	Pré-RCS	MBP	1
10	5	3	Pré-RCS	MBP	1
11	5	3	Pré-RCS	MBP	2
12	5	4	Pré-RCS	MBP	2
13	5	4	Pré-RCS	MBP	2
14	4	5	Pré-RCS	FIFO	2
15	4	5	Pré-RCS	FIFO	2
16	4	5	Pós-RCS	FIFO	2
17	4	5	Pós-RCS	FIFO	3
18	4	5	Pós-RCS	FIFO	3
19	4	5	Pós-RCS	FIFO	3
20	4	5	Pós-RCS	FIFO	4
21	4	5	Pós-RCS	FIFO	4
22	3	5	Pós-RCS	FIFO	4
23	2	6	Pós-RCS	FIFO	5
24	2	6	Pós-RCS	FIFO	5
25	2	6	Pós-RCS	FIFO	5
26	2	6	Pós-RCS	FIFO	5
27	2	6	Pré-RCS	MBP	5
28	2	6	Pré-RCS	MBP	5
29	2	6	Pré-RCS	MBP	5
30	2	6	Pré-RCS	MBP	5

D.3 - RESULTADOS DE PONTOS DISCRETOS DA OTIMIZAÇÃO FUZZY PARA O CENÁRIO DE MACRO-CÉLULA

Os valores abaixo representam uma das execuções do controlador *fuzzy*, para o intervalo [1, 30] Erlangs de tráfego ofertado e situação onde o tráfego simulado de *handover* e de novas chamadas sejam aproximadamente iguais, ou seja, 50% e 50%, respectivamente.

Tráfego Ofertado	Canais GPRS	Canais Handover	Esquema RCS	Esquema QPS	Canais Sub-Taxados
1	7	1	Pré-RCS	Não	0
2	7	1	Pré-RCS	Não	0
3	7	1	Pré-RCS	Não	0
4	7	1	Pré-RCS	Não	0

5	7	1	Pré-RCS	Não	0
6	7	1	Pré-RCS	Não	0
7	7	1	Pré-RCS	Não	0
8	7	1	Pré-RCS	Não	1
9	7	2	Pré-RCS	MBP	1
10	7	3	Pré-RCS	MBP	1
11	7	3	Pré-RCS	MBP	2
12	6	4	Pré-RCS	MBP	2
13	5	4	Pré-RCS	MBP	2
14	5	4	Pré-RCS	FIFO	2
15	4	5	Pré-RCS	FIFO	2
16	4	5	Pós-RCS	FIFO	2
17	4	5	Pós-RCS	FIFO	3
18	4	5	Pós-RCS	FIFO	3
19	4	5	Pós-RCS	FIFO	3
20	4	5	Pós-RCS	FIFO	4
21	4	5	Pós-RCS	FIFO	4
22	3	6	Pós-RCS	FIFO	4
23	2	6	Pós-RCS	FIFO	4
24	2	6	Pós-RCS	FIFO	4
25	2	6	Pós-RCS	FIFO	5
26	2	6	Pós-RCS	FIFO	5
27	2	6	Pré-RCS	MBP	5
28	2	6	Pré-RCS	MBP	5
29	2	6	Pré-RCS	MBP	5
30	2	6	Pré-RCS	MBP	5

E - REDE NEURAL ARTIFICIAL

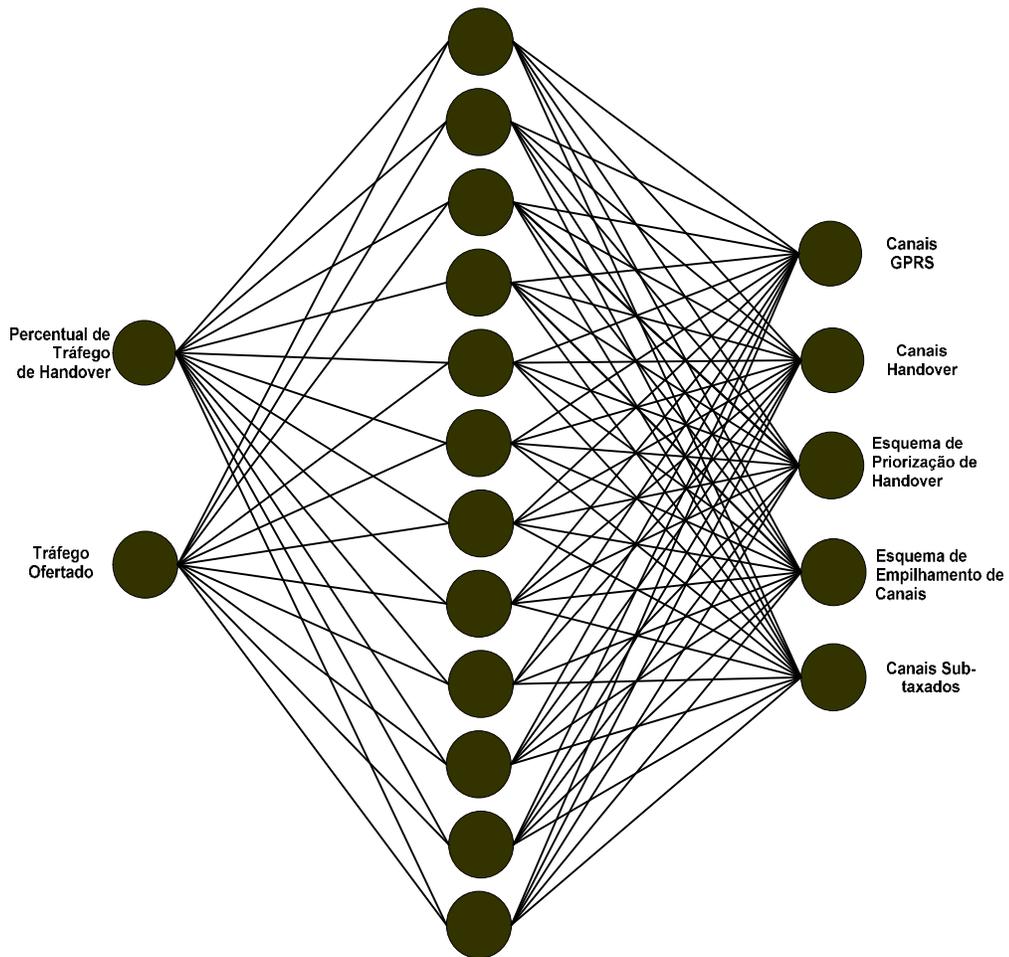


Figura E.1 Topologia da Rede Neural Artificial

E.1 - TREINAMENTO DA REDE NEURAL

Como o treinamento da rede neural pode ocupar várias horas de CPU, é aconselhável realizar essa tarefa como um arquivo batch. Para isso, a Ferramenta SNNS oferece o programa 'batchman.exe'. Esse programa é uma interface para o *kernel* do SNNS, permitindo uma execução fácil em *background*.

O treinamento da Rede Neural é feito por meio da construção de um *script* de comandos específicos para a execução do programa 'batchman.exe'. Sua linguagem suporta todas as funções necessárias para o treinamento e teste das redes neurais.

Essa linguagem *batch* foi modelada depois das linguagens: AWK, Pascal, Modula2 e C; portanto, é um avanço em relação a essas linguagens. A linguagem permite que o usuário obtenha o resultado desejado sem usar estruturas sintáticas complexas. A linguagem aqui utilizada é interpretada; portanto, se um erro ocorrer no programa, o usuário terá a mensagem de erro apresentada em tempo de execução, juntamente com o número da linha correspondente.

O trecho de código abaixo mostra o *script* utilizado nesse trabalho de tese.

```
# Carrega a estrutura física da Rede Neural
loadNet("RNDaTese.net");
# Carrega os valores de entrada para o treinamento da Rede Neural
loadPattern("treinamento.pat");
loadPattern("validacao.pat");
# Inicializa os pesos da rede neural com valores no intervalo [-1.0, 1.0]
setInitFunc("Randomize_Weights", 1.0, -1.0)
# A função de aprendizado neural é a Backpropagation padrão.
setLearnFunc("Std_Backpropagation", 0.1);
# Seleciona os diferentes algoritmos usados na Rede Neural
setPruningFunc("OptimalBrainDamage", "Std_Backpropagation");
# Seleciona a próxima estrutura de entrada, randomicamente.
setShuffle(TRUE)
# Inicializa a Rede Neural.
initNet()
# Habilita para que o próximo padrão a ser usado seja o abaixo.
setPattern("treinamento.pat")
# Operação de Atribuição
valid_error = SSE
# Treina a Rede Neural.
while SSE > 0.1 and CYCLES < 10000000 and SIGNAL == 0 do
  if CYCLES mod 100 == 0 then
    print("cycles = ", CYCLES, " SSE = ", SSE, " valid_error = ", valid_error)
    setPattern("validacao.pat")
    testNet()
    setPattern("treinamento.pat")
    valid_error := SSE
  endif
  trainNet()
  if valid_error < 0.1 then
    break
  endif
endwhile
# Salva a Rede Neural correntemente localizada no kernel do simulador.
saveNet("test"+CYCLES+"cycles.net")
# Salva o resultado do treinamento neural.
saveResult("result.res")
```

```

# Tratamento de erro de execução
if SIGNAL != 0 then
    print("Stopped due to signal reception: signal "+SIGNAL)
endif
# Número de ciclos neurais.
print("Cycles trained: ", CYCLES)
# Soma das diferenças quadradas dos neurônio de saída.
print("Training stopped at error: ", SSE)
print("Testing error: ", valid_error)

```

E.2 - ROTEIRO DE EXECUÇÃO DO CENÁRIO DE MICRO-CÉLULA

Prompt_de_execução> batchman -f <arquivo de script.bat>

```

# Net RNDaTese.net loaded
# Patternset treinamento.pat loaded; 1 patternset(s) in memory
# Patternset validacao.pat loaded; 2 patternset(s) in memory
# Init function is now Randomize_Weights
# Parameters are: 1 -1
# Learning function is now Std_Backpropagation
# Parameters are: 0.1
# Pruning function is now OptimalBrainDamage
# Subordinate learning function is now Std_Backpropagation

# Pattern shuffling enabled
# Net initialized
cycles = 0 SSE = 3.40282e+038 valid_error = 3.40282e+038
cycles = 100 SSE = 70.9979 valid_error = 210.224
cycles = 200 SSE = 70.9978 valid_error = 210.224
cycles = 300 SSE = 70.9978 valid_error = 210.224
cycles = 400 SSE = 70.9977 valid_error = 210.223
cycles = 500 SSE = 70.9977 valid_error = 210.223
cycles = 600 SSE = 70.9976 valid_error = 210.223
cycles = 700 SSE = 70.9975 valid_error = 210.223
cycles = 800 SSE = 70.9975 valid_error = 210.223
cycles = 900 SSE = 70.9974 valid_error = 210.222
cycles = 1000 SSE = 70.9973 valid_error = 210.222
cycles = 1100 SSE = 70.9972 valid_error = 210.222
cycles = 1200 SSE = 70.9971 valid_error = 210.222
cycles = 1300 SSE = 70.997 valid_error = 210.221
cycles = 1400 SSE = 70.9969 valid_error = 210.221
cycles = 1500 SSE = 70.9968 valid_error = 210.221
cycles = 1600 SSE = 70.9967 valid_error = 210.221
cycles = 1700 SSE = 70.9966 valid_error = 210.22
cycles = 1800 SSE = 70.9964 valid_error = 210.22
cycles = 1900 SSE = 70.9963 valid_error = 210.219
cycles = 2000 SSE = 70.9961 valid_error = 210.219
cycles = 2100 SSE = 70.9959 valid_error = 210.218

```

cycles = 2200 SSE = 70.9957 valid_error = 210.218
cycles = 2300 SSE = 70.9955 valid_error = 210.217
cycles = 2400 SSE = 70.9952 valid_error = 210.216
cycles = 2500 SSE = 70.9949 valid_error = 210.216
cycles = 2600 SSE = 70.9946 valid_error = 210.215
cycles = 2700 SSE = 70.9942 valid_error = 210.214
cycles = 2800 SSE = 70.9938 valid_error = 210.213
cycles = 2900 SSE = 70.9932 valid_error = 210.211
cycles = 3000 SSE = 70.9926 valid_error = 210.21
cycles = 3100 SSE = 70.9918 valid_error = 210.208
cycles = 3200 SSE = 70.9909 valid_error = 210.206
cycles = 3300 SSE = 70.9896 valid_error = 210.203
cycles = 3400 SSE = 70.9878 valid_error = 210.199
cycles = 3500 SSE = 70.9852 valid_error = 210.194
cycles = 3600 SSE = 70.9807 valid_error = 210.186
cycles = 3700 SSE = 70.9697 valid_error = 210.172
cycles = 3800 SSE = 70.7462 valid_error = 210.139
cycles = 3900 SSE = 51.7798 valid_error = 209.431
cycles = 4000 SSE = 51.7608 valid_error = 153.209
cycles = 4100 SSE = 34.7387 valid_error = 153.153
cycles = 4200 SSE = 34.6905 valid_error = 102.714
cycles = 4300 SSE = 22.3904 valid_error = 102.568
cycles = 4400 SSE = 22.3243 valid_error = 66.2191
cycles = 4500 SSE = 11.851 valid_error = 65.9933
cycles = 4600 SSE = 11.4676 valid_error = 33.7603
: : : : : : : : : :
: : : : : : : : : :

E.3 - RESULTADOS DE PONTOS DISCRETOS DA OTIMIZAÇÃO NEURAL PARA O CENÁRIO DE MICRO-CÉLULA

Os valores abaixo representam uma das execuções da Rede Neural Perceptron Multicamadas com algoritmo de treinamento *feed-forward backpropagation*, para o intervalo [1, 30] Erlangs de tráfego ofertado e situação onde o tráfego simulado de *handover* e de novas chamadas sejam aproximadamente iguais, ou seja, 50% e 50%, respectivamente.

Tráfego Ofertado	Canais GPRS	Canais Handover	Esquema RCS	Esquema QPS	Canais Sub-Taxados
1	7	1	Pós-RCS	FIFO	0
2	7	1	Pós-RCS	FIFO	0
3	7	1	Pós-RCS	FIFO	0
4	7	1	Pós-RCS	FIFO	0
5	7	1	Pós-RCS	FIFO	0
6	7	1	Pós-RCS	FIFO	0

7	7	1	Pós-RCS	FIFO	0
8	7	2	Pós-RCS	FIFO	1
9	5	3	Pós-RCS	FIFO	1
10	5	3	Pós-RCS	FIFO	1
11	5	2	Pós-RCS	FIFO	2
12	2	4	Pós-RCS	FIFO	2
13	4	1	Pós-RCS	FIFO	2
14	1	2	Pós-RCS	FIFO	2
15	3	0	Pós-RCS	FIFO	2
16	4	5	Pós-RCS	FIFO	2
17	4	5	Pós-RCS	FIFO	3
18	4	5	Pós-RCS	FIFO	3
19	3	5	Pós-RCS	FIFO	3
20	3	4	Pós-RCS	FIFO	4
21	3	3	Pós-RCS	FIFO	4
22	1	4	Pós-RCS	FIFO	4
23	0	3	Pós-RCS	FIFO	5
24	1	1	Pós-RCS	FIFO	5
25	0	5	Pós-RCS	FIFO	5
26	1	3	Pós-RCS	FIFO	5
27	0	5	Pós-RCS	FIFO	5
28	0	5	Pós-RCS	FIFO	5
29	0	4	Pós-RCS	FIFO	5
30	0	5	Pós-RCS	FIFO	5

E.4 - ROTEIRO DE EXECUÇÃO DO CENÁRIO DE MACRO-CÉLULA

Prompt_de_execução> batchman -f <arquivo de script.bat>

```
# Net RNDaTese.net loaded
# Patternset treinamento.pat loaded; 1 patternset(s) in memory
# Patternset validacao.pat loaded; 2 patternset(s) in memory
# Init function is now Randomize_Weights
# Parameters are: 1 -1
# Learning function is now Std_Backpropagation
# Parameters are: 0.1
# Pruning function is now OptimalBrainDamage
# Subordinate learning function is now Std_Backpropagation

# Pattern shuffling enabled
cycles = 0 SSE = 3.40282e+038 valid_error = 3.40282e+038
cycles = 100 SSE = 57.0303 valid_error = 168.913
cycles = 200 SSE = 57.0302 valid_error = 168.913
cycles = 300 SSE = 57.0302 valid_error = 168.913
cycles = 400 SSE = 57.0301 valid_error = 168.913
cycles = 500 SSE = 57.0301 valid_error = 168.912
cycles = 600 SSE = 57.03 valid_error = 168.912
```

cycles = 700 SSE = 57.03 valid_error = 168.912
cycles = 800 SSE = 57.0299 valid_error = 168.912
cycles = 900 SSE = 57.0299 valid_error = 168.912
cycles = 1000 SSE = 57.0298 valid_error = 168.912
cycles = 1100 SSE = 57.0297 valid_error = 168.911
cycles = 1200 SSE = 57.0297 valid_error = 168.911
cycles = 1300 SSE = 57.0296 valid_error = 168.911
cycles = 1400 SSE = 57.0295 valid_error = 168.911
cycles = 1500 SSE = 57.0294 valid_error = 168.911
cycles = 1600 SSE = 57.0294 valid_error = 168.91
cycles = 1700 SSE = 57.0293 valid_error = 168.91
cycles = 1800 SSE = 57.0291 valid_error = 168.91
cycles = 1900 SSE = 57.029 valid_error = 168.91
cycles = 2000 SSE = 57.0289 valid_error = 168.909
cycles = 2100 SSE = 57.0288 valid_error = 168.909
cycles = 2200 SSE = 57.0286 valid_error = 168.909
cycles = 2300 SSE = 57.0285 valid_error = 168.908
cycles = 2400 SSE = 57.0283 valid_error = 168.908
cycles = 2500 SSE = 57.0281 valid_error = 168.907
cycles = 2600 SSE = 57.0279 valid_error = 168.906
cycles = 2700 SSE = 57.0276 valid_error = 168.906
cycles = 2800 SSE = 57.0274 valid_error = 168.905
cycles = 2900 SSE = 57.027 valid_error = 168.904
cycles = 3000 SSE = 57.0266 valid_error = 168.903
cycles = 3100 SSE = 57.0262 valid_error = 168.902
cycles = 3200 SSE = 57.0256 valid_error = 168.901
cycles = 3300 SSE = 57.0249 valid_error = 168.899
cycles = 3400 SSE = 57.024 valid_error = 168.897
cycles = 3500 SSE = 57.0228 valid_error = 168.894
cycles = 3600 SSE = 57.0211 valid_error = 168.891
cycles = 3700 SSE = 57.0183 valid_error = 168.886
cycles = 3800 SSE = 57.0128 valid_error = 168.877
cycles = 3900 SSE = 56.9942 valid_error = 168.861
cycles = 4000 SSE = 41.411 valid_error = 168.806
cycles = 4100 SSE = 41.3734 valid_error = 123.558
cycles = 4200 SSE = 41.2414 valid_error = 123.463
cycles = 4300 SSE = 26.8478 valid_error = 123.064
cycles = 4400 SSE = 26.81 valid_error = 79.8239
cycles = 4500 SSE = 26.7715 valid_error = 79.7453
cycles = 4600 SSE = 26.7249 valid_error = 79.6199
: : : : : : : : : :
: : : : : : : : : :

E.5 - RESULTADOS DE PONTOS DISCRETOS DA OTIMIZAÇÃO NEURAL PARA O CENÁRIO DE MACRO-CÉLULA

Os valores abaixo representam uma das execuções da Rede Neural Perceptron Multicamadas com algoritmo de treinamento *feed-forward backpropagation*, para o interva-

lo [1, 30] Erlangs de tráfego ofertado e situação onde o tráfego simulado de *handover* e de novas chamadas sejam aproximadamente iguais, ou seja, 50% e 50%, respectivamente.

Tráfego Ofertado	Canais GPRS	Canais <i>Handover</i>	Esquema RCS	Esquema QPS	Canais Sub-Taxados
1	7	1	Pós-RCS	FIFO	0
2	7	1	Pós-RCS	FIFO	0
3	7	1	Pós-RCS	FIFO	0
4	7	1	Pós-RCS	FIFO	0
5	7	1	Pós-RCS	FIFO	0
6	7	1	Pós-RCS	FIFO	0
7	7	1	Pós-RCS	FIFO	0
8	7	2	Pós-RCS	FIFO	1
9	5	2	Pós-RCS	FIFO	1
10	5	3	Pós-RCS	FIFO	1
11	5	2	Pós-RCS	FIFO	2
12	2	4	Pós-RCS	FIFO	2
13	4	1	Pós-RCS	FIFO	2
14	1	1	Pós-RCS	FIFO	2
15	3	1	Pós-RCS	FIFO	2
16	4	4	Pós-RCS	FIFO	2
17	4	4	Pós-RCS	FIFO	3
18	4	4	Pós-RCS	FIFO	3
19	3	4	Pós-RCS	FIFO	3
20	3	4	Pós-RCS	FIFO	4
21	3	3	Pós-RCS	FIFO	4
22	1	3	Pós-RCS	FIFO	4
23	0	3	Pós-RCS	FIFO	5
24	1	1	Pós-RCS	FIFO	5
25	0	4	Pós-RCS	FIFO	5
26	1	3	Pós-RCS	FIFO	5
27	0	4	Pós-RCS	FIFO	5
28	0	4	Pós-RCS	FIFO	5
29	0	4	Pós-RCS	FIFO	5
30	0	4	Pós-RCS	FIFO	5

F - MIB INTELIGENTE

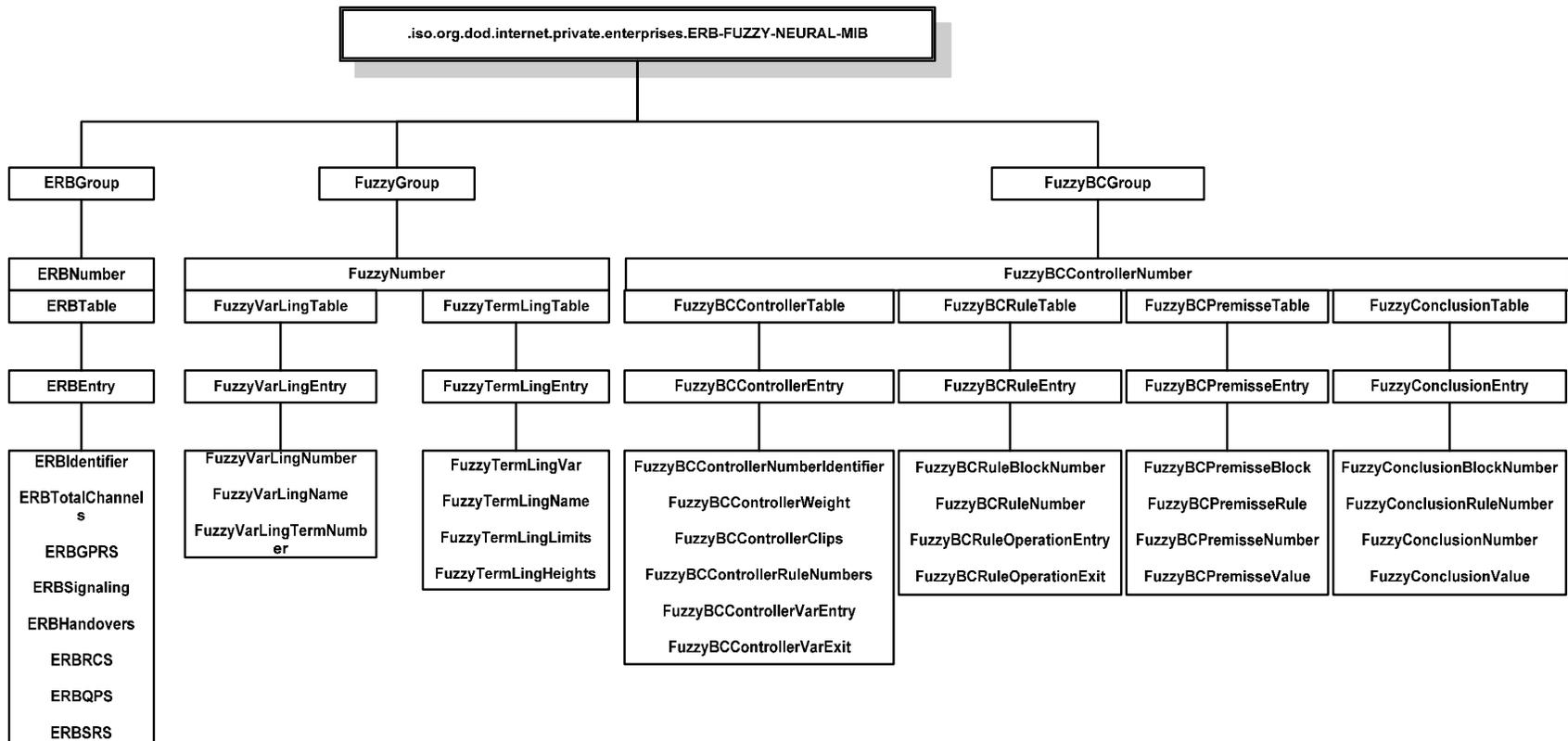


Figura F.1 – Estrutura da .iso.org.dod.internet.private.enterprises.ERB-FUZZY-NEURAL-MIB (parte 1)

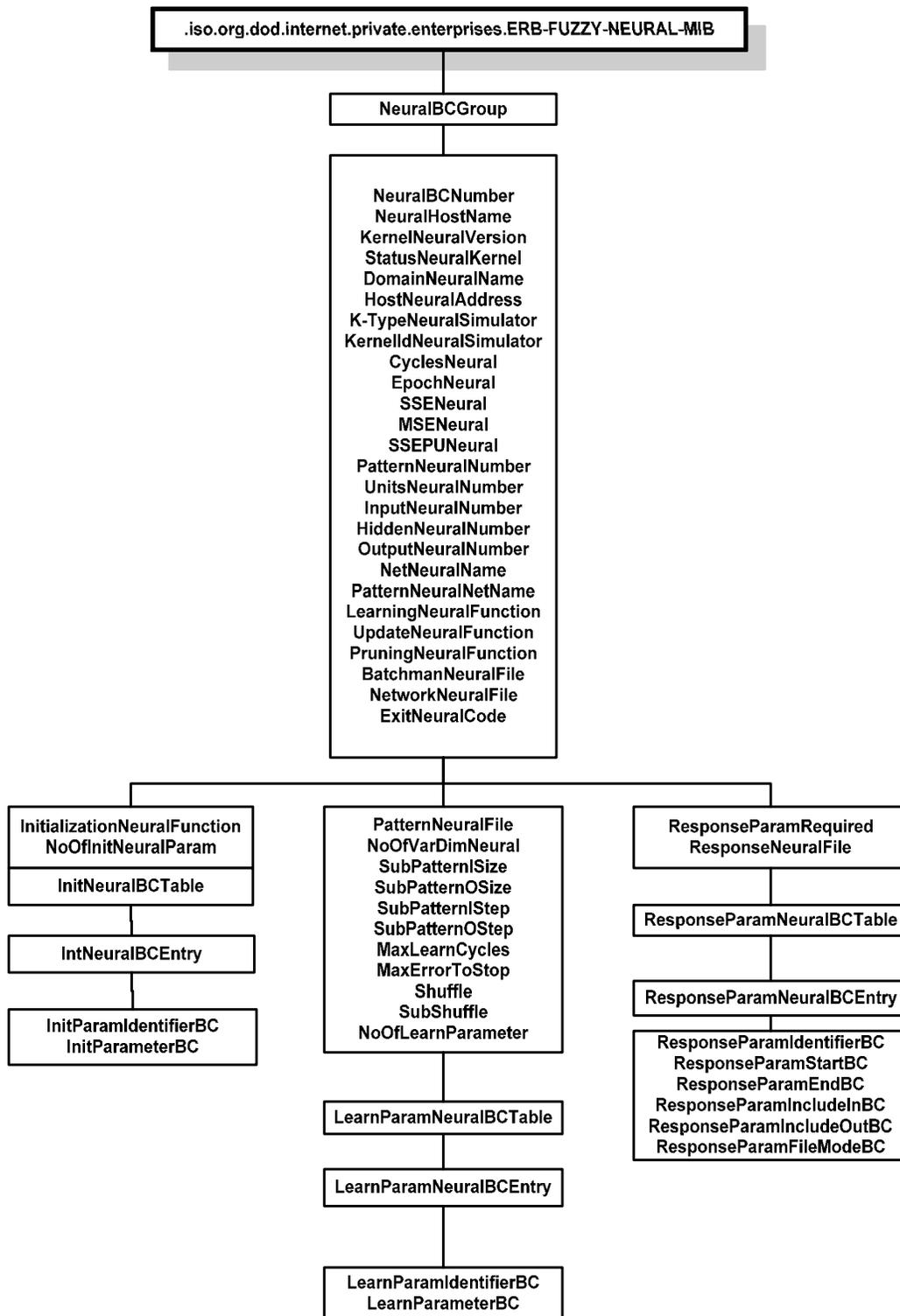


Figura F.2 – Estrutura da .iso.org.dod.internet.private.enterprises.ERB-FUZZY-NEURAL-MIB (parte 2)

F.1 MIB NEURAL-FUZZY

Este anexo apresenta a descrição da MIB NEURAL-FUZZY, que está de acordo com a *Structure of Management Information – SMIV2*. Esta MIB contém as informações necessárias para a inferência difusa presente no Módulo de Inteligência.

```
ERB-FUZZY-NEURAL-MIB DEFINITIONS ::= BEGIN

IMPORTS
    NetworkAddress,
    IPAddress,
    Counter,
    Gauge,
    TimeTicks
    FROM RFC1155-SMI
    OBJECT-TYPE
    FROM RFC-1212;

-- This MIB module uses the extended OBJECT-TYPE macro as defined in [14];

ERB-FUZZY-NEURAL-MIB      OBJECT IDENTIFIER ::= { enterprises 1 }

-- textual conventions

DisplayString ::= OCTET STRING
-- This data type is used to model textual information taken from the NVT
-- ASCII character set.  By convention, objects with this syntax are de-
-- clared
-- as having (SIZE (0..255)).

PhysAddress ::= OCTET STRING
-- This data type is used to model media addresses.  For many types of me-
-- dia,
-- this will be in a binary representation.  For example, an ethernet ad-
-- dress
-- would be represented as a string of 6 octets.

-- groups in ERB-FUZZY-NEURAL-MIB

ERB      OBJECT IDENTIFIER ::= { ERB-FUZZY-NEURAL-MIB 1 }

Fuzzy    OBJECT IDENTIFIER ::= { ERB-FUZZY-NEURAL-MIB 2 }

FuzzyBC  OBJECT IDENTIFIER ::= { ERB-FUZZY-NEURAL-MIB 3 }

NeuralBC OBJECT IDENTIFIER ::= { ERB-FUZZY-NEURAL-MIB 4 }

-- the ERB group

-- Implementation of the ERB group is mandatory for all systems.  If an a-
-- gent
-- is not configured to have a value for any of these variables, a string
-- of
-- length 0 is returned.

ERBNumber OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-write
```

```

STATUS mandatory
DEFVAL 0
DESCRIPTION
    "Um numero inteiro que descreve quantas ERBs estão cataloga-
das
    na interface."
::= { ERB 1 }

-- the ERB table

-- A tabela ERB contém informações de gerenciamento catalogadas das ERB da
-- malha telefônica.
-- Tais informações caracterizam a configuração da rede sobre o plano de
-- gerência.

ERBTable OBJECT-TYPE
SYNTAX SEQUENCE OF ERBEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "Uma lista das ERBs com suas respectivas informações de
    gerência."
::= { ERB 2 }

ERBEntry OBJECT-TYPE
SYNTAX FhostEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "Uma entrada de ERB contendo informações de gerenciamento."
INDEX { ERBIdentifier }
::= { ERBTable 1 }

ERBEntry ::=
SEQUENCE {
    ERBIdentifier PhysAddress,
    ERBTotalChannels INTEGER,
    ERBGPRS INTEGER,
    ERBSignalling INTEGER,
    ERBHandovers INTEGER,
    ERBRCS INTEGER,
    ERBQPS INTEGER,
    ERBSRS INTEGER,
}

ERBIdentifier OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "Endereco Físico de uma ERB gerenciada."
::= { ERBEntry 1 }

ERBTotalChannels OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DEFVAL 0
DESCRIPTION
    "Este objeto informa a quantidade TOTAL de canais

```

```

                existente na interface física da ERB."
 ::= { ERBEntry 2 }

ERBGPRS      OBJECT-TYPE
              SYNTAX INTEGER
              ACCESS read-only
              STATUS mandatory
              DEFVAL 0
              DESCRIPTION
                "Este objeto informa a quantidade total de canais exclu-
                 sivamente para tráfego GPRS, existente na interface fí-
                 sica da ERB."
 ::= { ERBEntry 3 }

ERBSignalling OBJECT-TYPE
              SYNTAX INTEGER
              ACCESS read-only
              STATUS mandatory
              DEFVAL 0
              DESCRIPTION
                "Este objeto informa a quantidade total de canais exclu-
                 sivamente para tráfego de SINALIZAÇÃO, existente na in-
                 terface física da ERB."
 ::= { ERBEntry 4 }

ERBHandovers OBJECT-TYPE
              SYNTAX INTEGER
              ACCESS read-only
              STATUS mandatory
              DEFVAL 0
              DESCRIPTION
                "Este objeto informa a quantidade total de canais exclu-
                 sivamente para tráfego de HANDOVER existente na interfa-
                 ce física da ERB."
 ::= { ERBEntry 5 }

ERBRCS      OBJECT-TYPE
              SYNTAX INTEGER
              ACCESS read-only
              STATUS mandatory
              DEFVAL 0
              DESCRIPTION
                "Este objeto informa o esquema adotado para alocação RCS
                 (Reserved Channel Scheme), a se realizar na interface
                 física da ERB.
                 Os tipos de alocações RCS são:
                 - Pré-RCS, e
                 - Pós-RCS."
 ::= { ERBEntry 6 }

ERBQPS      OBJECT-TYPE
              SYNTAX INTEGER
              ACCESS read-only
              STATUS mandatory
              DEFVAL 0
              DESCRIPTION
                "Este objeto informa a modalidade adotada para alocação
                 QPS (Queue Priority Scheme, a se realizar na interface
                 física da ERB.
                 As modalidades de QPS são:

```

```

        - none,
        - FIFO (First In First Out), e
        - MBP (Measurable Band Priority)."
 ::= { ERBEntry 7 }

ERBSRS      OBJECT-TYPE
            SYNTAX INTEGER
            ACCESS read-only
            STATUS mandatory
            DEFVAL 0
            DESCRIPTION
                "Este objeto informa a quantidade total de canais SUB-
                TAXADOS na interface física da ERB."
 ::= { ERBEntry 8 }

-- the Fuzzy group

-- A implementação do grupo Fuzzy trata de armazenar informações sobre to-
das
-- as variáveis fuzzy catalogadas.
-- Este grupo permite que se construa, em tempo de execução, a estrutura
-- adotada para a aplicação de construções fuzzy ao modelo de simulação.

FuzzyNumber OBJECT-TYPE
            SYNTAX INTEGER
            ACCESS read-write
            STATUS mandatory
            DEFVAL 0
            DESCRIPTION
                "Informa o numero de variáveis lingüísticas catalogadas."
 ::= { Fuzzy 1 }

-- the FuzzyVarLing table

-- A tabela FuzzyVarLing contém informações de cada variável lingüística em
particular.

FuzzyVarLingTable OBJECT-TYPE
            SYNTAX SEQUENCE OF FuzzyVarLingEntry
            ACCESS not-accessible
            STATUS mandatory
            DESCRIPTION
                "Uma lista das variáveis lingüísticas possíveis de serem
                monitoradas."
 ::= { Fuzzy 2 }

FuzzyVarLingEntry OBJECT-TYPE
            SYNTAX FuzzyVarLingEntry
            ACCESS not-accessible
            STATUS mandatory
            DESCRIPTION
                "Uma entrada contendo informações de uma variável
                lingüística particular."
            INDEX { FuzzyVarLingNumber }
 ::= { FuzzyVarLingTable 1 }

FuzzyVarLingEntry ::=
    SEQUENCE {
        FuzzyVarLingNumber    INTEGER,
        FuzzyVarLingName      DisplayString,

```

```

        FuzzyVarLingTermNumber INTEGER,
    }

FVarLingNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Ordem da Variável Lingüística Fuzzy catalogada."
    ::= { FuzzyVarLingEntry 1 }

FuzzyVarLingName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Nome da Variável Lingüística Fuzzy envolvida no mó-
        dulo de inferência da simulação."
    ::= { FuzzyVarLingEntry 2 }

FuzzyVarLingTermNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DEFVAL 5
    DESCRIPTION
        "Informa o numero do termos lingüísticos as-
        sociados a uma variável lingüística particu-
        lar."
    ::= { FuzzyVarLingEntry 3 }

-- A implementação da tabela FuzzyTermLing trata de armazenar informações
-- sobre todos os termos Lingüísticos catalogados para todas as variáveis
-- lingüísticas.

FuzzyTermLingTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuzzyTermLingEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma lista de todos os termos lingüísticos para as vá-
        rias Variáveis Lingüísticas catalogadas."
    ::= { Fuzzy 3 }

FuzzyTermLingEntry OBJECT-TYPE
    SYNTAX FuzzyTermLingEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma tabela contendo informações dos termos lingüísti-
        cos referentes as variáveis lingüísticas fuzzy.
        Chave da Tabela: (FuzzyTermLingVar, FuzzyTermLingNa-
        me)"
    INDEX { FuzzyTermLingVar, FuzzyTermLingName }
    ::= { FuzzyTermLingTable 1 }

FuzzyTermLingEntry ::=
    SEQUENCE {
        FuzzyTermLingVar    INTEGER,

```

```

FuzzyTermLingName   DisplayString,
FuzzyTermLingLimit  DisplayString,
FuzzyTermLingHeights DisplayString,
}

```

```

FuzzyTermLingVar OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Referencia uma das variáveis lingüísticas presente na
        tabela FuzzyVarLing."
    ::= { FuzzyTermLingEntry 1 }

```

```

FuzzyTermLingName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Nome do termo lingüístico."
    ::= { FuzzyTermLingEntry 2 }

```

```

FuzzyTermLingLimits OBJECT-TYPE
    SYNTAX DisplayString(SIZE(0..255))
    ACCESS read-write
    STATUS mandatory
    DEFVAL 0
    DESCRIPTION
        "Possui todos os limites do termo lingüístico associado.
        A sintaxe para o preenchimento deste objeto é":
        <termo1>[;<termo2>][;<termo3>]..."
    ::= { FTermLingEntry 3 }

```

```

FuzzyTermLingHeights OBJECT-TYPE
    SYNTAX DisplayString(SIZE(0..255))
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valores reais armazenados em um objeto do tipo Dis-
        play String. Este objeto é referente a altura dos
        termos fuzzy encontrados no objeto FuzzyTermLingLi-
        mits. A sintaxe de preenchimento deste objeto é simi-
        lar a do objeto FuzzyTermLingLimits: <hei-
        ght1>[;<height2>]
        [;<height3>]..."
    ::= { FuzzyTermLingEntry 4 }

```

-- the FuzzyBC group

-- Implementation of the FuzzyBC group is mandatory for all systems.
-- If an agent is not configured to have a value for any of these vari-
ables,
-- a string of length 0 is returned.

```

FuzzyBCControllerNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION

```

```

        "Numero de Blocos presentes na base de conhecimento fuzzy."
 ::= { FuzzyBC 1 }

-- the FuzzyBCControllerTable table
-- A tabela FuzzyBCController contém as Regras de Conhecimento Fuzzy sobre
a
-- construção do controlador Fuzzy.
-- Tais informações são utilizadas no Módulo de Inteligência do simulador
de
-- tráfego telefônico.

FuzzyBCControllerTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuzzyBCControllerEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma lista dos blocos que são utilizados na infe-
        rência fuzzy."
 ::= { FuzzyBC 2 }

FuzzyBCControllerEntry OBJECT-TYPE
    SYNTAX FuzzyBCControllerEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma entrada de um bloco que é utilizado na infe-
        rência fuzzy."
    INDEX { FuzzyBCControllerNumberIdentifier }
 ::= { FuzzyBCControllerTable 1 }

FuzzyBCControllerEntry ::=
    SEQUENCE {
        FuzzyBCControllerNumberIdentifier    INTEGER,
        FuzzyBCControllerWeight             INTEGER,
        FuzzyBCControllerFuzzyClips        INTEGER,
        FuzzyBCControllerRuleNumbers        INTEGER,
        FuzzyBCControllerVarEntry           INTEGER,
        FuzzyBCControllerVarExit            INTEGER,
    }

FuzzyBCControllerNumberIdentifier OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro referente à ordem do bloco
        na tabela."
 ::= { FuzzyBCControllerEntry 1 }

FuzzyBCControllerWeight OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor INTEGER(0..100) cuja divisão por 100 refere-
        se ao peso do bloco junto ao conjunto."
 ::= { FuzzyBCControllerEntry 3 }

FuzzyBCControllerFuzzyClips OBJECT-TYPE
    SYNTAX INTEGER

```

```

ACCESS read-write
STATUS mandatory
DESCRIPTION
    "Valor 0(zero) se os números não forem fuzzy (por-
    tanto, números clássicos) e valor 1(um), caso con-
    trário."
 ::= { FuzzyBCControllerEntry 4 }

FuzzyBCControllerRuleNumbers OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Número de regras catalogadas dentro de cada
        bloco (controle)."
 ::= { FuzzyBCControllerEntry 5 }

FuzzyBCControllerVarEntry OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Numero de variável de entrada do bloco."
 ::= { FuzzyBCControllerEntry 6 }

FuzzyBCControllerVarExit OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DEFVAL 0
    DESCRIPTION
        "Numero de variável de saída do bloco."
 ::= { FuzzyBCControllerEntry 7 }

-- A implementação da tabela FuzzyRule armazena informações sobre todas as
-- regras catalogadas dentro de um Bloco da Base de Conhecimentos.

FuzzyRuleTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuzzyRuleEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Estrutura as regras de um controle fuzzy. Essas re-
        gras são compostas pela sua premissa e pela sua
        conclusão."
 ::= { FuzzyBC 3 }

FuzzyBCRuleEntry OBJECT-TYPE
    SYNTAX FuzzyRuleEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma entrada de uma regra que é utilizada em um blo-
        co."
    INDEX { FuzzyBCRuleControllerNumber, FuzzyBCRuleNumber }
 ::= { FuzzyBCRuleTable 1 }

FuzzyBCRuleEntry ::=
    SEQUENCE {
        FuzzyBCRuleControllerNumber      DisplayString,

```

```

        FuzzyBCRuleNumber          INTEGER,
        FuzzyBCRuleOperationEntry  INTEGER,
        FuzzyBCRuleOperationExit   INTEGER,
    }

FuzzyBCRuleControllerNumber OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Esse valor identifica o bloco ao qual esta regra
        pertence, ou seja, o objeto FuzzyBCControllerNumbe-
        rIdentifier."
    ::= { FuzzyBCRuleEntry 1 }

FuzzyBCRuleNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro referente à ordem da regra no con-
        junto de todas as regras pertencentes a esse
        bloco (controle)."
    ::= { FuzzyBCRuleEntry 2 }

FuzzyBCRuleOperationEntry OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Flag que indica uma operação lógica entre
        dois ou mais operandos de entrada".
        Se 0(zero) = nenhuma operação lógica;
        Se 1(um) = operação AND;
        Se 2(dois) = operação OR;
        Se 3(três) = operação NOT."
    ::= { FuzzyBCRuleEntry 3 }

FuzzyBCRuleOperationExit OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Flag que indica a operação lógica aplica-
        da sobre dois ou mais operandos de saída.
        Se 0(zero) = nenhuma operação;
        Se 1(um) = AND;
        Se 2(dois) = OR;
        Se 3(três) = NOT."
    ::= { FuzzyBCRuleEntry 4 }

-- A implementação da tabela FuzzyPremissa trata de armazenar informações
-- sobre todas as premissas das regras catalogadas dentro de um Bloco
-- (Controle) da Base de Conhecimentos.

FuzzyBCPremisseTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuzzyPremisseEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION

```

```

        "Uma lista das regras que são utilizadas no
        respectivo bloco em questão."
 ::= { FuzzyBC 4 }

FuzzyBCPremisseEntry OBJECT-TYPE
    SYNTAX FuzzyBCPremisseEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A entrada de uma regra que é utilizada em um blo-
        co."
    INDEX { FuzzyBCPremisseController, FuzzyBCPremisse-
            Rule,
            FuzzyBCPremisseNumber }
 ::= { FuzzyBCPremisseTable 1 }

FuzzyBCPremisseEntry ::=
    SEQUENCE {
        FuzzyBCPremisseController    DisplayString,
        FuzzyBCPremisseRule          INTEGER,
        FuzzyBCPremisseNumber        INTEGER,
        FuzzyBCPremisseValue         OBJECT IDENTIFI-
        FIER,
    }

FuzzyBCPremisseController OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro indicando o bloco (controle) ao
        qual esta regra pertence."
 ::= { FuzzyBCPremisseEntry 1 }

FuzzyBCPremisseRule OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro referente à ordem da regra (con-
        trole) no conjunto de todas as regras pertencen-
        tes a esse bloco."
 ::= { FuzzyBCPremisseEntry 2 }

FuzzyBCPremisseNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro referente à ordem da premissa no
        conjunto de entradas da regra."
 ::= { FuzzyBCPremisseEntry 3 }

FuzzyBCPremisseValue OBJECT-TYPE
    SYNTAX OBJECT IDENTIFIER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION

```

```

                                "Valor de FuzzyBCTermLingLimits, referente ao
                                par ordenado (VarLing, TermLing) para a en-
                                trada de cada regra."
 ::= { FuzzyBCPremisseEntry 4 }

-- A implementação da tabela FuzzyBCConclusion armazena informações sobre
-- todas as variáveis conclusivas das regras catalogadas dentro de um Bloco
-- (Controle) da Base de Conhecimentos.

FuzzyBCConclusionTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuzzyBCConclusionEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma lista das conclusões das regras que são
        utilizadas no respectivo bloco (controle) em
        questão."
 ::= { FuzzyBC 5 }

FuzzyBCConclusionEntry OBJECT-TYPE
    SYNTAX FuzzyBCConclusionEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Saída de uma regra que é utilizada em um
        bloco (controle)."
```

```
 ::= { FuzzyBCConclusionTable 1 }
```

```

FuzzyBCConclusionEntry ::=
    SEQUENCE {
        FuzzyBCConclusionControllerNumber Dis-
        playString,
        FuzzyBCConclusionRuleNumber INTEGER,
        FuzzyBCConclusionNumber INTEGER,
        FuzzyBCConclusionValue INTEGER,
    }

FuzzyBCConclusionControllerNumber OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro referente à ordem do bloco
        no conjunto de todos os blocos."
 ::= { FuzzyBCConclusionEntry 1 }

FuzzyBCConclusionRuleNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro referente à ordem da re-
        gra no conjunto de todas as regras de
        um bloco."
 ::= { FuzzyBCConclusionEntry 2 }

FuzzyBCConclusionNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
```

DESCRIPTION

"Valor inteiro referente à ordem da conclusão no conjunto de saídas da regra."
::= { FuzzyBCConclusionEntry 3 }

FuzzyBCCconclusionValue OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Valor do objeto FuzzyBCTermLingLimits, referente ao par ordenado (VarLing, TermLing) para a saída de cada regra."
::= { FuzzyBCConclusionEntry 4 }

END

```

-- the NeuralBC group

-- Implementation of the NeuralBC group is mandatory for all systems.
-- If an agent is not configured to have a value for any of these
-- variables, a string of length 0 is returned.

NeuralBCNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Numero de Redes Neurais Artificiais presentes na base
        de conhecimento neural."
    ::= { NeuralBC 1 }

NeuralHostName OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Nome do computador onde o kernel do simulador está
        executando."
    ::= { NeuralBC 2 }

KernelNeuralVersion OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O número da versão do kernel do simulador. Esta
        informação também pode ser mostrada com 'rpcinfo
        -p'."
    ::= { NeuralBC 3 }

StatusNeuralKernel OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O último status conhecido do kernel do simulador."
    ::= { NeuralBC 4 }

DomainNeuralName OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O domínio da rede onde se encontra o kernel do
        simulador."
    ::= { NeuralBC 5 }

HostNeuralAddress OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O endereço do host do kernel do simulador."
    ::= { NeuralBC 6 }

K-TypeNeuralSimulator OBJECT-TYPE

```

```

        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "O tipo do kernel do simulador. Tipos possíveis:
        'coop-master' e 'normal'."
        ::= { NeuralBC 7 }

KernelIDNeuralSimulator OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "Identificador do kernel do simulador no xgui."
        ::= { NeuralBC 8 }

CyclesNeural OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "Número de ciclos usado para o treinamento da rede
        neural artificial."
        ::= { NeuralBC 9 }

EpochNeural OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "Número de épocas usado para o treinamento da rede neural
        artificial."
        ::= { NeuralBC 10 }

SSENeural OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "O valor da Soma de Erros Quadrados (Sum Square Error -
        SSE) obtido pela Rede Neural Artificial. O valor aqui ar-
        mazenado deve ser dividido por 1.000 para refletir o va-
        lor exato."
        ::= { NeuralBC 11 }

MSENeural OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
        "O valor do Erro Médio Quadrado (Mean Square Error - MSE)
        obtido pela Rede Neural Artificial. O valor aqui armaze-
        nado deve ser dividido por 1.000 para refletir o valor
        exato."
        ::= { NeuralBC 12 }

SSEPUNeural OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-write
        STATUS mandatory

```

```

DESCRIPTION
    "O valor da Soma de Erros Quadrados (Sum Square Error
    - SSE) obtido pela Rede Neural Artificial. O valor
    aqui armazenado deve ser dividido por 1.000 para
    refletir o valor exato."
 ::= { NeuralBC 13 }

PatternNeuralNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O número de padrões da Rede Neural Artificial."
 ::= { NeuralBC 14 }

UnitsNeuralNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O número de neurônios da Rede Neural Artificial."
 ::= { NeuralBC 15 }

InputNeuralNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O número de neurônios de entrada da Rede Neural Ar-
        tificial."
 ::= { NeuralBC 16 }

HiddenNeuralNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O número de neurônios da camada intermediária da Re-
        de Neural Artificial."
 ::= { NeuralBC 17 }

OutputNeuralNumber OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O número de neurônios da camada de saída da Rede
        Neural Artificial."
 ::= { NeuralBC 18 }

NetNeuralName OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O nome da Rede Neural Artificial."
 ::= { NeuralBC 19 }

PatternNeuralNetName OBJECT-TYPE
    SYNTAX DisplayString

```

```

ACCESS read-write
STATUS mandatory
DESCRIPTION
    "O nome das configurações padrões da Rede Neural
    Artificial."
::= { NeuralBC 20 }

InitializationNeuralFunction OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "A função de aprendizado da Rede Neural Artifi-
    cial.
    As funções usadas para o aprendizado da Rede
    Neural Artificial são:
    01 = ART1_Weights
    02 = ART2_Weights
    03 = ARTMAP_Weights
    04 = CC_Weights
    05 = ClippHebb
    06 = CPN_Weights_v3.2
    07 = CPN_Weights_v3.3
    08 = CPN_Rand_Pat
    09 = DLVQ_Weights
    10 = Hebb
    11 = Hebb_Fixed_Act
    12 = JE_Weights
    13 = Kohonen_Rand_Pat
    14 = Kohonen_Weights_v3.2
    15 = Kohonen_Const
    16 = PseudoInv
    17 = Random_Weights_Perc
    18 = Randomize_Weights
    19 = RBF_Weights
    20 = RBF_Weights_Kohonen
    21 = RBF_Weights_Redo
    22 = RCC_Weights
    23 = RM_Random_Weights"
::= { NeuralBC 21 }

NoOfInitNeuralParam OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "Número de parâmetros de inicialização da Rede
    Neural Artificial."
::= { NeuralBC 22 }

LearningNeuralFunction OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "A função de aprendizado da Rede Neural Artifi-
    cial.
    As funções usadas para o aprendizado da Rede
    Neural Artificial são:
    01 = ART1

```

```

02 = ART2
03 = ARTMAP
04 = BackPercolation
05 = BackpropBatch
06 = BackpropMomentum
07 = BackpropWeightDecay
08 = BPTT
09 = BBPTT
10 = CC
11 = Counterpropagation
12 = Dynamic_LVQ
13 = Hebbian
14 = JE_BP
15 = JE_BP_Momentum
16 = JE_Quickprop
17 = JE_Rprop
18 = Kohonen
19 = Monte-Carlo
20 = PruningFeedForward
21 = QPTT
22 = Quickprop
23 = RadialBasisLearning
24 = RBF-DDA
25 = RCC
26 = RM-delta
27 = Rprop
28 = Sim_Ann_SS
29 = Sim_Ann_WTA
30 = Sim_Ann_WWTA
31 = Std_Backpropagation
32 = TimeDelayBackprop"
 ::= { NeuralBC 23 }

```

UpdateNeuralFunction OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"A função de atualização da Rede Neural Artificial.

As funções usadas para a atualização da Rede Neural Artificial são:

```

01 = Topological Order
02 = ART1_Stable
03 = ART1_Synchronous
04 = ART2_Stable
05 = ART2_Synchronous
06 = ARTMAP_Stable
07 = ARTMAP_Synchronous
08 = Auto_Synchronous
09 = BAM_Order
10 = BPTT_Order
11 = CounterPropagation
12 = Dynamic_LVQ
13 = Hopfield_Fixed_Act
14 = Hopfield_Synchronous
15 = JE_Order
16 = JE_Special
17 = Kohonen_Order
18 = Random_Order

```

```

        19 = Random_Permutation
        20 = RCC_Order
        21 = Synchronous_Order
        22 = TimeDelay_Order"
 ::= { NeuralBC 24 }

PruningNeuralFunctionOne OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "A primeira função para o pruning da Rede
        Neural Artificial.
        Classes de Funções de Pruning:
        1 = MagPruning
        2 = Noncontributing_Units
        3 = OptimalBrainSurgeon
        4 = Skeletonization
        5 = OptimalBrainDamage"
 ::= { NeuralBC 25 }

PruningNeuralFunctionTwo OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "A segunda função para o pruning da Rede Neural
        Artificial.
        Classes de Funções de Pruning:
        1 = BackpropBatch
        2 = BackpropMomentum
        3 = Quickprop
        4 = Rprop
        5 = BackpropWeightDecay
        6 = Std_Backpropagation"
 ::= { NeuralBC 26 }

BatchmanNeuralFile OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Arquivo .bat para execução do batchman.exe da
        Rede Neural Artificial."
 ::= { NeuralBC 27 }

NetworkNeuralFile OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Arquivo .net de construção da Rede Neural Artificial."
 ::= { NeuralBC 28 }

PatternNeuralFile OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Arquivo .pat de treinamento da Rede Neural Artificial."

```

```

 ::= { NeuralBC 29 }

NoOfVarDimNeural OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "O número de dimensões que compõe o Esquema de Sub-padrão de
        deslocamento (Subpattern-Shifting-Scheme). A definição des-
        se esquema tem que adequar-se ao arquivo padrão usado e a
        arquitetura da Rede Neural Artificial.
        O formato da função de execução do simulador, referencia
        nesse por esse Objeto da MIB é:
        setSubPattern(InputSize1, InputStep1, OutputSize1, Output-
        Step1[,InputSize2, InputStep2, OutputSize2, OutputStep2])"
 ::= { NeuralBC 30 }

SubPatternISize OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa o tamanho do padrão da camada de entrada da Rede
        Neural Artificial."
 ::= { NeuralBC 31 }

SubPatternOSize OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa o tamanho do padrão da camada de saída da Rede Neu-
        ral Artificial."
 ::= { NeuralBC 32 }

SubPatternIStep OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa o subpadrão do passo de treinamento para a camada
        de entrada da Rede Neural Artificial."
 ::= { NeuralBC 32 }

SubPatternOStep OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa o subpadrão do passo de treinamento para a camada
        de saída da Rede Neural Artificial."
 ::= { NeuralBC 33 }

MaxLearnCycles OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa o número máximo de ciclo que a Rede Neural Artifi-
        cial deve executar durante o treinamento."

```

```

    ::= { NeuralBC 34 }

MaxErroToStop OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa a taxa de erro máxima tolerável pela Rede Neural
        Artificial."
    ::= { NeuralBC 35 }

Shuffle OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa se habilita ou não a seleção da próxima estrutura
        de treinamento da Rede Neural Artificial de forma randômi-
        ca".
        Os valores usados são:
        0 = FALSE
        1 = TRUE."
    ::= { NeuralBC 36 }

SubShuffle OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa se habilita ou não a seleção da próxima sub-
        estrutura de treinamento da Rede Neural Artificial de forma
        randômica.
        Os valores usados são:
        0 = FALSE
        1 = TRUE."
    ::= { NeuralBC 37 }

NoOfLearnParameter OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Informa o número de parâmetros de treinamento usados na
        função setLearnFunction(function name, parameters...) para
        o treinamento da Rede Neural Artificial."
    ::= { NeuralBC 38 }

ExitNeuralCode OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Status de saída de uma chamada de execução do trei-
        namento da Rede Neural Artificial."
    ::= { NeuralBC 39 }

ResponseNeuralFile OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory

```

```

        DESCRIPTION
        "Arquivo .res com o resultado do treinamento da Rede Neural
        Artificial."
        ::= { NeuralBC 40 }

ResponseParamRequired OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
    "Informa se há ou não parâmetros necessários quando da ma-
    nipulação do arquivo de resposta .res referente a Rede Neu-
    ral Artificial.
    A manipulação é feita pela função: saveResult(objeto Res-
    ponseNeuralFile, paramStart, paramEnd, paramInclIn, para-
    mInclOut, paramFileMode).
    Os valores possíveis são:
    0 = FALSE
    1 = TRUE."
    ::= { NeuralBC 41 }

-- the InitNeuralBCTable table
-- A tabela InitNeuralBCTable contém as informações necessárias à
-- inicialização da Rede Neural Artificial. Tais informações são
-- utilizadas no Módulo de Inteligência do simulador de tráfego
-- telefônico.

InitNeuralBCTable OBJECT-TYPE
    SYNTAX SEQUENCE OF InitNeuralBCEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma lista das informações de inicialização da es-
        trutura de configuração neural para a construção
        da Rede Neural Artificial."
    ::= { NeuralBC 42 }

InitNeuralBCEntry OBJECT-TYPE
    SYNTAX InitNeuralBCEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma entrada da tabela de informações neurais que é
        utilizado na rede neural do Módulo de Inteligên-
        cia."
    INDEX { InitParamIdentifierBC }
    ::= { InitNeuralBCTable 1 }

InitNeuralBCEntry ::=
    SEQUENCE {
        InitParamIdentifierBC          INTEGER,
        InitParameterBC              INTEGER,
    }

InitParamIdentifierBC OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION

```

```

        "Valor inteiro referente ao índice do parâmetro na
        tabela."
        ::= { InitNeuralBCEntry 1 }

InitParameterBC OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor referente ao parâmetro de inicialização da Re-
        de Neural Artificial na tabela."
        ::= { InitNeuralBCEntry 2 }

-- the LearnParamNeuralBCTable table
-- A tabela LearnParamNeuralBCTable contém as informações necessárias
-- ao treinamento da Rede Neural Artificial. Tais informações são
-- utilizadas no Módulo de Inteligência do simulador de tráfego
-- telefônico.

LearnParamNeuralBCTable OBJECT-TYPE
    SYNTAX SEQUENCE OF LearnParamNeuralBCEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma lista das informações de treinamento da estru-
        tura de configuração neural para a construção da
        Rede Neural Artificial."
        ::= { NeuralBC 43 }

LearnParamNeuralBCEntry OBJECT-TYPE
    SYNTAX LearnParamNeuralBCEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma entrada da tabela de informações de aprendiza-
        do neural que é utilizado na rede neural do Módulo
        de Inteligência."
    INDEX { LearnParamIdentifierBC }
    ::= { LearnParamNeuralBCTable 1 }

LearnNeuralBCEntry ::=
    SEQUENCE {
        LearnParamIdentifierBC    INTEGER,
        LearnParameterBC         INTEGER,
    }

LearnParamIdentifierBC OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro referente ao índice do parâmetro na
        tabela."
        ::= { LearnNeuralBCEntry 1 }

LearnParameterBC OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION

```

```

        "Valor referente ao parâmetro de aprendizado da Rede
        Neural Artificial na tabela."
 ::= { LearnNeuralBCEntry 2 }

-- the ResponseParamNeuralBCTable table
-- A tabela ResponseParamNeuralBCTable contém as informações
-- necessárias ao treinamento da Rede Neural Artificial. Tais
-- informações são utilizadas no Módulo de Inteligência do simulador de
-- tráfego telefônico.

ResponseParamNeuralBCTable OBJECT-TYPE
    SYNTAX SEQUENCE OF ResponseParamNeuralBCEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma lista das informações de configuração do ar-
        quivo de resposta da estrutura de configuração
        neural para a construção da Rede Neural Artifici-
        al.
        Os dados dessa tabela são úteis para a execução da
        função: saveResult(ResponseNeuralFile, ResponsePa-
        ramStartBC, ResponseParamEndBC,
        ResponseParamInclúdeInBC, ResponseParamInclude-
        OutBC, ResponseParamFileModeBC)."
 ::= { NeuralBC 44 }

ResponseParamNeuralBCEntry OBJECT-TYPE
    SYNTAX ResponseParamNeuralBCEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Uma entrada da tabela de informações de resposta
        neural que é utilizado na rede neural do Módulo de
        Inteligência."
    INDEX { ResponseParamIdentifierBC }
 ::= { ResponseParamNeuralBCTable 1 }

ResponseNeuralBCEntry ::=
    SEQUENCE {
        ResponseParamIdentifierBC    INTEGER,
        ResponseParamStartBC        INTEGER,
        ResponseParamEndBC          INTEGER,
    }

ResponseParamIdentifierBC OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor inteiro referente ao índice do parâmetro na
        tabela."
 ::= { ResponseNeuralBCEntry 1 }

ResponseParamStartBC OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Valor referente ao primeiro padrão que será tratado
        como resposta da Rede Neural Artificial na tabela."

```

Se o usuário desejar tratar todos os padrões, pode-se configurar para a variável PAT (= 10)."
 ::= { ResponseNeuralBCEntry 2 }

ResponseParamEndBC OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Valor referente ao último padrão que será tratado como resposta da Rede Neural Artificial na tabela."
 ::= { ResponseNeuralBCEntry 3 }

ResponseParamIncludeInBC OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Informa se a estrutura padrão de entrada deve ser salva no arquivo de resposta (.res).
Os valores são:
0 = FALSE
1 = TRUE."
 ::= { ResponseNeuralBCEntry 3 }

ResponseParamIncludeOutBC OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Informa se a estrutura padrão de saída deve ser salva no arquivo de resposta (.res).
Os valores são:
0 = FALSE
1 = TRUE."
 ::= { ResponseNeuralBCEntry 4 }

ResponseParamFileModeBC OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Informa se um arquivo de resposta (.res) deve ser criado, ou se ele deve apenas ser adicionado em um arquivo existente.
Os valores são:
0 = 'create'
1 = 'append'."
 ::= { ResponseNeuralBCEntry 5 }

G - AGENTE DE DADOS INTELIGENTE

.1.3.6.1.4.1.1.1.1.0 ERBNumber 2 INTEGER
.1.3.6.1.4.1.1.1.2.1.1.1 ERBIdentifier 200.129.0.41 PhysAddress
.1.3.6.1.4.1.1.1.2.1.1.2 ERBIdentifier 200.129.0.43 PhysAddress
.1.3.6.1.4.1.1.1.2.1.2.200.129.0.41 ERBTTotalChannels 32 INTEGER
.1.3.6.1.4.1.1.1.2.1.2.200.129.0.43 ERBTTotalChannels 32 INTEGER
.1.3.6.1.4.1.1.1.2.1.3.200.129.0.41 ERBGPRS 4 INTEGER
.1.3.6.1.4.1.1.1.2.1.3.200.129.0.43 ERBGPRS 4 INTEGER
.1.3.6.1.4.1.1.1.2.1.4.200.129.0.41 ERBSignaling 3 INTEGER
.1.3.6.1.4.1.1.1.2.1.4.200.129.0.43 ERBSignaling 3 INTEGER
.1.3.6.1.4.1.1.1.2.1.5.200.129.0.41 ERBHandovers 4 INTEGER
.1.3.6.1.4.1.1.1.2.1.5.200.129.0.43 ERBHandovers 4 INTEGER
.1.3.6.1.4.1.1.1.2.1.6.200.129.0.41 ERBRCS 0 INTEGER
.1.3.6.1.4.1.1.1.2.1.6.200.129.0.43 ERBRCS 1 INTEGER
.1.3.6.1.4.1.1.1.2.1.7.200.129.0.41 ERBQPS 1 INTEGER
.1.3.6.1.4.1.1.1.2.1.7.200.129.0.43 ERBQPS 2 INTEGER
.1.3.6.1.4.1.1.1.2.1.8.200.129.0.41 ERBSRS 0 INTEGER
.1.3.6.1.4.1.1.1.2.1.8.200.129.0.43 ERBSRS 10 INTEGER
.1.3.6.1.4.1.1.2.1.0 FuzzyNumber 8 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.1 FuzzyVarLingNumber 1 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.2 FuzzyVarLingNumber 2 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.3 FuzzyVarLingNumber 3 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.4 FuzzyVarLingNumber 4 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.5 FuzzyVarLingNumber 5 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.6 FuzzyVarLingNumber 6 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.7 FuzzyVarLingNumber 7 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.8 FuzzyVarLingNumber 8 INTEGER
.1.3.6.1.4.1.1.2.2.1.2.1 FuzzyVarLingName Offered_Traffic DisplayString
.1.3.6.1.4.1.1.2.2.1.2.2 FuzzyVarLingName GPRS_Channels DisplayString
.1.3.6.1.4.1.1.2.2.1.2.3 FuzzyVarLingName Handover_Channels DisplayString
.1.3.6.1.4.1.1.2.2.1.2.4 FuzzyVarLingName SRS_Channels DisplayString
.1.3.6.1.4.1.1.2.2.1.2.5 FuzzyVarLingName percentual_Voice DisplayString
.1.3.6.1.4.1.1.2.2.1.2.6 FuzzyVarLingName percentual_Handover DisplayString
.1.3.6.1.4.1.1.2.2.1.2.7 FuzzyVarLingName RCS DisplayString
.1.3.6.1.4.1.1.2.2.1.2.8 FuzzyVarLingName QPS DisplayString
.1.3.6.1.4.1.1.2.2.1.3.1 FuzzyVarLingTermNumber 5 INTEGER
.1.3.6.1.4.1.1.2.2.1.3.2 FuzzyVarLingTermNumber 5 INTEGER
.1.3.6.1.4.1.1.2.2.1.3.3 FuzzyVarLingTermNumber 5 INTEGER
.1.3.6.1.4.1.1.2.2.1.3.4 FuzzyVarLingTermNumber 3 INTEGER
.1.3.6.1.4.1.1.2.2.1.3.5 FuzzyVarLingTermNumber 3 INTEGER
.1.3.6.1.4.1.1.2.2.1.3.6 FuzzyVarLingTermNumber 3 INTEGER
.1.3.6.1.4.1.1.2.2.1.3.7 FuzzyVarLingTermNumber 2 INTEGER
.1.3.6.1.4.1.1.2.2.1.3.8 FuzzyVarLingTermNumber 3 INTEGER
.1.3.6.1.4.1.1.2.2.1.1.1 FuzzyTermLingVar .1.3.6.1.4.1.1.2.2.1.1.1 OBJECT_IDENTIFIER
.1.3.6.1.4.1.1.2.2.1.1.2 FuzzyTermLingVar .1.3.6.1.4.1.1.2.2.1.1.2 OBJECT_IDENTIFIER
.1.3.6.1.4.1.1.2.2.1.1.3 FuzzyTermLingVar .1.3.6.1.4.1.1.2.2.1.1.3 OBJECT_IDENTIFIER
.1.3.6.1.4.1.1.2.2.1.1.4 FuzzyTermLingVar .1.3.6.1.4.1.1.2.2.1.1.4 OBJECT_IDENTIFIER
.1.3.6.1.4.1.1.2.2.1.1.5 FuzzyTermLingVar .1.3.6.1.4.1.1.2.2.1.1.5 OBJECT_IDENTIFIER
.1.3.6.1.4.1.1.2.2.1.1.6 FuzzyTermLingVar .1.3.6.1.4.1.1.2.2.1.1.6 OBJECT_IDENTIFIER
.1.3.6.1.4.1.1.2.2.1.1.7 FuzzyTermLingVar .1.3.6.1.4.1.1.2.2.1.1.7 OBJECT_IDENTIFIER
.1.3.6.1.4.1.1.2.2.1.1.8 FuzzyTermLingVar .1.3.6.1.4.1.1.2.2.1.1.8 OBJECT_IDENTIFIER
.1.3.6.1.4.1.1.2.3.1.2.1 FuzzyTermLingName baixo DisplayString

.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.5.baixo FuzzyTermLingHeights 1;0;0 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.5.medio FuzzyTermLingHeights 0;1;0 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.5.alto FuzzyTermLingHeights 0;1;1 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.6.baixo FuzzyTermLingHeights 1;0;0 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.6.medio FuzzyTermLingHeights 0;1;0 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.6.alto FuzzyTermLingHeights 0;1;1 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.7.pos FuzzyTermLingHeights 1 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.7.pre FuzzyTermLingHeights 1 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.8.FIFO FuzzyTermLingHeights 1 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.8.MBP FuzzyTermLingHeights 1 DisplayString
.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.2.1.1.8.none FuzzyTermLingHeights 1 DisplayString
.1.3.6.1.4.1.1.3.1.1 FuzzyBCBlockNumber 5 INTEGER
.1.3.6.1.4.1.1.3.1.2 FuzzyBCBlockNumber 5 INTEGER
.1.3.6.1.4.1.1.3.2.1.1.1 FuzzyBCBlockNumberIdentifier 1 INTEGER
.1.3.6.1.4.1.1.3.2.1.1.2 FuzzyBCBlockNumberIdentifier 2 INTEGER
.1.3.6.1.4.1.1.3.2.1.1.3 FuzzyBCBlockNumberIdentifier 3 INTEGER
.1.3.6.1.4.1.1.3.2.1.1.4 FuzzyBCBlockNumberIdentifier 3 INTEGER
.1.3.6.1.4.1.1.3.2.1.1.5 FuzzyBCBlockNumberIdentifier 3 INTEGER
.1.3.6.1.4.1.1.3.2.1.2.1 FuzzyBCBlockWeight 100 INTEGER
.1.3.6.1.4.1.1.3.2.1.2.2 FuzzyBCBlockWeight 100 INTEGER
.1.3.6.1.4.1.1.3.2.1.2.3 FuzzyBCBlockWeight 100 INTEGER
.1.3.6.1.4.1.1.3.2.1.2.4 FuzzyBCBlockWeight 100 INTEGER
.1.3.6.1.4.1.1.3.2.1.2.5 FuzzyBCBlockWeight 100 INTEGER
.1.3.6.1.4.1.1.3.2.1.3.1 FuzzyBCBlockClips 1 INTEGER
.1.3.6.1.4.1.1.3.2.1.3.2 FuzzyBCBlockClips 1 INTEGER
.1.3.6.1.4.1.1.3.2.1.3.3 FuzzyBCBlockClips 1 INTEGER
.1.3.6.1.4.1.1.3.2.1.3.4 FuzzyBCBlockClips 1 INTEGER
.1.3.6.1.4.1.1.3.2.1.3.5 FuzzyBCBlockClips 1 INTEGER
.1.3.6.1.4.1.1.3.2.1.4.1 FuzzyBCBlockRuleNumbers 3 INTEGER
.1.3.6.1.4.1.1.3.2.1.5.1 FuzzyBCBlockVarEntry 2 INTEGER
.1.3.6.1.4.1.1.3.2.1.6.1 FuzzyBCBlockVarExit 5 INTEGER
.1.3.6.1.4.1.1.3.3.1.1.1 FuzzyRuleBlockNumber 1 DisplayString
.1.3.6.1.4.1.1.3.3.1.2.1 FuzzyRuleNumber 5 INTEGER
.1.3.6.1.4.1.1.3.3.1.3.1.1.1 FuzzyRuleOperationEntry 1 INTEGER
.1.3.6.1.4.1.1.3.3.1.3.1.1.2 FuzzyRuleOperationEntry 1 INTEGER
.1.3.6.1.4.1.1.3.3.1.3.1.2.1 FuzzyRuleOperationEntry 1 INTEGER
.1.3.6.1.4.1.1.3.3.1.3.1.2.2 FuzzyRuleOperationEntry 1 INTEGER
.1.3.6.1.4.1.1.3.3.1.3.1.3.1 FuzzyRuleOperationEntry 1 INTEGER
.1.3.6.1.4.1.1.3.3.1.3.1.3.2 FuzzyRuleOperationEntry 1 INTEGER
.1.3.6.1.4.1.1.3.3.1.4.1.1.1 FuzzyRuleOperationExit 0 INTEGER
.1.3.6.1.4.1.1.3.3.1.4.1.1.2 FuzzyRuleOperationExit 0 INTEGER
.1.3.6.1.4.1.1.3.3.1.4.1.2.1 FuzzyRuleOperationExit 0 INTEGER
.1.3.6.1.4.1.1.3.3.1.4.1.2.2 FuzzyRuleOperationExit 0 INTEGER
.1.3.6.1.4.1.1.3.3.1.4.1.3.1 FuzzyRuleOperationExit 0 INTEGER
.1.3.6.1.4.1.1.3.3.1.4.1.3.2 FuzzyRuleOperationExit 0 INTEGER
.1.3.6.1.4.1.1.3.4.1.1.1 FuzzyPremisseBlock 1 DisplayString
.1.3.6.1.4.1.1.3.4.1.2.1 FuzzyPremisseRule 1 INTEGER
.1.3.6.1.4.1.1.3.4.1.2.2 FuzzyPremisseRule 2 INTEGER
.1.3.6.1.4.1.1.3.4.1.2.3 FuzzyPremisseRule 3 INTEGER
.1.3.6.1.4.1.1.3.4.1.3.1 FuzzyPremisseNumber 1 INTEGER
.1.3.6.1.4.1.1.3.4.1.3.2 FuzzyPremisseNumber 2 INTEGER
.1.3.6.1.4.1.1.3.4.1.3.3 FuzzyPremisseNumber 3 INTEGER
.1.3.6.1.4.1.1.3.4.1.4.1.1.1 FuzzyPremisseValue
.1.3.6.1.4.1.1.2.2.1.1.5.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.5.medio OBJECTIDENTIFIER

.1.3.6.1.4.1.1.3.4.1.4.2.5.2 FuzzyPremisseValue
.1.3.6.1.4.1.1.2.2.1.1.6.1.3.6.1.4.1.1.2.3.1.4.1.3.6.1.4.1.1.2.3.1.1.6.medio OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.4.1.4.2.5.3 FuzzyPremisseValue
.1.3.6.1.4.1.1.2.2.1.1.1.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.1.alto OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.1.1 FuzzyConclusionBlockNumber 1 INTEGER
.1.3.6.1.4.1.1.3.5.1.2.1 FuzzyConclusionRuleNumber 1 INTEGER
.1.3.6.1.4.1.1.3.5.1.2.2 FuzzyConclusionRuleNumber 2 INTEGER
.1.3.6.1.4.1.1.3.5.1.2.3 FuzzyConclusionRuleNumber 3 INTEGER
.1.3.6.1.4.1.1.3.5.1.2.4 FuzzyConclusionRuleNumber 3 INTEGER
.1.3.6.1.4.1.1.3.5.1.2.5 FuzzyConclusionRuleNumber 3 INTEGER
.1.3.6.1.4.1.1.3.5.1.3.1 FuzzyConclusionNumber 1 INTEGER
.1.3.6.1.4.1.1.3.5.1.3.2 FuzzyConclusionNumber 2 INTEGER
.1.3.6.1.4.1.1.3.5.1.3.3 FuzzyConclusionNumber 3 INTEGER
.1.3.6.1.4.1.1.3.5.1.3.4 FuzzyConclusionNumber 4 INTEGER
.1.3.6.1.4.1.1.3.5.1.3.5 FuzzyConclusionNumber 5 INTEGER
.1.3.6.1.4.1.1.3.5.1.4.1.1.1 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.2.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.2.alto OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.1.2 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.3.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.3.baixo OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.1.3 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.7.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.7.pre OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.1.4 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.8.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.8.noQPS OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.1.5 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.4.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.4.baixo OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.2.1 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.2.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.2.alto OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.2.2 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.3.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.3.baixo_medio OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.2.3 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.7.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.7.pre OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.2.4 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.8.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.8.noQPS OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.2.5 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.4.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.4.baixo OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.3.1 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.2.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.2.medio OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.3.2 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.3.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.3.medio_alto OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.3.3 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.7.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.7.pre OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.3.4 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.8.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.8.FIFO OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.3.5 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.4.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.4.baixo_medio OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.4.1 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.2.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.2.baixo OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.4.2 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.3.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.3.medio_alto OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.4.3 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.7.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.7.pos OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.4.4 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.8.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.8.FIFO OBJECTIDENTIFIER

.1.3.6.1.4.1.1.3.5.1.4.1.4.5 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.4.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.4.medio OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.5.1 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.2.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.2.baixo OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.5.2 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.3.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.3.alto OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.5.3 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.7.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.7.pos OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.5.4 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.8.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.8.MBP OBJECTIDENTIFIER
.1.3.6.1.4.1.1.3.5.1.4.1.5.5 FconclusionValue
.1.3.6.1.4.1.1.2.2.1.1.4.1.3.6.1.4.1.1.2.3.1.3.1.3.6.1.4.1.1.2.3.1.1.4.medio OBJECTIDENTIFIER

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.