

**Marta Emília Barria Martínez**

**ALGORITMOS PARA QoS EM REDES DE  
COMPUTADORES**

**Tese de Doutorado**

**Tese apresentada ao Departamento de  
Informática da PUC-Rio, como parte dos  
requisitos necessários para a obtenção do  
título de Doutor em Ciências da Informática.**

**Orientador: Prof. Luiz Fernando G. Soares**

Departamento de Informática  
Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 19 de Março de 2001

Ao meu marido e minhas filhas

## Meus Agradecimentos

- ◆ A meu marido por seu amor e apoio continuo durante toda nossa vida juntos.
- ◆ A minhas filhas por todo seu amor.
- ◆ Ao professor L. F. G. Soares por sua confiança e apoio nas distintas etapas do desenvolvimento desta tese.
- ◆ A minha querida amiga Iara Regina por sua amizade.

# Resumo

Um dos problemas relacionados com a avaliação de redes que oferecem garantias de QoS, é a falta de modelos simples que permitam obter uma primeira aproximação de parâmetros de QoS para os distintos tráfegos que transporta a rede.

O objetivo global desta tese é melhorar algumas das técnicas utilizadas na transmissão de diferentes tipos de tráfegos, através de uma rede de computadores, com o intuito de fornecer uma melhor QoS. Para isto, são propostos algoritmos, modelos, metodologias de análise de desempenho, e a análise de desempenho dos algoritmos propostos, calculando as medidas de desempenho de interesse de cada caso particular.

Foram analisados algoritmos tanto para redes comutadas por pacotes, exemplificadas pelas redes TCP/IP, quanto redes com comutação de rótulos, exemplificados pela rede ATM. Especificamente, os parâmetros de qualidade de serviço de interesse foram a probabilidade de perda de células e o retardo no sistema. Entretanto, outras medidas de desempenho foram também calculadas, como diversas funções de distribuição.

# Abstract

One of the problems related to the evaluation of networks that provide guarantees of QoS is the lack of simple models that would generate a first approximation of QoS parameters for the different types of traffic transported by the network.

The overall objective of this thesis is to suggest some improvements in the techniques used in the transmission of different types of traffic across a computer network in order to achieve a better QoS. To this end, various algorithms, models and performance analysis methodologies are proposed, and the latter are then applied to the performance analysis of the proposed algorithms, calculating the most relevant performance measures in each particular case.

The algorithms analyzed include those of the type used in packet-switching networks, such as TCP/IP, as well as those used in label switching networks, exemplified by the ATM network. The QoS parameters of greatest interest were the probability of loss of cells and system delay. Other performance measures were also derived, including various probability distribution functions.

# Tabela de Conteúdos

<b>LISTA DE FIGURAS</b> .....	<b>.IX</b>
<b>LISTA DE TABELAS</b> .....	<b>.XII</b>
<b>CAPÍTULO I INTRODUÇÃO</b> .....	<b>1</b>
I.1 OBJETIVOS E CONTRIBUIÇÕES DA TESE .....	3
I. 2 ORGANIZAÇÃO DA TESE .....	6
<b>CAPÍTULO II CONCEITOS UTILIZADOS</b> .....	<b>.9</b>
II.1 QoS EM REDES ATM.....	9
II.1.1 <i>Gerenciamento do Tráfego em Redes ATM</i> .....	10
II.1.2 <i>Qualidade de Serviço (QoS)</i> .....	11
II.1.2.1 Parâmetros de QoS .....	11
II.2 TAXONOMIA DE ALGORITMOS DE ESCALONAMENTO DE BANDA PASSANTE.....	12
II.2.1 <i>Algoritmos de Atribuição de Banda Passante Baseados em Prioridade</i> .....	15
II.2.2 <i>Mecanismo de Escalonamento de Banda Passante Baseado em Acesso Justo (Fair-Share)</i> .....	16
II.2.3 <i>Descrição dos Algoritmos de Atribuição de Banda Passante</i> .....	18
II.2.3.1 Mecanismos de Escalonamento Baseados em Prioridade.....	19
II.2.3.2 Mecanismos de Escalonamento Baseados em Acesso Justo Conservando o Trabalho .....	20
II.2.3.3 Mecanismos de Escalonamento Baseados em Acesso Justo de Trabalho Não Conservativo.....	24
II.3 ESTADO DA ARTE SOBRE ALGORITMOS DE CONGESTIONAMENTO .....	26
II.3.1 <i>Descarte pela Cauda (Dropping Tail (DT))</i> .....	28
II.3.2 <i>Descarte pela Frente da Fila (Front Dropping (FD))</i> .....	28
II.4 CONCEITOS SOBRE O TRÁFEGO DE VOZ.....	29
II.4.1 <i>Voz sobre ATM</i> .....	29
II.4.2 <i>Características do Tráfego de Voz</i> .....	29
II.4.2.1 Retardo Fim-a-Fim .....	29
II.4.2.2 Supressão de Silêncio .....	30
II.5 RESUMO .....	31
<b>CAPÍTULO III META-ALGORITMO PARA POLÍTICAS DE ESCALONAMENTO</b> .....	<b>32</b>
III.1 INTRODUÇÃO.....	32
III.2. DESCRIÇÃO DO META-ALGORITMO LDS .....	34
III.3 EMULAÇÃO DE ALGORITMOS USANDO O META-ALGORITMO LDS.....	37
III.3.1 <i>Emulação de Algoritmos de Escalonamento com Prioridade</i> .....	37
III.3.2 <i>Emulação de Algoritmos de Escalonamento Tipo Round-Robin (RR)</i> .....	38
III.4 ALGORITMOS DE ESCALONAMENTO PROPOSTOS NESTA TESE .....	39

III.4.1 Transmissão proporcional.....	39
III.4.2 Algoritmo Preditivo Baseado na Estatística das Chegadas.....	40
III.4.3 Algoritmo Preditivo baseado na chegada do ciclo anterior .....	41
III.4.4 Algoritmo que aproxima a GPS .....	42
III.5 RESUMO.....	42
<b>CAPÍTULO IV ANÁLISE DE DESEMPENHO DE MULTIPLEXADORES.....</b>	<b>43</b>
IV.1 SOLUÇÃO USANDO <i>PERFORMABILITY</i> .....	43
IV.2 ANÁLISE DE DESEMPENHO DE UM MULTIPLEXADOR PARA TRÁFEGO DE VOZ.....	45
IV.2.1 Introdução.....	45
IV.2.2 Operação do Multiplexador ATM para Tráfego de Voz .....	48
IV.2.3 O Modelo Genérico e sua Solução Geral.....	50
IV.2.3.1 O Modelo Genérico.....	50
IV.2.3.2 Cálculo do autovetor $\pi$ .....	52
IV.2.3.3 Instanciação do Modelo Usando um Processo de Poisson para o Processo de Chegadas.....	53
IV.2.4 Avaliação de Medidas de Desempenho.....	55
IV.2.4.1 Número Médio de Pares de Células no Multiplexador .....	55
IV.2.4.2 Cálculo do Tempo Médio de Permanência no Multiplexador, Tempo Médio na Fila e Número Médio de Pares de Células na Fila do Multiplexador.....	60
IV.2.4.3 Avaliação de Medidas de Desempenho que Caracterizam um Período de Congestionamento .....	62
IV.2.5 Exemplos Numéricos.....	75
IV.2.5.1 Exemplos de FDC.....	75
IV.2.5.2 Exemplos de Medidas de Valor Médio.....	80
IV.2.6 Resumo da seção IV.2.....	83
IV.3 METODOLOGIA GENÉRICA PARA A ANÁLISE DE DESEMPENHO DO META-ALGORITMO LDS.....	84
IV.3.1 Multiplexador ATM para Tráfegos de Voz e Dados com Controle de Congestionamento.....	84
IV.3.2 Avaliação de Desempenho do Multiplexador.....	85
IV.3.3. Modelo de Operação do Multiplexador .....	86
IV.3.3.1 Processos de Chegada .....	86
IV.3.3.2 Serviço das Células .....	86
IV.3.3.3. Modelos de Filas do Multiplexador.....	87
IV.3.4. Definições Básicas.....	88
IV.3.4.1. Elementos da Matriz E .....	91
IV.3.4.2 Elementos da Matriz V.....	92
IV.3.4.3 Elementos da Matriz D.....	98
IV.3.5 Avaliação de Medidas de Desempenho.....	102
IV.3.5.1 Tamanho Médio do Ciclo .....	102
IV.3.5.2 Número médio de células .....	105
IV.3.5.3 Retardo Médio.....	113
IV.3.5.4. Probabilidade de Perda.....	115
IV.3.5.5 Vazão ( <i>throughput</i> ).....	116

IV.3.6 Exemplos Numéricos.....	117
IV.3.7 Resumo da seção IV.3.....	121
<b>CAPÍTULO V ALGORITMO PARA A RECUPERAÇÃO DE ERROS DE UMA TRANSMISSÃO MULTIPONTO CONFIÁVEL.....</b>	<b>122</b>
V.1 INTRODUÇÃO.....	122
V.2 TAXONOMIA PROPOSTA.....	123
V.2.1 Quem detecta a perda de um pacote?.....	123
V.2.2 Como se detecta a perda de um pacote?.....	126
V.2.3 Quem envia o NACK?.....	126
V.2.4 Como diminuir o tamanho do grupo de potenciais candidatos a enviar NACKs?.....	128
V.2.5 Para onde enviar o NACK?.....	128
V.2.6 Quem retransmite o pacote perdido e para onde?.....	129
V.3 CLASSIFICAÇÃO DE ALGUNS PROTOCOLOS.....	129
V.3.1 Métodos de Comparação dos protocolos.....	131
V.4 PROPOSIÇÃO DE ALGORITMO PARA A TRANSMISSÃO MULTIPONTO CONFIÁVEL BASEADO EM ÁRVORE DE RECUPERAÇÃO.....	132
V.4.1 Motivação.....	132
V.4.2 Modelo.....	133
V.4.3 Algoritmo Proposto.....	133
V.4.3.1 Determinação da probabilidade de enviar um NACK.....	138
V.4.4 Análise de Desempenho do Algoritmo.....	140
V.4.4.1 Topologia Estrela.....	140
V.4.4.2 Outras Topologias.....	151
V.5 RESUMO.....	152
<b>CAPÍTULO VI CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>154</b>
VI.1 INTRODUÇÃO.....	154
VI.2 CONCLUSÕES.....	154
VI.3 TRABALHOS FUTUROS.....	156
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>158</b>
<b>APÊNDICE A.....</b>	<b>173</b>

# Lista de Figuras

Figura II.1. Mecanismos de Administração de Banda Passante Não Hierarquizados.....	13
Figura II.2 Mecanismo de Atribuição de Banda Passante Hierarquizado. ....	14
Figura II.3. Taxonomia de algoritmos de escalonamento.....	15
Figura II.4 Atribuição de taxa e atribuição de taxa controlada .....	17
Figura II.5 Algoritmo HRR.....	25
Figura II.6 Caracterização da voz.....	31
Figura III.1. Multiplexador genérico que serve a ‘N’ tráfegos diferentes. ....	34
Figura III.2. Diagrama temporal da operação do LDS em um ciclo. ....	35
Figura III.3. Exemplo de uma Tabela LDS.....	35
Figura IV.1. Multiplexador ATM para tráfego de voz .....	46
Figura IV.2. Caracterização da voz. ....	48
Figura IV.3. Estrutura das células de voz (figura obtida de [SRI91]). ....	49
Figura IV.4 Algoritmo de descarte de células. ....	49
Figura IV.5. Diagrama dos pontos embutidos e cadeia Markov de tempo discreto definida nos pontos embutidos.....	50
Figura IV. 6. Recompensa acumulada no ciclo $\delta_i$ ( $i>0$ ) quando a memória do multiplexador não está saturada. ....	58
Figura IV.7 : Recompensa acumulada no ciclo $\delta_i$ ( $i>0$ ), quando a memória do multiplexador está saturada.....	59
Figura IV.8. Cadeia de $Z_r$ usada para modelar o tempo de residência no multiplexador.....	64
Figura IV.9. Probabilidades de transição da cadeia de Markov que modela o tempo de residência do par de células no multiplexador de voz .....	65
Figura IV.10. Matriz de probabilidades de transição $Z_0$ da cadeia de Markov que modela o comportamento do sistema durante um período de congestionamento. ....	67
Figura IV.11. Matriz de probabilidade de Transição $Z_L$ da cadeia de Markov que modela o número de células perdidas durante um período de congestionamento.....	70
Figura IV.12: Matriz de probabilidades de transição $Z_u$ da cadeia de Markov que modela o comportamento do sistema durante um período não congestionado. ....	72
Figura IV.13. Função de Distribuição Cumulativa do tempo de residência. ....	76
Figura IV.14. FDC do tempo de residência para diferentes valores do limiar de descarte. ....	77

Figura IV.15. Função de Distribuição cumulativa do comprimento do período de congestionamento. ....	78
Figura IV.16. Função de Distribuição Cumulativa do Comprimento de um Período Congestionado para diferentes limiares de descarte de células. ....	79
Figura IV.17. Função de Distribuição Cumulativa do Número de Células Descartadas.....	80
Figura IV.18. Número Médio de células no Multiplexador.....	80
Figura IV.19. Tempo Médio na Fila.....	81
Figura IV.20. Tamanho médio de um período congestionado.....	82
Figura IV.21: Número Médio de Células Descartadas .....	83
Figura IV.22. Multiplexador ATM para Tráfegos de Voz e Dados. ....	85
Figura IV.23 Pontos embutidos nos quais é definida a Cadeia de Markov $C$ .....	88
Figura IV.24. Pontos embutidos nos quais se definem as Cadeias de Markov $C$ e $E$ . ....	89
Figura IV.25. Pontos embutidos nos quais se definem as Cadeias de Markov $C$ e $V$ . ....	89
Figura IV.26. Pontos embutidos nos quais se definem as Cadeias de Markov $C$ e $D$ .....	90
Figura IV.27 Cadeias de Markov $C$ , $E$ , $V$ e $D$ , e vetores de probabilidade $\vec{\pi}_C$ , $\vec{\pi}_V$ e $\vec{\pi}_D$ .....	91
Figura IV.28. Pontos embutidos nos quais se definem as Cadeias de Markov $C$ , $V$ e $W$ . ....	94
Figura IV.29 Matriz de probabilidades de transição $\mathbf{W}$ .....	95
Figura IV.30 Pontos embutidos nos quais se definem as Cadeias de Markov $C$ , $D$ e $M$ .....	99
Figura IV.31 Matriz de probabilidades de transição $\mathbf{M}$ .....	100
Figura IV.32. Número de células de dados durante a fase de voz, quando não existe saturação do buffer de dados.....	107
Figura IV.33 Número de células de dados durante a fase de voz, quando o buffer de dados fica saturado.....	107
Figura IV.34. Número de pares de células de dados nos instantes em que termina a atenção de uma célula de dados versus tempo.....	109
Figura IV.35 Vazão para o tráfego de dados em função da carga de dados .....	117
Figura IV.37 Probabilidade de perda de células de dados versus a carga normalizada do canal .....	120
Figura V.1 Região de interesse para a proposição de algoritmos de recuperação de erros. ..	132
Figura V.2: Árvore de distribuição, árvore de recuperação e árvores de espera.....	134
Figura V.3: Topologia estrela.....	141

Figura V.4. Implosão Média ( $E[I]$ ) versus Tamanho do Grupo Multiponto (R) para diferentes valores de $\alpha$ .	145
Figura V.5. Implosão Média versus $\alpha$	146
Figura V.6. Latência média normalizada versus Tamanho do Grupo Multiponto (R) para diferentes valores de $\alpha$	147
Figura V.7. Latência Média Normalizada em função de $\alpha$	148
Figura V.8. Latência média versus tamanho do grupo para o algoritmo ACMBAR ( $\alpha=2.5$ ) e o algoritmo SRM (para diferentes valores de C2).	149
Figura V.9. Implosão versus tamanho do grupo para o algoritmo ACMBAR ( $\alpha=2.5$ ) e o algoritmo SRM (para diferentes valores de C2).	150
Figura V.10. Topologia cadeia	151
Figura V.11. Topologia em árvore balanceada de grau G, onde todos os nós pertencem ao grupo multiponto.	152

# Lista de Tabelas

Tabela II.1 Mecanismos de Escalonamento de Banda Passante. ....	19
Tabela II.2 Limites de retardo dados pela recomendação G.114 da ITU-T.....	30
Tabela III.1. Tabela LDS para o Algoritmo de Prioridade Simples [DRA93].....	37
Tabela III.2. Tabela LDS para algoritmo DTS.....	39
Tabela III.3. Tabela LDS para Algoritmo Proporcional.....	40
Tabela III.4. Tabela LDS para um Algoritmo Preditivo baseado na estatística das chegadas	41
Tabela III.5. Tabela LDS para um Algoritmo Preditivo baseado nas chegadas do ciclo anterior. ....	41
Tabela III.6. Tabela LDS para um Algoritmo que aproxima a operação de GPS.....	42
Tabela IV.1. Probabilidade de Perda de Células de Voz .....	119
Tabela V.1. Classificação de alguns protocolos de Transporte Multiponto Confiável. ....	130

# Capítulo I

## Introdução

O volume de informação que é transmitido pelas redes atuais e a natureza das novas aplicações distribuídas, tais como vídeo-conferência, vídeo-sob-demanda, tele-medicina, WWW, telefonia; impõem uma demanda crescente da banda passante e exigem a necessidade de dispor de redes de banda larga capazes de integrar e transportar informações, de diferentes tipos, de forma eficiente.

Devido principalmente a restrições tecnológicas e econômicas, as redes tradicionais foram desenvolvidas independentemente uma das outras. Desta maneira, foram projetadas e implementadas redes que transportavam especificamente tráfegos de voz, dados textuais ou vídeo. Uma desvantagem deste tipo de tecnologia vem do fato que essa rede somente é capaz de transportar, de forma eficiente, o tráfego para o qual foi projetada, apresentando grande ineficiência na utilização de seus recursos, se por ela são transportados outros tipos de tráfegos.

Com o desenvolvimento tecnológico das comunicações ópticas, foi possível gerar a chamada Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL) [CIT90A, CIT90B] capaz de transportar de maneira unificada serviços de voz, dados e vídeo. A ITU-T (International Telecommunications Union-Telecommunication) adotou o Modo de Transferência Assíncrono (Asynchronous Transfer Mode (ATM)) [ONV94,RAMA98], como técnica de multiplexação e comutação para transportar, de forma simultânea, diferentes tipos de tráfegos com garantia de QoS.

Na mesma direção caminharam as redes baseadas no protocolo IP, apresentando diversas soluções para o encaminhamento de tráfegos de mídias diferentes, como exemplificado pelos Serviços Integrados IP, Serviços Diferenciados IP, MPLS (Multiprotocol Label Switching) [FER99].

Tanto para as redes ATM como nas baseadas no protocolo IP, têm sido definidas várias categorias de serviço, as quais cobrem o espectro das aplicações atuais e futuras que usem esta tecnologia. Essas categorias oferecem requisitos de qualidade de serviço (QoS) distintos, e são diferenciadas em termos de tolerância à perdas, exigência de sincronização temporal entre transmissores e receptores, etc.

Por outro lado, um serviço tem uma certa qualidade se reúne, de uma forma consistente, o mesmo nível para um conjunto de parâmetros possíveis de serem medidos. Os serviços também diferem pela forma que a rede atribui a banda passante e aplica as diferentes funções de administração de tráfego [TM499].

Em uma rede que dá suporte a comunicações multimídia é fundamental garantir a QoS necessária às diferentes aplicações. A QoS de uma aplicação faz referência à especificação de um conjunto de medidas de desempenho e aos valores (ou conjunto de valores) que cada uma dessas medidas deveria ter, para que uma aplicação opere satisfatoriamente [TM499]. Por exemplo, podem ser especificadas as restrições de QoS relacionadas com o retardo na transferência da informação, a disponibilidade de banda passante de um canal ou a probabilidade de perda de células.

Para poder usar de melhor forma os recursos, as redes devem ser capazes de compartilhar a banda passante entre os diferentes usuários, garantindo que o tráfego de um usuário não afete a QoS dos outros usuários que usam a rede.

Para uma rede, não é uma tarefa trivial garantir a QoS de uma conexão. Isto deve-se, em parte, ao fato da demanda de recursos feita por uma determinada conexão poder variar no tempo. Outra razão que torna difícil prever se uma rede pode garantir a QoS de uma conexão, é que os recursos da rede (processadores, canais de comunicações e, outros) devem ser compartilhados com outras conexões. Para resolver esta dificuldade é normal proceder-se em duas etapas. Em primeiro lugar, a QoS da conexão é traduzida a uma determinada QoS que deveria ser satisfeita por cada um dos canais que fazem parte desta conexão (isto é, se cada um dos canais cumprir com sua QoS, então o caminho completo cumpre com a QoS requerida pela aplicação). Em segundo lugar, uma vez traduzida a QoS por canal, é feito um estudo probabilístico, no qual são consideradas as características do canal, assim como

também as características do tráfego que o usa, para se determinar se um determinado canal cumpre a QoS solicitada.

## **I.1 Objetivos e Contribuições da Tese**

Um dos problemas relacionados com a avaliação de redes que oferecem garantias de QoS, é a falta de modelos simples que permitam obter uma primeira aproximação de parâmetros de QoS para os distintos tráfegos que transporta a rede.

O objetivo global desta tese é melhorar algumas das técnicas utilizadas na transmissão de diferentes tipos de tráfegos, através de uma rede de computadores, com o intuito de fornecer uma melhor QoS. Para isto, são propostos algoritmos, modelos, metodologias de análise de desempenho, e a análise de desempenho dos algoritmos propostos, calculando as medidas de desempenho de interesse de cada caso particular.

As análises de desempenho foram realizadas em forma matemática. Foram obtidas expressões fechadas que, apropriadamente parametrizadas, permitiram resolver os modelos propostos e obter as medidas de desempenho de interesse, para cada problema.

O primeiro tópico analisado refere-se ao controle de congestionamento do tráfego de voz transportado em uma rede ATM. A análise da transmissão de voz nas redes ATM é um objetivo muito importante, dado que a voz, por ser o serviço de telecomunicação preferido para a comunicação interpessoal, gerará um dos tráfegos mais presentes nestas redes [NEW99]. O congestionamento é causado pelas variações estatísticas na chegadas de rajadas de tráfego de voz aos pontos de contenção (filas). O controle de congestionamento é realizado de forma que possam ser descartadas células de voz de uma maneira “justa” e, além disso, que a QoS das conexões seja garantida.

O objetivo específico nessa parte da tese foi avaliar medidas transientes para um multiplexador de voz durante períodos de congestionamento, tais como a função de distribuição dos períodos de congestionamento, a função de distribuição do número de células descartadas e outras mais. Estas medidas são importantes, pois permitem caracterizar melhor

o comportamento de um multiplexador de voz durante os períodos de congestionamento. É interessante fazer notar que estas medidas permitem projetar de melhor forma um multiplexador de voz. Projetar um multiplexador ATM significa encontrar o valor adequado para cada um de seus parâmetros característicos de forma a assegurar a QoS negociada em cada conexão.

Embora os cálculos desenvolvidos tomem por base uma rede ATM, eles podem ser facilmente estendidos a qualquer outro tipo de rede. Por exemplo, em [BAR95] a análise foi feita para a rede IACS [HEF86].

Como já mencionado, uma rede pode transportar simultaneamente múltiplos tráfegos com requisitos de QoS diferentes. O conflito referente a necessidade de otimizar a banda passante enquanto assegura-se diferentes requisitos de QoS, pode ser resolvido usando uma combinação de técnicas de administração de tráfego. Uma das componentes dessas técnicas é a técnica de Escalonamento e Enfileiramento de tráfego. O tráfego que passa sucessivamente através de funções de monitoramento, entra na rede e é multiplexado em diferentes pontos dela. Para poder atingir um ganho na multiplexação estatística (tráfego de taxa variável), o tráfego deve ser enfileirado antes de ser transmitido nos canais intermediários. As filas são servidas de acordo com um mecanismo de escalonamento especificamente projetado para garantir a QoS dos diferentes tipos de conexões.

Nesta tese, foi proposta uma metodologia de emulação e análise de algoritmos de escalonamento, que é utilizada para avaliar diferentes algoritmos de escalonamento.

Essa metodologia inclui as seguintes três etapas:

- 1) A proposição de um meta-algoritmo, denominado LDS, que permite emular a maioria dos algoritmos de escalonamento publicados na literatura, através do uso de Tabelas especiais, denominadas Tabelas LDS (Capítulo III).
- 2) A análise de desempenho do meta-algoritmo LDS. Esta análise é feita de forma geral (paramétrica) para dois tipos de tráfegos: voz e dados textuais. Os parâmetros instanciados em cada caso dependem do algoritmo de escalonamento que deseja-se avaliar (Seção IV.3).

- 3) A proposição de algoritmos de escalonamento, que distribuem dinamicamente banda passante entre os diferentes tipos de tráfego que compartilham um mesmo canal em uma rede, os quais são avaliados usando a análise de desempenho desenvolvida no ponto 2).

Uma das áreas mais interessantes, na presente evolução das telecomunicações, é o desenvolvimento de aplicações multiponto, devido ao grande impacto que elas têm e terão para a sociedade. Exemplos iniciais destas aplicações são as teleconferências, “quadros brancos” compartilhados, distribuição de filmes, jornais interativos [MIL98].

O desenvolvimento de serviços multiponto está ainda em um estágio inicial, sendo necessário melhorar as soluções existentes e também encontrar novas soluções para muitos problemas básicos, sobretudo os que tem relação com QoS. Um desses problemas é a confiabilidade requerida por algumas aplicações na transmissão da informação, de uma fonte para múltiplos destinos, que requer que todos os receptores recebam sem erros todos os pacotes transmitidos pela fonte.

O objetivo nessa parte da tese é propor e avaliar o desempenho de um novo algoritmo para a recuperação de erros em uma transmissão multiponto confiável, realizada em uma rede comutada por pacotes. No processo de recuperação de erros, o algoritmo proposto diminui o tempo de recuperação de um erro (latência) e o número de pacotes que chegam simultaneamente a uma determinada fonte pedindo retransmissão (implosão). Melhorar essas duas medidas de desempenho é de grande importância, pois ajuda a atingir os requisitos de QoS que os usuários solicitam de uma transmissão multiponto confiável<sup>1</sup>.

Os três problemas enfocados nesta tese estão intimamente relacionados na concepção de comutadores para redes de computadores (ATM, redes TCP/IP).

No escalonamento de células ou pacotes, muitas vezes opções de descarte devem ser tomadas para o controle de congestionamento (seção II.2), baseadas na confiabilidade exigida pela

---

<sup>1</sup> Em relação a esse capítulo do trabalho, gostaria de agradecer aos grupos RODEO e MISTRAL do *Institute National de Informatique e Automatique* (INRIA, França) onde, durante uma estadia de pesquisa, foi desenvolvida uma parte deste trabalho.

aplicação e na redundância de informação presentes em um sinal (Seção II.4). Aplicações que exigem confiabilidade devem minimizar o tráfego de controle (reconhecimentos, etc.) e de dados (retransmissões) ao máximo, uma vez que essas informações não podem sofrer descartes e muitas vezes devem ter prioridade no escalonamento (ver Seção V). Os três tópicos, no entanto, podem ser considerados individualmente, pois algumas das soluções aplicam-se em redes roteadas e não apenas comutadas e, algumas delas até a sistemas de comunicação que não se restringem a redes, como por exemplo, a comunicação através de trocas de mensagens em um sistema operacional.

## **I. 2 Organização da Tese**

Esta tese está organizada da seguinte forma.

No Capítulo II, são apresentados os conceitos que serão utilizados nos capítulos posteriores desta tese. Na seção II.1 são apresentados conceitos sobre QoS, em especial em redes ATM. Posteriormente, na seção II.2, é feita uma taxonomia sobre algoritmos de atribuição ou de escalonamento de banda passante, que será a base para a criação das diferentes tabelas com as quais pode ser implementado e avaliado o meta-algoritmo para administração de banda passante que é proposto no Capítulo III. Na seção II.3, é apresentado o estado da arte de algoritmos que servem para o controle de congestionamento. Os algoritmos para o controle de congestionamento de tráfego de voz são descritos em forma mais detalhada. Posteriormente, na seção II.4 é dada uma explicação sobre o método usado para a transmissão do tráfego de voz sobre redes ATM. Os tópicos descritos nas seções II.3 e II.4 serão usados na análise de desempenho de multiplexadores ATM realizada no capítulo IV.

No Capítulo III, é apresentado o Meta-Algoritmo LDS (Load Dependent Scheduling) para Escalonamento de Banda Passante. Usando LDS, podem ser emulados mecanismos de escalonamento distintos, publicados na literatura. Esta emulação é simples e baseia-se no uso de tabelas especiais, que serão descritas no capítulo. Uma proposição inicial do algoritmo foi feita em [BAR96a]. A proposição final de LDS e sua potencialidade foi publicada em [BAR00a].

No Capítulo IV é apresentada uma análise de desempenho de multiplexadores ATM. A seção IV.2 contém uma análise transiente de um multiplexador ATM para tráfego de voz [BAR98]. O objetivo aqui foi avaliar medidas transientes de desempenho, que permitirão caracterizar o comportamento de um multiplexador de voz durante congestionamento. Este conjunto de medidas inclui, a função de distribuição do tamanho do período de congestionamento; a função de distribuição do período de não congestionamento; a função de distribuição do número de células descartadas durante o congestionamento. Os valores médios de algumas delas também foram avaliadas. Todas essas medidas são importantes dado que a distorção da voz é altamente relacionada com o comportamento do multiplexador durante os períodos de congestionamento. Outro importante requisito na transmissão das células de voz é o retardo da transferência. Para poder manter a interatividade na comunicação, este retardo deve ser pequeno (da ordem de uns poucos milisegundos (mseg)). Para caracterizar este requisito, foi avaliada a função de distribuição cumulativa do tempo de residência das células no multiplexador [BAR95a,BAR95b,BAR96b].

Na Seção IV.3 é apresentada uma metodologia de análise de desempenho do meta-algoritmo LDS, a qual permite avaliar o desempenho de diversos mecanismos de escalonamento. Tal análise foi particularizada para um multiplexador ATM, que usa o meta-algoritmo LDS para atribuir a banda passante de um canal de saída entre os tráfegos de voz e de dados. Entre as medidas avaliadas estão o número médio de células de voz, o número médio de células de dados, a probabilidade de perda de células de voz e a probabilidade de perda de células de dados.

Os resultados numéricos permitem comparar os diferentes algoritmos de escalonamento e analisar, sob diferentes condições de operação, a entrega da QoS requerida para cada um dos tipos de tráfego.

No Capítulo V é proposto um algoritmo para a transmissão multiponto confiável em uma rede comutada por pacote (particularizada para uma rede IP), denominado ACMBAR. A avaliação de desempenho do algoritmo é então realizada. Mesmo em uma rede com ausência de erros é possível que exista perda de pacotes devido a transbordo de buffers, devido a variações estatísticas. Este tipo de problema é bem compreendido no caso de transmissão ponto-a-ponto, mas para o caso da transmissão multiponto existem muitos fenômenos produzidos pela necessidade de recuperação de erros. Para recuperar erros em grupos de tamanho grande, é

necessário que os algoritmos de recuperação sejam eficientes. Em primeiro lugar, na Seção V.2, apresenta-se uma conceituação do problema da transmissão multiponto confiável [BAR99]. Nesta conceituação são identificados os diferentes requisitos necessários para a resolução do problema. Também são classificados alguns importantes protocolos propostos na literatura para a recuperação de erros. Em segundo lugar, na Seção V.4, é proposto um novo algoritmo para a recuperação de pacotes perdidos em uma transmissão multiponto confiável [BAR00b]. Para diminuir a implosão de reconhecimentos, a recuperação é realizada somente pelos membros que são vizinhos à falha. Estes membros delimitam o que se tem denominado de árvore de recuperação. Para poder atingir simultaneamente uma baixa implosão e uma baixa latência, é proposto um modo particular de efetuar a retransmissão do pacote perdido. Pela proposta, cada um dos membros pertencentes a árvore de recuperação decide pedir ou não a retransmissão do pacote perdido. O desempenho do algoritmo foi matematicamente avaliado para algumas situações típicas, descritas na Seção V.4. Os resultados obtidos até o presente mostram que o algoritmo atinge simultaneamente uma baixa implosão e uma baixa latência no processo de recuperação de pacotes perdidos, para qualquer tamanho de grupo multiponto. Isto significa que o algoritmo possui uma alta escalabilidade [BAR01].

Finalmente, no capítulo VI são apresentadas as conclusões obtidas nesta tese e as sugestões para trabalhos futuros.

No Apêndice A é apresentada uma explicação da forma de obter o tamanho médio de um período ocupado, que foi utilizado para a análise de desempenho feita na seção IV.2.

# Capítulo II

## Conceitos Utilizados

### II.1 QoS em Redes ATM

As antigas redes de comunicações usavam uma infra-estrutura separada para cada tipo de aplicação. Atualmente, a tecnologia ATM permite a implementação de uma única rede que pode entregar a QoS necessária às diferentes aplicações existentes, assim como novas aplicações que venham a aparecer.

O intuito desta seção não é dar uma visão exaustiva sobre a tecnologia ATM, pois existe muita literatura especializada sobre o tema [PRY95,SOA95,RAM98], apenas destacar algumas características principais que será utilizada no decorrer dos próximos capítulos.

O modo de transferência assíncrono está baseado na segmentação do fluxo de informação em pacotes de tamanho fixo igual a 53 bytes, denominados células. O pequeno tamanho das células, aliado às altas velocidades de transmissão utilizadas, proporciona a flexibilidade e a eficiência que são necessárias para a integração dos diferentes serviços. As características que definem um determinado serviço são negociadas entre o usuário e a rede, antes da transferência da informação, durante a fase de estabelecimento da conexão.

Uma rede ATM pode ser entendida como um conjunto de multiplexadores e comutadores ATM, conectados mediante canais que permitem a transmissão de células de uma fonte para um ou mais destinos. ATM é uma tecnologia de redes orientada à conexão na qual cada fonte transmite dados sobre circuitos virtuais permanentes (VPs) ou comutados (VCs) [RAM98].

A camada ATM trabalha salto-a-salto (hop-to-hop) e é a camada responsável pela comutação das células. Isto é, cada nó de comutação dirige a célula que entra a seu canal de saída correspondente, até que a célula alcança seu destino. Para armazenar aquelas células que não podem ser imediatamente atendidas (transmitidas a frente) quando chegam ao comutador, este dispositivo deve possuir uma memória (buffer).

Ao comutador ATM chegam células provenientes de várias aplicações, cada uma delas com diferentes características de tráfego e, requisitos de QoS distintos. Para que as aplicações recebam o serviço que elas precisam, os nós intermediários devem fazer cumprir esses requisitos. Com esse propósito, a camada ATM pode possuir seis categorias de serviço: Taxa de Bits Constante (Constant Bit Rate (CBR)), Taxa Variável de Bits de Tempo Real (Real-Time Variable Bit Rate (rt-VBR)), Taxa Variável de Bits de Tempo Não Real (Non-Real-Time Variable Bit Rate (nrt-VBR)), Taxa de Bits Não Especificada (Unspecified Bit Rate (UBR)), Taxa de Bits Disponível (Available Bit Rate (ABR)) [AtmForum] e Taxa de Frame Garantido (Guaranteed Frame Rate (GFR)) [TM499]. Estas categorias de serviço relacionam as características de tráfego e requerimentos de QoS com o comportamento da rede.

### **II.1.1 Gerenciamento do Tráfego em Redes ATM**

O gerenciamento de rede permite utilizar de maneira eficiente os recursos da rede mediante diversos algoritmos e controles, usados para prevenir o congestionamento nos nós da rede. O Controle de Admissão de Conexões (Connection Admissão Control (CAC)) é o algoritmo encarregado de aceitar ou rejeitar uma nova conexão, de maneira que as conexões já existentes não sejam afetadas. Policiamento é aplicado a cada célula que entra na rede, para conferir se cumpre ou não o contrato de tráfego que, entre outras coisas, caracteriza o fluxo de células da conexão, especificado através de descritores de tráfego. A célula “conforme” entrará na rede e a “não conforme” será descartada ou marcada com prioridade baixa.

Os descritores de tráfego utilizados são:

- PCR (Peak Cell Rate): Taxa máxima de chegada de células.
- SCR (Sustained Cell Rate): Taxa média de chegada de células.
- MBS (Maximum Burst Size): Número máximo de células que podem ser transmitidas na taxa PCR.
- MCR (Minimum Cell Rate) : Taxa mínima de chegada de células.
- CDVT (Cell Delay Variation Tolerance): margem na medida de CDV .

Além de utilizar eficientemente os recursos da rede, o gerenciamento de tráfego ATM deve garantir a qualidade do serviço contratado por cada conexão. Para garantir a qualidade dos

diferentes tipos de tráfego, o ATM Forum define seis categorias de serviço, cada um deles com diferentes restrições nos parâmetros de qualidade de serviço.

## II.1.2 Qualidade de Serviço (QoS)

A QoS é definida para cada conexão em uma base fim-a-fim, e é medida por um conjunto de parâmetros que caracterizam o desempenho de uma conexão na camada ATM. Estes parâmetros quantificam o desempenho da rede fim-a-fim [ATM499]. A recomendação I.356 do ITU-T, os referencia como parâmetros de desempenho da rede [I356].

### II.1.2.1 Parâmetros de QoS

Os parâmetros de QoS podem ser negociados ou não negociados.

Os parâmetros básicos de QoS, e que são negociados ao iniciar uma conexão são:

- Proporção de Células Perdidas (Cell Loss Ratio (CLR)): Percentual de células perdidas devido a erros o congestionamento. É dada pela razão entre células perdidas e células totais transmitidas.

$$CLR = \frac{\text{células perdidas}}{\text{células totais transmitidas}}$$

Cada célula ATM possui um bit de prioridade de perda da célula (Cell Loss Priority (CLP)) no cabeçalho. Durante os períodos de congestionamento, a rede descarta primeiro as células que tem o CLP = 1. O CLR pode ser especificado de maneira distinta para as células que possuem CLP=1 ou CLP=0.

- Retardo na Transferência das Células (Cell Transfer Delay (CTD)): Tempo transcorrido entre o ponto de entrada à rede e o tempo de chegada ao destino. O CTD através da rede é a acumulação dos retardos de transferência em cada um dos nós do caminho. O CTD inclui:
  - retardos internos: é o tempo gasto devido ao enfileiramento em um ou mais pontos de contenção (comutadores), ao processamento e, à transmissão nos canais internos. A distribuição do retardo nas filas varia como uma função da carga e do tipo de algoritmo de escalonamento usado.

- retardos externos às filas e retransmissão: é o tempo requerido para enfileirar e transmitir os bits no canal de saída. A distribuição do retardo nas filas varia como uma função da carga e do tipo de algoritmo de escalonamento usado.
  - retardos de processamento: é o tempo decorrido no processamento das células. Por exemplo: analisar o cabeçalho.
  - retardo de propagação: é o tempo requerido para transmitir os bits pelo meio físico.
- 
- Variação do Retardo das Células (Cell Delay Variation (CDV)): É uma componente do CTD, induzida ao enfileirar e escalonar células, o que introduz variações nas características de tráfego. Quando as células de duas ou mais conexões ATM são multiplexadas, as células de uma determinada conexão podem ser retardadas, enquanto as células das outras conexões estão sendo inseridas no canal de saída. Em consequência, efeitos aleatórios podem afetar o tempo de chegada entre duas células consecutivas pertencentes a uma conexão. Isto é denominado variação de retardo pico-a-pico (peak-to-peak cell delay variation (P2PCDV)).

## **II.2 Taxonomia de Algoritmos de Escalonamento de Banda Passante**

Como foi descrito na Seção II.1, uma rede ATM entrega suporte para uma grande variedade de serviços com diferentes requerimentos de QoS. Estes serviços são transportados através da rede usando os mesmos canais e comutadores. Portanto, compartilham recursos da rede tais como: banda passante, espaço de buffers e outros. Além disso, estes serviços utilizam os recursos de forma simultânea, produzindo-se contenção neles. O ponto onde ocorre esta contenção é denominado ponto de contenção. Um ponto de contenção, como é o caso, por exemplo, de um multiplexador ou de um comutador, pode ser implementado com uma ou mais filas. Estas filas podem representar diferentes tipos de serviço ou tráfego. O termo escalonamento refere-se ao mecanismo que determina que fila terá a oportunidade de transmitir suas células. É necessário, então, contar com uma função que permita extrair as células das filas para poder transmiti-las (ou servi-las) apropriadamente, de forma que cumpram com os requisitos de QoS de cada uma das conexões. Um servidor é responsável

por esta atribuição da banda passante às filas (ver Figura II.1). A função implementada neste servidor denomina-se algoritmo de atribuição de banda passante ou algoritmo de escalonamento de banda passante.

As filas em um ponto de contenção podem ser divididas em conjuntos lógicos, cada um dos quais é atendido por um servidor. Dependendo de como as filas e os servidores sejam organizadas, os esquemas de escalonamento podem ser divididos em: mecanismos de atribuição de banda passante não hierarquizados (ou de um nível) e mecanismos hierarquizados.

Em um esquema de atribuição de banda passante não hierarquizado (ver Figura II.1), o ponto de contenção usa um servidor para atender todas as filas, e esse servidor usa uma função simples de escalonamento entre elas.

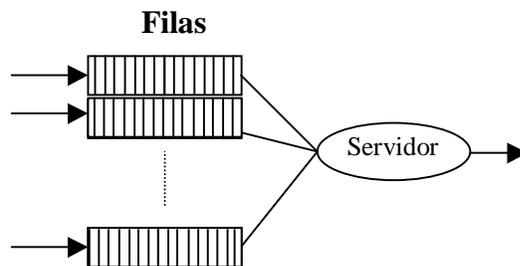


Figura II.1. Mecanismos de Administração de Banda Passante Não Hierarquizados

No mecanismo de administração de banda passante hierarquizado existem vários servidores. Um servidor trabalha com um conjunto de filas. Por sua vez estes servidores são atendidos por servidores de maior hierarquia, e assim sucessivamente. Cada um dos servidores, em um sistema hierarquizado, pode implementar um algoritmo de escalonamento diferente para alcançar a partição de banda passante especificada (ver Figura II.2).

O escalonamento hierárquico [BEN97] é útil para dividir a banda passante de um canal entre diferentes conjuntos de filas. Por exemplo, filas que pertencem a um determinado usuário, a um determinado serviço, etc. Com o esquema hierárquico, a banda passante que sobra de uma conexão é redistribuída primeiro entre as conexões do mesmo nível, e depois entre conexões dos outros níveis. Este objetivo não pode ser atingido com os mecanismos de escalonamento simples, já que a banda passante que não é utilizada, é redistribuída entre todas as conexões atendidas pelo servidor e não somente para um conjunto delas

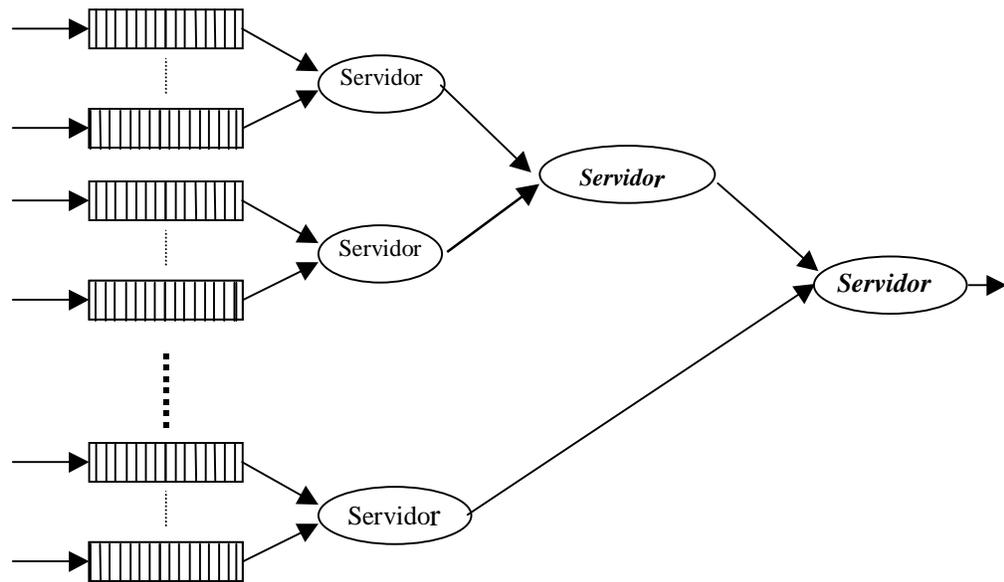


Figura II.2 Mecanismo de Atribuição de Banda Passante Hierarquizado.

A figura II.3 mostra a taxonomia de algoritmos de atribuição de banda passante que será apresentada a seguir.

Os algoritmos de atribuição de banda passante, que podem estar implementados em um servidor, podem ser classificados em dois grandes tipos:

- ◆ Algoritmos de Atribuição (ou de Escalonamento) de Banda Passante Baseados em Prioridade.
- ◆ Algoritmos de Escalonamento de Banda Passante Baseados em Acesso Justo.

A seguir são explicados em detalhes cada um destes tipos de algoritmos

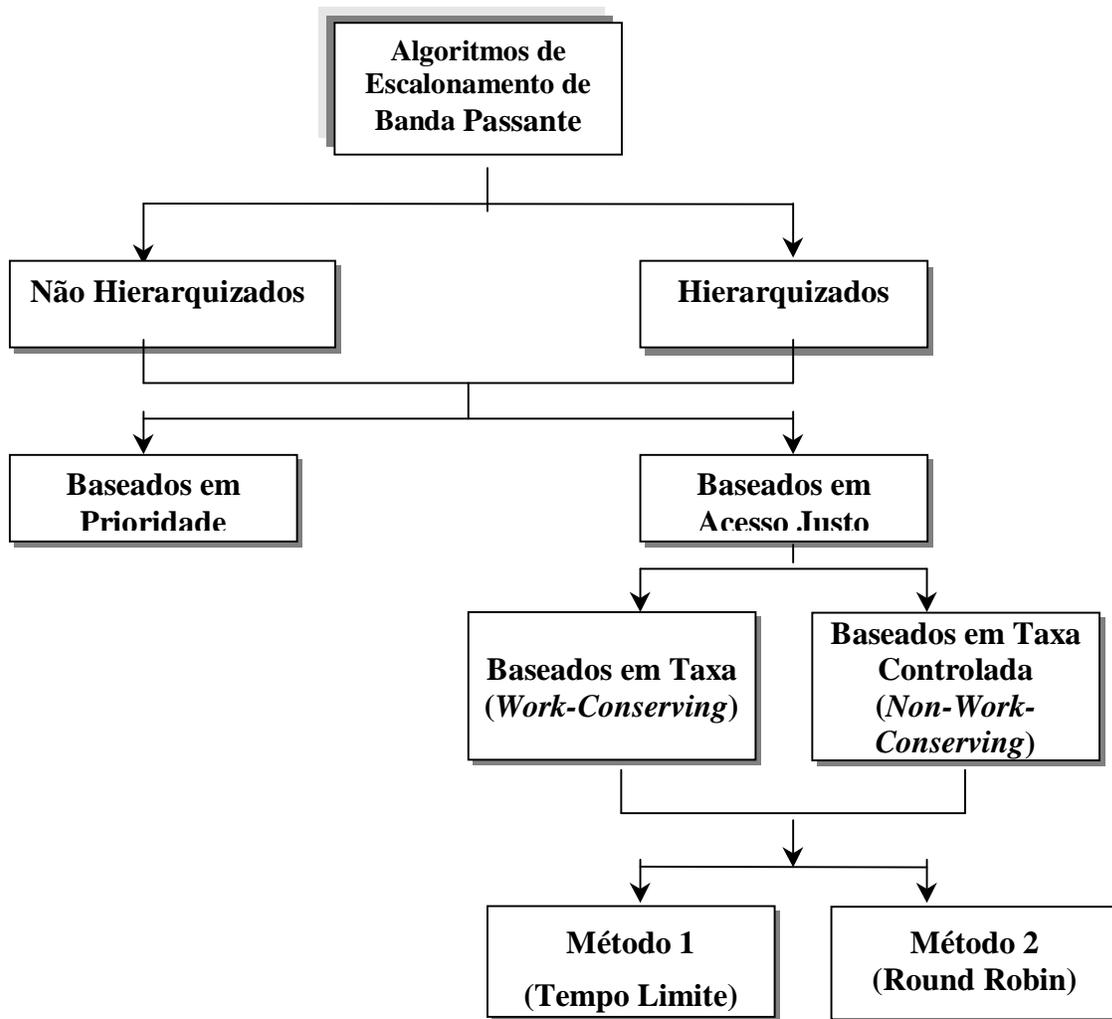


Figura II.3. Taxonomia de algoritmos de escalonamento.

### II.2.1 Algoritmos de Atribuição de Banda Passante Baseados em Prioridade

Neste tipo de mecanismo, o servidor atribui prioridade a cada uma das filas e as serve na ordem de prioridade. Uma fila de baixa prioridade é servida somente quando não há células esperando em filas de prioridade mais alta. Isto é, as células com prioridade mais alta são servidas primeiro ainda que, as células de baixa prioridade tenham chegado primeiro. A oportunidade de transmitir da fila de baixa prioridade, depende da carga de tráfego das células de prioridade mais alta, o que pode variar dinamicamente. Portanto, é difícil manter múltiplos serviços com QoS garantidas para cada uma das filas em forma separada. Por exemplo, dado

que os tráfegos CBR e rt-VBR apresentam restrições de retardo estritas, a estes geralmente se atribuem maior prioridade.

O escalonamento por prioridade pode ser aplicado a grupos de filas, como também a filas agrupadas por Canal Virtual (Virtual Channel (VC)). No caso de filas agrupadas por VC, todas as filas que pertencem a uma categoria de serviço particular, pode ser vista como um nível de prioridade. Se ao serviço CBR se atribui a mais alta prioridade e, ao serviço UBR a mais baixa, isto significa que o serviço CBR obterá um menor retardo e um menor percentual de células perdidas. Portanto, com um esquema de prioridades é difícil reunir os requisitos de qualidade de serviço para as filas com prioridade mais baixa.

Exemplos deste tipo de algoritmos são Algoritmo de Prioridade Simples (Simple Priority Algorithm (SP)) [DRA93], Algoritmo de Prioridades Baseados em Limiar (Threshold Based Priority Algorithm (TBP)) [LEE93].

Quando múltiplas classes de serviço compartilham a mesma fila, é possível ordenar a fila em uma forma não-FIFO, ordenando as células da mesma classe em forma FIFO, ou seja, as células dentro da mesma classe são servidas em forma FIFO. Este esquema é complexo de implementar mas entrega prioridades entre as classes de serviço que compartilham uma mesma fila física.

O esquema de escalonamento baseado em prioridades é um mecanismo simples de implementar, mas eficiente para um pequeno número de filas. Tem a desvantagem de que os requisitos de QoS para as filas com prioridades menores são difíceis de cumprir.

## **II.2.2 Mecanismo de Escalonamento de Banda Passante Baseado em Acesso Justo (*Fair-Share*)**

Uma alternativa ao esquema de escalonamento com prioridades é o escalonamento baseado em acesso justo, no qual cada fila tem garantia de obter a banda passante de acordo com um certo peso. O servidor divide a banda passante entre as filas baseado no peso que cada fila possui. Este tipo de escalonamento introduz isolamento entre as filas de um ponto de contenção, minimizando a interação entre o tráfego das diferentes filas. Este mecanismo, tenta distribuir a banda passante entre as diferentes filas em forma justa. É muito útil para serviços ABR e UBR, onde a banda passante é atribuída sobre a base da disponibilidade. Seria possível atribuir algum peso ao tráfego UBR, de forma tal que compartilhe a capacidade do canal.

Note que a atribuição justa é possível somente entre as filas e não entre as conexões que compartilham a mesma fila (se o tráfego fosse assim dividido). Por exemplo, se conexões de cada categoria de serviço fossem agrupadas em uma fila, então a atribuição justa seria possível entre categorias de serviço, mas não dentro de uma mesma categoria de serviço.

Os algoritmos de escalonamento baseados em acesso justo garantem uma certa taxa mínima a ser atribuída entre as filas. Estes algoritmos são conhecidos como algoritmos “baseados na taxa”, que podem ser classificados em duas categorias: 1) algoritmos de escalonamento baseados na atribuição de taxa e, 2) algoritmos de escalonamento baseados na atribuição de taxa controlada.

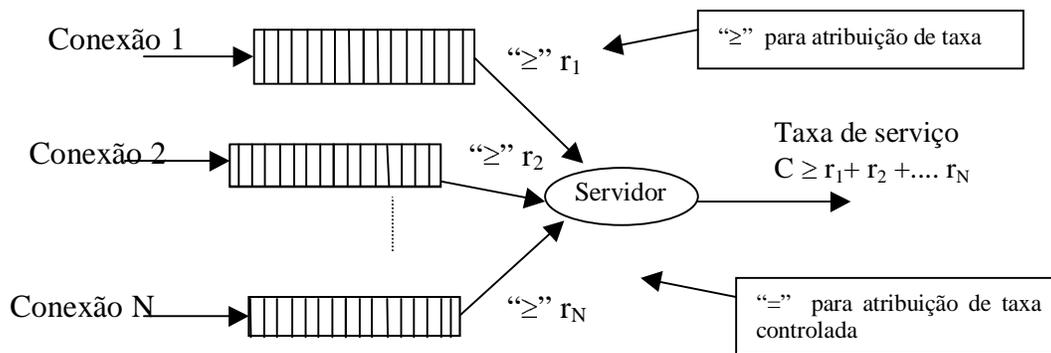


Figura II.4 Atribuição de taxa e atribuição de taxa controlada

No primeiro caso, uma fila pode ser servida a uma taxa maior que a taxa mínima garantida, sempre que as garantias oferecidas a outros serviços não sejam afetadas. Estes esquemas são denominados também como esquemas de trabalho conservativo (*work-conserving*). Um servidor com este tipo de esquema de trabalho nunca está ocioso enquanto tiver células para enviar em alguma das filas.

No segundo caso, na disciplina de serviço de atribuição de taxa controlada, o servidor não atende, sob nenhuma circunstância, a uma taxa maior que a estipulada. Estes mecanismos são classificados como de trabalho não conservativo (*non-work-conserving*).

Um servidor ideal baseado em acesso justo é aquele que usa o método de Compartilhar Processador (Processor Sharing (PS)) entre as conexões (ver figura II.4). Nesse caso, o processador divide igualmente a capacidade do canal entre as conexões com uma granularidade muito fina. Quando as conexões têm pesos diferentes e, a capacidade do canal

não é compartilhada igualmente, mas em proporção ao peso atribuído, o PS é denominado PS generalizado (Generalized Processor Sharing (GPS)). Na prática, é muito difícil implementar PS ou GPS, pois esses esquemas supõem que o tráfego é infinitamente divisível e que todas as conexões com filas ocupadas podem ser servidas em forma simultânea. Na realidade, uma célula somente pode ser servida por completo. Portanto, os esquemas baseados em acesso justo podem ser vistos como uma implementação prática de PS ou GPS.

Na literatura tem sido propostos vários esquemas para tentar atingir um acesso justo ao canal, e que basicamente podem ser implementados por dois métodos diferentes:

**Método 1:** é aquele que atribui a cada célula de uma conexão um tempo limite (também denominados etiquetas de tempo (*time stamps*)), ou um mecanismo de tempo virtual de término (*finishing time*) antes do qual uma célula deve ser atendida. Posteriormente, serve as células na ordem crescente do tempo limite. Se há duas ou mais células que tem o mesmo tempo limite, estas são reordenadas aleatoriamente.

A principal diferença entre os distintos algoritmos que utilizam este método é a maneira como é estimado este tempo limite.

**Método 2:** neste método o servidor atende primeiro a fila A, depois a fila B, e assim sucessivamente até atender todas as filas. Nesse momento termina um ciclo e o servidor executa o ciclo seguinte, no qual as filas são atendidas na mesma ordem do ciclo anterior. Esta forma é conhecida como Round Robin (RR). Quando há N conexões, uma disciplina RR atribui a cada conexão uma banda passante equivalente a  $\frac{(\text{capacidade do canal})}{N}$ .

### II.2.3 Descrição dos Algoritmos de Atribuição de Banda Passante

A Tabela II.1 apresenta uma visão dos algoritmos que serão descritos nesta seção. Os algoritmos do tipo *non-work-conserving* apenas são mencionados, por não ser de interesse para este trabalho.

No primeiro bloco da Tabela II.1 apresentam-se os algoritmos de escalonamento baseados em prioridade, e no segundo bloco estão os algoritmos de escalonamento baseados em acesso justo.

O segundo bloco é dividido em dois sub-blocos: algoritmos de trabalho conservativo, e algoritmos que não conservam o trabalho. Também, são diferenciados os algoritmos

implementados utilizando o Método 1 (Tempo Limite) e os que utilizam o Método 2 (Round Robin).

Mecanismos de Escalonamento Baseados em Prioridade	<ul style="list-style-type: none"> <li>• <i>Static Priority (SP)</i></li> <li>• <i>Threshold Based Priority (TBP)</i></li> </ul>	
Mecanismos de Escalonamento Baseados em Acesso Justo	<b>Trabalho Conservativo</b>	
	<i>Método 1 (Tempo Limite)</i>	<i>Método 2 (RR)</i>
	<ul style="list-style-type: none"> <li>• <i>Fair Queueing (FQ)</i></li> <li>• <i>Weighted Fair Queueing (PGPS)</i></li> <li>• <i>Self-Clocked Fair Queueing (SCFQ)</i></li> <li>• <i>Virtual Clock (VC)</i></li> <li>• <i>Worst-Case Fair-Weighted Fair Queueing (WF<sup>2</sup>Q)</i></li> <li>• <i>Delay-EDD</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>Weighted Round Robin (WRR)</i></li> <li>• <i>Deficit Round Robin (DRR)</i></li> <li>• <i>Dynamic Time Slice (DTS)</i></li> </ul>
	<b>Trabalho Não Conservativo</b>	
	<i>Método 1 (Tempo Limite)</i>	<i>Método 2 (RR)</i>
		<ul style="list-style-type: none"> <li>• <i>Hierarchical Round Robin (HRR)</i></li> <li>• <i>Stop&amp;Go</i></li> <li>• <i>Jitter EDD</i></li> </ul>

Tabela II.1 Mecanismos de Escalonamento de Banda Passante.

### II.2.3.1 Mecanismos de Escalonamento Baseados em Prioridade

#### 1.- Prioridade Estática (*Static Priority (SP)*)

Neste algoritmo, cada fila tem atribuída uma determinada prioridade. Uma fila com baixa prioridade é servida somente quando não há células nas filas de maior prioridade que ela, ou seja as células com maior prioridade são atendidas primeiro, ainda que as células de prioridade menor tenham chegado antes.

#### 2.- Algoritmo de Prioridades Baseado em Limiares [LEE93]

Este algoritmo é uma versão do algoritmo SP, e foi proposto para um sistema que somente serve duas classes de tráfego. Uma das filas possui um determinado limiar, medido no número de células que contém a fila. Enquanto esse limiar não for atingido, o algoritmo de

escalonamento atende as células de ambas filas usando uma forma RR. Se o limiar é ultrapassado, então o algoritmo transmite exclusivamente as células da fila que é servida em base a esse limiar.

### **II.2.3.2 Mecanismos de Escalonamento Baseados em Acesso Justo Conservando o Trabalho**

#### **1.- Enfileiramento Justo (*Fair Queuing* (FQ)) [DEM89]**

Este algoritmo aproxima-se de PS emulando uma política de serviço Round Robin Bit a Bit (BRR). A cada célula é atribuída um tempo de serviço limite, que é o tempo no qual a célula deveria abandonar a fila se fosse servida usando um esquema BRR. As células são servidas segundo a ordem crescente de seus tempos de serviço limites.

#### **2.- Enfileiramento Justo Ponderado (*Fair Queuing* (WFQ)) [PAR93]**

Este algoritmo é uma versão generalizada de FQ (também conhecido Packet-Based Generalized Processor Sharing (PGPS)). Opera com a mesma filosofia, mas podem ser distinguidos dois aspectos:

- Emula uma disciplina GPS, portanto as filas têm ponderações diferentes. A ponderação permite que cada uma das conexões obtenha uma porção ponderada da banda passante total do canal
- Os tempos de finalização (ou tempos virtuais de término) com que se marca cada célula de uma conexão, corresponde ao tempo em que essa célula deveria abandonar a fila, segundo o estabelecido por uma disciplina GPS.

#### **3.- *Self-Clocked Fair Queuing* (SCFQ) [GOL94]**

Este algoritmo também atribui tempos limites de serviço às células e as serve na ordem crescente de seus tempos. Os algoritmos FQ e WFQ calculam estes tempos virtuais baseados em um sistema hipotético, mas SCFQ calcula estes tempos de acordo a como progride o trabalho que tem o sistema no momento em que chega essa célula. Calcular os tempos virtuais é muito complexo e, pode não ser prático de implementar. Portanto em SCFQ [GOL94] as células são estampadas com um tempo virtual de término. A fórmula que

determina o valor para o limite de serviço,  $F_k^i$ , da  $i$ -ésima célula da  $k$ -ésima fila está dado por:

$$F_k^i = \max\{F_k^{i-1}, v(a_k^i)\} + t_a$$

onde  $t_a$  é o tempo que o servidor demora em atender uma célula,  $a_k^i$  é o instante de chegada da  $i$ -ésima célula da  $k$ -ésima fila, e  $v(a_k^i)$  é o tempo virtual de finalização da célula que está sendo servida no instante  $a_k^i$ .

SCFQ a diferencia de FQ e WFQ, tem a vantagem de requerer pouco esforço computacional para calcular os tempos limites.

#### 4.- *Espaçamento Virtual (Virtual Spacing (VS)) [ROB94]*

O método VS publicado em [ROB94] é similar ao SCFQ. Neste esquema as células são estampadas com um tempo virtual de finalização igual a  $F_k^i = \max\{F_k^{i-1}, v(a_k^i)\} + t_a$ , onde  $v(a_k^i)$  não pode ser maior que o tempo registrado de qualquer célula que já esteja esperando quando seu valor é atualizado. Isto implica que as células em espera são espaçadas em um intervalo de  $t_a$ .

#### 5.- *Relógio Virtual (Virtual Clock (VC)) [ZHA90]*

O algoritmo de relógio virtual é uma “imitação” da disciplina de serviço TDM (*Time Division Multiplexing*). TDM é baseado em fatias (*slots*) de tempo, não permite nenhuma multiplexação estatística e garante uma banda passante fixa para as conexões ao atribuir fatias fixas de tempo para a transmissão. Se uma fila não tem dados para enviar no slot atribuído, essa fatia de tempo é perdida e a banda passante não é usada. A disciplina de serviço TDM é trabalho não conservativo. No entanto, com o esquema de VC a multiplexação estatística pode ser realizada e, os slots são utilizados enquanto exista uma fila não vazia em algum momento.

A cada célula atribui-se um tempo de serviço limite (ou tempo virtual), que é o tempo no qual uma célula deixaria a fila se estivesse usando TDM. As células são enviadas de acordo à ordem de seu tempo virtual.

#### 6. - *Escalação Pulso (Pulse Scheduling) [MUK92]*

Este método generaliza o método de relógio virtual de Zhang [ZHA90] para incluir fontes com rajadas. Lembre que o método de relógio virtual usa somente a taxa promedio do tráfego

fonte na computação dos tempos de finalização. O Escalonamento Pulso estende este conceito ao considerar as propriedades em rajada da conexão. Seja uma conexão  $i$  que transmite a taxa pico  $\lambda_i$  por uma duração  $t_{1i}$  e não transmite por um tempo  $t_{2i}$ . O método de escalonamento pulso atribui tempos de finalização usando a taxa máxima para  $\lambda_i \times t_{1i}$  células. À célula seguinte atribui-se um tempo virtual de  $t_{2i}$  e o processo repete-se desta maneira. O tempo virtual segue as propriedades das rajadas da conexão e o esquema de escalonamento pulso entrega um tratamento preferencial às conexões baseadas em sua taxa máxima advertida. Este esquema é idêntico ao esquema de relógio virtual quando  $t_{2i}=0$ .

### **7.- Worst Case Fair Weighted Fair Queuing (WF<sup>2</sup>Q) [BEN96]**

Em [BEN96] foi proposta uma política chamada WF<sup>2</sup>Q, e é considerada a aproximação mais exata de GPS. Em um sistema WFQ a célula escolhida para ser servida é aquela que teria completado seu serviço em um tempo  $t$  no sistema GPS. Em um sistema WF<sup>2</sup>Q são calculados dois tempos 1) o tempo virtual de início (o instante em que a célula deveria começar a ser transmitida se o sistema operasse com GPS, e 2) o tempo virtual de finalização (instante em que a célula deveria abandonar a fila se fosse servida usando GPS). Em um determinado tempo  $t$  uma célula é elegível para ser transmitida somente se seu tempo virtual de início é menor ou igual a  $t$ . As células pertencentes a este conjunto de células elegíveis são transmitidas na ordem crescente do tempo. Foi demonstrado que ao fazer isto, WF<sup>2</sup>Q entrega um serviço quase idêntico a GPS, diferenciando-se apenas em uma célula. Ao reduzir a complexidade de implementação, [BEN97] estende este esquema a WF<sup>2</sup>Q+ usando uma função de tempo virtual de baixa complexidade.

### **8.- Escalonamento Round Robin Ponderado (Weighted Round Robin (WRR)).**

WRR é uma extensão de RR. No método RR, no caso de existir  $N$  conexões, cada uma em uma fila separada, o servidor RR visita em cada ciclo cada uma das filas, em ordem fixa, e serve uma célula das que está em espera. Isto significa que o servidor RR compartilha a banda passante do canal equitativamente entre as filas. O comprimento do ciclo neste caso é de  $N$  células.

O algoritmo WRR diferencia-se do RR, no fato de atribuir porções ponderadas de banda passante as filas, em vez de atribuir porções iguais. Para isto são dados pesos ou ponderações a cada fila e, a banda passante é atribuída dependendo dessas ponderações, em cada ciclo que

o servidor faz. Isto significa que se uma fila tiver uma ponderação  $w_i$ , essa fila obtém uma banda passante equivalente a  $\frac{w_i \times (\text{capacidade do canal})}{\sum_{1 \leq i \leq N} w_i}$ .

### 9.- Fatia de Tempo Dinâmica (*Dynamic Time Slice (DTS)*) [SRI93]

No algoritmo DTS, para cada fila é atribuída uma fatia de tempo. A fila é atendida até que o tempo expire ou se esgotem as células, o que acontecer primeiro. Se existem  $n$  filas e, a cada uma é atribuída uma fatia de tempo  $T_i$ , então cada fila obtém uma banda passante de  $\frac{T_i}{\sum_i T_i}$ , vezes a capacidade do canal, com  $0 \leq i \leq n$  e,  $\sum_i T_i \leq D_c$ ; onde  $D_c$  é o tempo do ciclo DTS. Dado que o tempo de ciclo é fixo, pode ser garantido um certo retardo para as filas que assim o requeiram.

### 10.- Deficit Round Robin (DRR) [SHR96]

O esquema DRR é uma extensão do RR. Este algoritmo mantém um “contador de déficit” por cada conexão. O contador de déficit é colocado em zero cada vez que a fila está vazia. O contador de déficit é iniciado com a ponderação da conexão quando a fila chega a backlogged. A ponderação neste esquema é o número de células que uma fila pode transmitir em um determinado ciclo. A  $i$ -ésima fila pode transmitir uma quantidade  $Q_i$  de células em cada ciclo. Dado que uma conexão pode não ter suficientes células para transmitir em um ciclo, pode ser construído um déficit por cada uma das conexões. Se a fila não tem suficientes células para transmitir em um ciclo, registra-se um déficit de células ( que é igual ao número de células que faltou para completar  $Q_i$ ). Este déficit é cumulativo, isto é, os déficit dos diferentes ciclos são somados. Desta forma, o déficit atual é igual ao déficit anterior, se há algum, menos o número de células enviadas, é dizer  $DC_i = DC_i + (Q_i - \text{células enviadas}_i)$ . A conexão pode obter vantagem do déficit disponível no ciclo seguinte, o qual significa que, a quantidade de banda passante que o fluxo pode usar é a soma de seus déficit acumulados mais  $Q_i$ .

### 11.- Delay Earliest-Due-Date (Delay-EDD) [FER90]

Os esquemas descritos acima referem-se somente a entregar uma banda passante justa a cada uma das conexões. Mas, a garantia do retardo é essencial para serviços de tempo real. Um método para garantir retardos em redes de área estendidas (WAN) é proposto em [FER90], o

qual é chamado *Delay Earliest Due Date (Delay DD)*. Este esquema proporciona dois serviços: um com limites de retardos deterministas e o outro com limites de retardos estatísticos. Os tempos limites para cada célula são atribuídos com base no tempo de serviço. Os tempos limites para as células que pertencem ao serviço com limites de retardo deterministas são enfileirados de uma maneira tal que, se há células com tempos limite sobrepostos, estes tempos limite são apropriadamente reduzidos de forma que sejam cumpridas as garantias de retardo. Para servir as filas, o servidor compara as etiquetas de tempo limite de cada célula que está primeira na fila de espera, tanto das filas com serviço de retardo determinista como estatística. A comparação é feita com relação ao tempo de saída (tempo limite da célula na fila estatística) e ao tempo de começo (tempo limite do tempo de serviço) da célula da fila determinística. Se o último é menor que o primeiro, a fila de retardo determinístico é servida imediatamente.

### **12.- Rotating Priority Queues + (RPQ+) [WRE97]**

Este esquema, usa um conjunto priorizado de filas FIFO (*First-In-First-Out*). A ordem das prioridades é revisada periodicamente para incrementar a prioridade das células que estão esperando. Ao fazer isto, é provado que RPQ+ pode fornecer garantias de retardo de pior caso superiores ao de um esquema de prioridade estática e aproximadamente ao ótimo do servidor *Delay-DD* [WRE97]. A eficiência de RPQ+ aumenta com as prioridades re-etiquetadas e aproxima-se a *Delay-DD* no limite. RPQ+ usa  $2P$  filas FIFO ordenadas. Estas filas são etiquetadas da prioridade mais alta às mais baixas com índice como:  $0^+$ ,  $1$ ,  $1^+$ ,  $2$ ,  $2^+$ , ...,  $(P-1)$ ,  $(P-1)^+$ ,  $P$ . RPQ+ sempre seleciona a célula da fila de maior prioridade para a transmissão.

## **II.2.3.3 Mecanismos de Escalonamento Baseados em Acesso Justo de Trabalho Não Conservativo**

### **1.- Parar-e-Ir (*Stop-and-Go*) [GOL90]**

O algoritmo Parar-e-Ir trata de preservar “suaves” as características do tráfego quando atravessa a rede. O tempo é dividido em quadros, isto é supõe-se que o canal de entrada e o canal de saída tem quadros de tamanho fixo. Estes quadros não precisam estar sincronizados. A transmissão das células que chegam em um quadro de entrada é retardada até que comece o quadro seguinte no canal de saída. Com este algoritmo, uma célula tem limitado os retardos

máximo e mínimo que vão desde uma fonte a um destino. Uma vez que o retardo depende do tempo de quadro, este algoritmo propõe múltiplos tamanhos de quadro.

## 2.- Round Robin Hierárquico (*Hierarchical Round Robin (HRR)*) [KAL90]

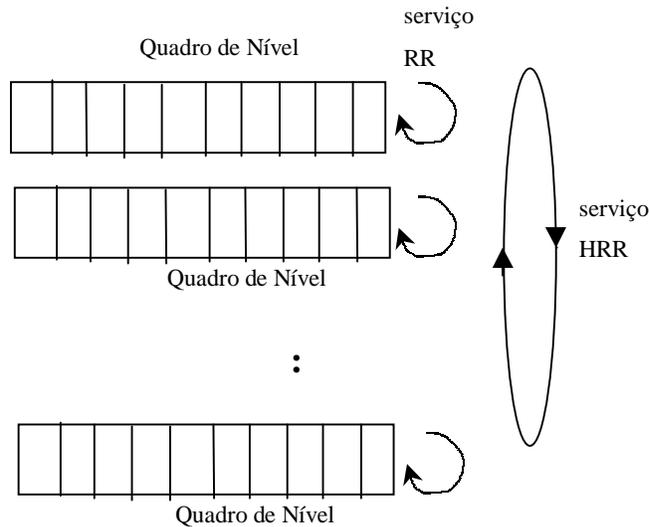


Figura II.5 Algoritmo HRR

Um servidor de HRR tem vários níveis, onde cada nível fornece um serviço RR a um número fixo de segmentos (slots). A um canal é atribuído um número determinado de segmentos de serviço em cada um dos níveis. O servidor serve em forma cíclica as filas dos diferentes níveis. O tempo que o servidor toma para servir os segmentos de um determinado nível é denominado tempo de quadro desse nível. HRR tem a capacidade de dar a cada nível uma quantidade constante de banda passante. Os níveis “maiores” (com maior número de segmentos) obtêm mais banda passante que os níveis “menores”, assim o tempo de quadro de um nível maior é mais curto que o tempo de quadro nos níveis menores.

Dado que um servidor sempre completa um ciclo através de seus segmentos uma vez a cada tempo de quadro, pode ser fornecido um limite para o retardo máximo aos canais atribuídos nesse nível. Note que, de existir somente um nível HRR torna-se um WRR.

O algoritmo HRR original é não-conservativo, já que se não há células esperando em uma fila quando corresponde servi-la, o servidor fica ocioso. Dessa forma, a duração de um ciclo é sempre constante. HRR pode fazer trabalho conservativo ao atribuir temporariamente os slots não usados a outras conexões.

### **3.- Jitter Earliest Due Date (jitter-EDD) [VER91]**

O algoritmo de Jitter -EDD estende o algoritmo Delay-EDD para entregar limites aos retardos máximo e mínimo (isto é, um limite a variação do retardo (jitter)). Depois que uma célula foi servida em um servidor, é colocada nesta uma etiqueta com o valor da diferença entre seu tempo limite para ser transmitida e seu tempo de transmissão atual. Um regulador de tráfego é usado no comutador seguinte, que guarda a célula por um período igual ao período etiquetado pelo comutador anterior, antes de que seja declarada elegível para ser transmitida novamente. Dado que cada célula obtém um retardo constante em cada comutador, pode ser entregue um limite à variação de retardo. Isto é, o algoritmo entrega uma garantia de obter os retardos mínimos e máximos requeridos.

Os algoritmos de escalonamento descritos anteriormente podem ser emulados usando o meta-algoritmo proposto na seção III e, seu desempenho pode ser avaliado usando a nova metodologia proposta na Seção IV.2.

## **II.3 Estado da Arte sobre Algoritmos de Congestionamento**

Para transportar múltiplos serviços, uma rede ATM requer do uso de funções de controle de admissão (CAC), algoritmos de escalonamento (explicados em II.2) e, mecanismos de controle de fluxo. Apesar disto, os pontos de contenção podem ficar congestionados. O congestionamento acontece cada vez que a carga instantânea de tráfego que chega a uma fila excede sua taxa instantânea de serviço. Este excesso de tráfego causa um rápido incremento no tamanho das filas, como também no retardo. Nos casos extremos pode causar a sobrecarga dos buffers, o qual implica a perda de células por falta de espaço de armazenamento.

Para manter as condições dos buffers de acordo com os objetivos de QoS, é preciso implementar mecanismos de controle de congestionamento nos pontos de contenção. A função de controle de congestionamento decide quais células descartar quando a ocupação do buffer é tal, que deteriora a QoS das conexões.

Existem duas classes importantes de tratamento do congestionamento: a) a classe dos métodos que pretendem evitar o congestionamento e, b) a classe dos métodos que controlam e recuperam do congestionamento.

Os métodos que pretendem evitar o congestionamento aplicam mecanismos de controle de fluxo em laço aberto ou fechado [YAN95] e, requerem a cooperação do sistemas finais. A reserva de recursos, o CAC e as políticas de controle de tráfego estão entre os mecanismos habitualmente usados para evitar o congestionamento. Os mecanismos de laço fechado baseiam-se na retroalimentação para prevenir e recuperar do congestionamento. A retroalimentação pode ser explícita, implícita, baseada em janelas, baseada na taxa, fim-a-fim e salto-a salto (hop-to-hop) . Os mecanismos para evitar ou recuperar congestionamento são aplicados dentro da rede e não requer da colaboração dos sistemas finais [YAN95].

Os dois enfoques explicados anteriormente, tem sido implementados nas atuais redes ATM. Por exemplo, as categorias de serviço CBR e VBR usam mecanismos para evitar o congestionamento com controle de fluxo de laço aberto, dado que este método tem um comportamento melhor para tráfegos de tempo real. Para ABR, é usado o método de evitar o congestionamento com mecanismos de laço fechado. Para o caso de serviço UBR geralmente deveria ser usado o espaço de buffer disponível. Então, para estes serviços é crítico decidir quando descartar células ou quando descartar quadros AAL5 completos, para melhorar o acesso ao recurso buffer.

Em [LAB99] é entregue um resumo de políticas de descarte seletivo usadas sobre serviços baseados em quadros (tal como AAL5). Se cada vez que uma célula que pertence a um quadro é descartada, o quadro AAL5 completo também é descartado, os recursos da rede podem ser conservados e a eficiência global pode ser incrementada. Para atingir este objetivo, podem ser usadas diversas técnicas tais como: Descarte Parcial de Pacotes (*Partial Packet Discard* (PPD)) [FLO95b], PPD preemptivo (pPPD)[LAB99b], Descarte Prematuro de Pacotes (*Early Packet Discard* (EPD)) [FLO95b], Descarte Prematuro Seletivo de Pacotes (*Early Selective Packet Discard* (ESPD)) [CHE98], EPD com Hysteresis [TUR96], EPD com Atribuição Justa de buffer (EPDFBA)[GOY97], Descarte de Pacotes com Prioridade de Idade (*Age priority packet discarding* (APPD))[CHI98] .

O intuito da seguinte seção é entregar uma visão dos algoritmos de descarte de células usados em serviços de tráfego VBR.

Diversas propostas para controlar as situações de congestionamento têm sido publicadas na literatura [YAN95]. Entre elas, técnicas de descarte de células foram propostas para evitar a sobrecarga nas filas em redes comutadas de voz e dados [PET89,CID95]. Algumas dessas técnicas são descritas a seguir.

### **II.3.1 Descarte pela Cauda (*Dropping Tail (DT)*)**

O procedimento comum desses mecanismos de descarte é eliminar pacotes do fim da fila quando ocorrem situações de congestionamento, isto é, cada vez que não tem espaço de buffer disponível. No caso de tráfego VBR, como por exemplo tráfego de voz, para diminuir o congestionamento pode ser suficiente descartar as células de menor prioridade que chegam ao buffer quando atingir um limiar predeterminado (que especifica o congestionamento). No caso de serviços de tempo real, esse limiar pode ser calculado baseado no retardo máximo permitido.

### **II.3.2 Descarte pela Frente da Fila (*Front Dropping (FD)*)**

Um esquema alternativo de descarte, chamado descarte pela frente da fila [YIN93], permite a entrada na fila de todos os pacotes que chegam, descartando o pacote que está na cabeça da fila quando ocorre congestionamento. Em [YIN93] foi mostrado que os mecanismos que realizam o descarte pela frente não somente melhoram o desempenho em relação ao retardo, em um canal inter-nó, mas também diminuem as perdas globais para um tráfego com restrição de retardo.

Um esquema de FD eficiente foi definido em [SRI89,SRI91]. Na proposta de [SRI91], o descarte é realizado prioritariamente em células ATM que transportam somente os bits menos significativos das amostras de voz, quando a fila do multiplexador excede um limite predefinido. Desta forma, o esquema de controle de congestionamento acelera a transmissão de alguns bits das amostras de voz (através do descarte de outros bits), quando o multiplexador cai em congestionamento. O descarte de alguns bits de amostras de voz é possível devido à informação redundante presente nos sinais de voz [BOW87]. A perda até certo limite causa pouca degradação da qualidade do sinal, podendo até ser imperceptível. A idéia intuitiva que existe neste mecanismo de descarte é que, é melhor perder os bits menos

significativos de algumas amostras de voz, do que perder amostras completas. Alternativamente, implementações mais complexas e eficientes consistem em enfileirar todas as células que cheguem quando exista espaço disponível no buffer [RAM96,CHO98]. Quando o buffer fica saturado, é feita a decisão de remover uma célula qualquer para acomodar a célula que está chegando. Se nenhuma das células pode ser removida, então a célula que chega é descartada.

Para conhecer com maior profundidade os fenômenos que acontecem em um multiplexador operando com o mecanismo de descarte de células, um conjunto de novas medidas transientes de desempenho foram avaliadas e seus resultados são apresentados no Capítulo IV.

## **II.4 Conceitos sobre o Tráfego de Voz**

### **II.4.1 Voz sobre ATM**

Desde o início ATM foi projetada para ser uma tecnologia multi-serviço. Embora uma rede ATM esteja equipada para transferir voz eficientemente, ainda este serviço não é bastante popular. As razões para isto são varias, entre elas a existência de tecnologias altamente competitivas como Ethernet 100 Mbps / Gigabits, que oferecem serviços similares, com uma mínima atualização em infra-estrutura. Mas, quando o caso é transmitir voz em redes WAN, ATM tem seu próprio espaço, sendo competitiva por sobre outras tecnologias.

### **II.4.2 Características do Tráfego de Voz**

A seguir são apresentadas algumas características do tráfego de voz, que são de interesse para a análise de desempenho para multiplexadores ATM com tráfego de voz, que será feita no capítulo IV.

#### **II.4.2.1 Retardo Fim-a-Fim**

A voz é muito sensível ao retardo e medianamente sensível as variações de retardo (jitter). É crítico, então, manter o retardo ao mínimo possível em uma comunicação interativa. Quando o retardo atinge os 30 mseg, é preciso o uso de um supressor de eco. Quando esse supressor é usado, os retardos nas redes podem ser incrementados até atingir 150 mseg sem que exista

uma degradação na qualidade da voz. A recomendação G.114 da ITU-T permite os seguintes retardos limites, para uma transmissão de uma via e para conexões com controle de eco.

<i>Retardo</i>	<i>Aceitabilidade</i>
0-150 mseg	aceitável para a maioria das aplicações de usuários
150-400 mseg	aceitável quando o impacto da qualidade é informado
>400 mseg	inaceitável

Tabela II.2 Limites de retardo dados pela recomendação G.114 da ITU-T.

Há varios tipos de retardos que podem acontecer na rede ATM. Estes pode se dar por uma ou mais das seguintes razões. :

- a) **Retardo na construção da célula:** é o tempo tomado para preencher uma célula antes de transmiti-la.
- b) **Retardo de enfileiramento:** algumas vezes, as células podem chegar tarde sendo enfileiradas em um buffer para sua posterior transmissão. O tamanho do buffer é traduzido em retardo para as células, portanto o tamanho dos buffers devem ser tais que não excedam o retardo máximo estimado.
- c) **Retardo de codificação:** é o tempo tomado pelos algoritmos de compressão para codificar o sinal análogo em digital.

#### II.4.2.2 Supressão de Silêncio

A voz é de natureza variável. Os sinais de voz são tipicamente compostos por rajadas de informações (surto de voz) intercalados por períodos de silêncio [BRA68] (Figura II.3). Estes silêncios são produzidos porque existem pausas entre as sentenças e as palavras, sem fala na outra direção. Outro motivo, é o fato que comunicação humana é de duas vias (half-duplex), isto é uma pessoa fica em silêncio enquanto a outra fala. Tem sido calculado que as rajadas de voz, tem um comprimento médio de 0.4 a 1.2 seg. Os intervalos de silêncio tem um comprimento médio que varia entre 0.6 a 1.8 seg. [SCH96]. Estes estudos demostram que, a voz tem um fator de atividade média que fica entre 35% e 42%, dependendo de fatores sociais e da idiossincrasia. As caraterísticas anteriormente mencionadas, permitem economizar banda passante ao deixar de transmitir células durante os períodos de silêncio. Isto é

conhecido como supressão de silêncio. É claro que os períodos de silêncio devem poder ser restaurados pelo receptor [DAI86].

Por outro lado, durante os períodos de atividade, a fonte de voz é amostrada a uma taxa constante, sendo o número de amostras por surto variável, uma vez que a duração da rajada é variável.

No caso em que a informação de voz é transmitida por células ATM, o sinal de voz é transformado a formato digital através do uso de codificadores de voz, para posteriormente formar as células ATM com a informação digitalizada.

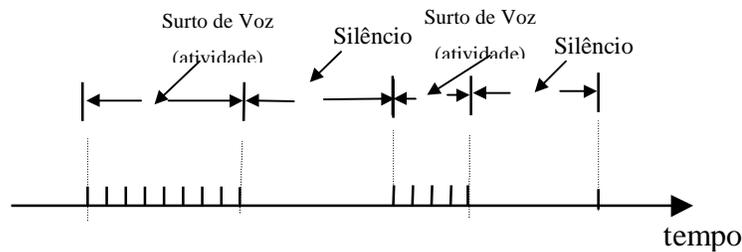


Figura II.6 Caracterização da voz

## II.5 Resumo

Neste capítulo foram apresentados diversos conceitos que serão usados, posteriormente, no desenvolvimento da tese.

Em primeiro lugar, foram fornecidos alguns conceitos sobre QoS em Redes ATM. A seguir, foram apresentados o estado da arte em algoritmos de congestionamento e o estado da arte em algoritmos de escalonamento de banda passante. Finalmente, foram também abordados alguns conceitos sobre a transmissão do tráfego de voz em redes ATM.

## Capítulo III

# Meta-Algoritmo para Políticas de Escalonamento

### III.1 Introdução

Explorar eficientemente os recursos da rede significa atender simultaneamente a maior quantidade possível de usuários, o que é conseguido através da utilização da multiplexação estatística. Devido à conduta aleatória dos usuários, é preciso considerar que, ao utilizar multiplexação estatística, existe a possibilidade que, em um determinado momento, a taxa instantânea de chegada de pacotes a um certo comutador da rede supere sua taxa de serviço. Para evitar (ou diminuir) a perda de pacotes que pode acontecer nestes casos, os comutadores implementam filas (buffers), que permitem armazenar temporariamente o excesso de pacotes recebidos, para transmiti-los posteriormente.

Dado que existem filas, surge a necessidade de implementar políticas de serviço (ou políticas de escalonamento) encarregadas de determinar a ordem de serviço, ou seja, a ordem na qual os pacotes de cada fila devem ser transmitidos. A política de escalonamento implementada influi na QoS oferecida aos diferentes tipos de tráfego já que, por exemplo, a política de escalonamento determina (a longo prazo) a banda passante atribuída a cada classe de usuário, e essa pode ser transformada em um determinado requisito de banda passante, através do método de Banda Passante Equivalente [VEC95].

Na literatura têm sido propostos variadas políticas (ou mecanismos) de escalonamento e, continuamente, são feitas novas propostas. Cada mecanismo tem uma peculiaridade que o torna mais adequado a certas situações. É necessário avaliá-los e compará-los com o objetivo de decidir qual deles se constitui na melhor alternativa para ser implementada segundo os requisitos de QoS. Por este motivo, é útil gerar um esquema global de avaliação que permita medir o desempenho dos vários algoritmos.

Nesta tese propõe-se uma metodologia para emular e, posteriormente, avaliar o desempenho dos vários algoritmos de escalonamento.

Para emular a operação de qualquer algoritmo de escalonamento propõe-se um meta-algoritmo para algoritmos de escalonamento, denominado Meta-Algoritmo LDS (Load Dependent Scheduling). A emulação dos distintos mecanismos de escalonamento utilizando o LDS é simples e é obtida modificando os parâmetros de operação do meta-algoritmo proposto.

Para avaliar o desempenho dos diferentes algoritmos de escalonamento propostos, avalia-se o desempenho de LDS em função de seus parâmetros de operação. Isto significa que, quando LDS é avaliado funcionando de acordo com os parâmetros que permitem emular um determinado algoritmo, as medidas de desempenho que são obtidas para o LDS correspondem às medidas de desempenho desse algoritmo em particular. Desta forma, o LDS apresenta a vantagem de poder avaliar o desempenho de qualquer algoritmo de escalonamento a partir de um único conjunto de equações.

Na seção IV.3 é apresentado a análise de desempenho do meta-algoritmo LDS, para um multiplexador que comporta tráfegos de voz e dados.

As características do meta-algoritmo proposto são as seguintes:

- ◆ é flexível, uma vez que pode atender diferentes tipos de serviço ou tráfego, podendo adaptar-se facilmente para atender novos serviços ainda não definidos;
- ◆ é cooperativo, na medida que permite garantir a QoS das conexões, já que é possível assegurar uma certa banda passante para cada classe de tráfego;
- ◆ é adaptativo, dado que a quantidade de células transmitidas de cada tipo de tráfego depende do estado de carga do multiplexador;
- ◆ é dinâmico, pois permite redistribuir de forma rápida a banda passante que não é momentaneamente usada por algum usuário (ao qual estava atribuída);
- ◆ é geral, já que ao escolher de forma adequada seus parâmetros, pode emular diferentes tipos de políticas de escalonamento propostas na literatura (ver seção II.2).

## III.2. Descrição do Meta-Algoritmo LDS

Antes de explicar a operação de LDS, é feita a descrição do tipo de multiplexador considerado nesta tese. Na Figura III.1 é apresentado um multiplexador genérico capaz de atender ‘ $N$ ’ tráfegos distintos. Para cada tipo o multiplexador mantém um buffer, cuja disciplina de serviço é do tipo FIFO. Este multiplexador transmite as células dos diferentes tipos de tráfego por um mesmo canal de saída. A ordem e a duração com que são atendidas as células de cada um dos tipos de tráfego são especificadas pelo meta-algoritmo LDS implementado no servidor.

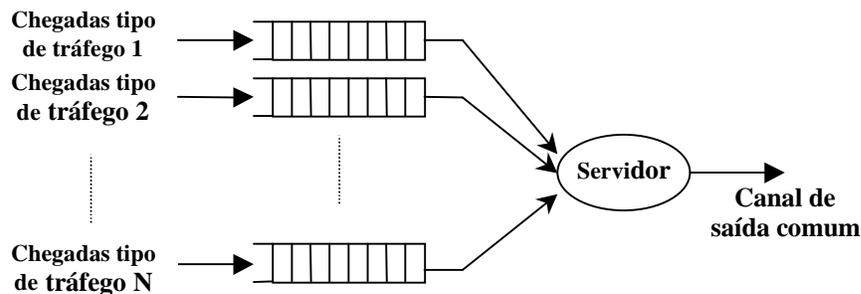


Figura III.1. Multiplexador genérico que serve a ‘ $N$ ’ tráfegos diferentes.

Na Figura III.2 é apresentado um diagrama temporal da operação do meta-algoritmo LDS. O LDS opera ciclicamente. Cada ciclo está composto de  $N+1$  fases, onde  $N$  corresponde ao número de tipos de tráfegos diferentes atendidos por um multiplexador.

A fase  $f_0$  corresponde a uma fase de inatividade. Esta fase tem uma duração diferente de zero somente quando, ao início de um ciclo, não existem células (de nenhum tipo) para transmissão pelo multiplexador. Nesse caso, o multiplexador espera a chegada da primeira célula, quando da início a fase do tipo de tráfego que chegou em primeiro lugar.

Durante a fase  $f_i$  ( $1 \leq i \leq N$ ), um multiplexador transmitirá exclusivamente células do tráfego  $i$ . A duração que cada uma das fases deve apresentar dependerá do estado em que se encontra o multiplexador (quantidade de células armazenadas em cada um dos buffers) ao início do ciclo. Isto implica que, em geral, o tamanho da fase  $f_i$  pode ser diferente em diferentes ciclos de operação do LDS. A duração de um ciclo corresponde à soma da duração de cada uma de suas fases.

Para exemplo e efeito de cálculo, nesta tese é considerada que o intervalo necessário para transmitir uma célula ATM corresponde a uma unidade de tempo. Por este motivo, daqui em diante não será diferenciado o tempo que dura uma fase da quantidade de células transmitidas em tal fase.

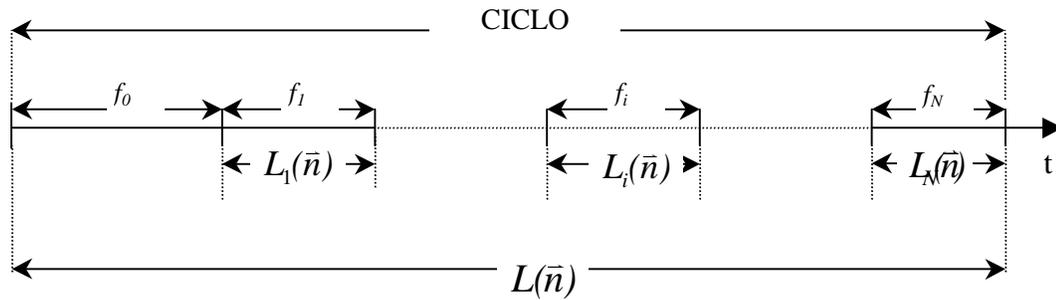


Figura III.2. Diagrama temporal da operação do LDS em um ciclo.

LDS opera de acordo com uma tabela, denominada Tabela LDS, implementada em um multiplexador (ver figura III.3).

1	2	3		N					
$n_1$	$n_2$	$n_3$		$n_N$	$L_1(\bar{n})$	$L_2(\bar{n})$	$L_3(\bar{n})$		$L_N(\bar{n})$
			.....					.....	

Figura III.3. Exemplo de uma Tabela LDS

Esta tabela possui  $2N$  colunas, onde  $N$  corresponde a quantidade de tráfegos distintos atendidos pelo multiplexador. As primeiras  $N$  colunas são rotuladas com  $n_i$  ( $1 \leq i \leq N$ ), onde  $n_i$  indica o estado do buffer do tipo de tráfego  $i$  (número de células desse tipo) no início do ciclo. As  $N$  colunas restantes são rotuladas com  $L_i(\bar{n})$ , onde  $L_i(\bar{n})$  corresponde ao número máximo de células que podem ser transmitidas durante a fase  $f_i$  (ver por exemplo a Tabela III.1, onde é mostrada a tabela para dois tipos de tráfego).  $L_i(\bar{n})$  depende do vetor de estado

$\bar{n} = (n_1, n_2, \dots, n_N)$ , que corresponde à quantidade de células presentes em cada um dos buffers no início de um ciclo. Por outro lado, a Tabela LDS possui tantas linhas quantas possam ser as condições lógicas diferentes para a carga do sistema

O algoritmo LDS opera da seguinte forma:

1. No início de um ciclo, o multiplexador determina o vetor de estado  $\bar{n}$  (que corresponde ao número de células que existe em cada fila nesse instante).
2. Se  $\bar{n} = \bar{0}$ , o sistema espera a chegada da primeira célula. A espera constitui a fase  $f_0$ . Quando a primeira célula chega, o multiplexador dá início a fase do tráfego que chega primeiro. Cabe destacar que a situação de carga  $\bar{n} = \bar{0}$ , não está especificada na tabela LDS, por não ser necessário.
3. Se  $\bar{n} \neq \bar{0}$ , o algoritmo passa então a ler as primeiras  $N$  colunas de cada fila da Tabela LDS, até achar a situação de carga que corresponde ao vetor de estado. Uma vez encontrada a linha, são lidas as  $N$  colunas seguintes, que especificam o valor de  $L_i(\bar{n})$ , com  $1 \leq i \leq N$ , para esse ciclo.
4. A seguir, são transmitidas as células da fase  $f_1$ , em seguida as da fase  $f_2$  e assim sucessivamente até a fase  $f_N$ .
5. Durante a fase  $f_i$  são transmitidas  $L_i(\bar{n})$  células, apenas se o multiplexador pode transmitir ininterruptamente  $L_i(\bar{n})$  células da fila  $i$ . Se ao início de um ciclo, o número de células do tráfego  $i$  é menor que  $L_i(\bar{n})$ , a fase  $f_i$  pode durar  $L_i(\bar{n})$ . Esta situação acontece somente se durante o tempo que transcorre, desde o início do ciclo até que termina de ser atendida a fase  $f_i$ , chegam células do tipo  $i$  suficientes para manter ocupado o multiplexador, até completar de atender as  $L_i(\bar{n})$  células, que devem ser transmitidas como quantidade máxima durante a fase.

6. Se as células da fase  $f_i$  acabam antes que expire o tempo  $L_i(\vec{n})$ , o algoritmo aborta a fase  $f_i$ , começando imediatamente a atender a fase  $f_{i+1}$ . Um caso particular ocorre quando, ao começo da fase  $f_i$  não existem células do tipo  $i$ , o que faz que o LDS salte a fase  $f_i$ .

### III.3 Emulação de Algoritmos Usando o Meta-Algoritmo LDS

A seguir é exemplificado como o meta-algoritmo LDS pode emular outros algoritmos propostos na literatura. Dois casos representativos de tipos de algoritmos são apresentados e, para cada caso, é especificada a tabela particular usada para a emulação do algoritmo.

#### III.3.1 Emulação de Algoritmos de Escalonamento com Prioridade

Nesses algoritmos, prioridades diferentes são atribuídas aos tipos de tráfego distintos. Deste modo, pode ser assegurado, por exemplo, que os tipos de tráfego com maiores restrições de retardo sejam transmitidos com maior prioridade através da rede. Existem vários algoritmos de escalonamento que usam a idéia de prioridades para administrar a banda passante do canal de saída, como foi dito na Seção II.3. Alguns deles são Filas+ de Prioridade Rotativa (*Rotating Priority Queue+*) [WRE97], Algoritmo de Prioridade Simples (*Simple Priority Algorithm*) [DRA93] e Algoritmo de Prioridades baseados em Limiar (*Threshold Based Priority Algorithm*) [LEE93].

$n_v$	$n_d$	$L_v(\vec{n})$	$L_d(\vec{n})$
$> 0$	$\geq 0$	$n_v$	0
0	$> 0$	0	1

Tabela III.1. Tabela LDS para o Algoritmo de Prioridade Simples [DRA93].

A Tabela III.1 ilustra como o meta-algoritmo LDS pode emular o Algoritmo de Prioridade Simples [DRA93]. Esse algoritmo foi proposto para um multiplexador que comporta dois tipos de tráfego: voz e dados. O tráfego de voz tem prioridade estática (não muda no tempo) sobre o tráfego de dados, tal como foi explicado na descrição dos algoritmos (ver seção II.2.1). Na tabela,  $n_v$  e  $n_d$  correspondem, respectivamente, ao número de células de voz e

dados presentes no multiplexador ao início de um ciclo. Por sua vez,  $L_v(\vec{n})$  e  $L_d(\vec{n})$  correspondem a duração das fases de voz e dados, respectivamente.

A primeira linha da Tabela III.1 indica que, enquanto existam células de voz no buffer correspondente, o multiplexador atenderá somente esse tráfego (independentemente da quantidade de células de dados que estejam armazenadas no buffer de dados). A fase de voz dura o tempo necessário para transmitir todas as células armazenadas no buffer de voz ao início do ciclo. Desta maneira, o tráfego de voz tem sempre maior prioridade de serviço que o tráfego de dados. A segunda linha da Tabela III.1 especifica que o tráfego de dados é servido pelo multiplexador somente quando não existem células no buffer de voz. A fase de dados dura apenas o tempo necessário para transmitir uma célula. Isto é feito como uma forma de garantir que o multiplexador não acumule uma quantidade excessiva de células de voz, enquanto é transmitido o tráfego de dados. Como este tipo de algoritmo é cíclico, imediatamente após ter transmitido uma célula de dados, o multiplexador volta a rever os buffers para garantir que as células de voz, que chegaram durante o serviço da célula de dados, sejam atendidas tão rapidamente quanto possível.

### **III.3.2 Emulação de Algoritmos de Escalonamento Tipo Round-Robin (RR)**

Essa classe de algoritmos também opera ciclicamente. Cada ciclo é formado por  $N$  fases. Em cada fase do ciclo é servido um só tipo de tráfego. A quantidade máxima de células servidas em uma mesma fase é igual para todos os ciclos.

Têm sido propostos diversos algoritmos de escalonamento que operam com essa filosofia. Alguns deles são: WRR, DTS e DRR, que foram explicados na seção II.2.3

A Tabela III.2 exhibe a tabela LDS que emula o algoritmo DTS para  $N$  classes de tráfego diferentes. A notação da Tabela III.2 é análoga àquela da Tabela III.1, com  $T_i$  ( $1 \leq i \leq N$ ) correspondendo ao número de células que devem ser transmitidas durante a fase  $f_i$ . Os valores de  $T_i$  são proporcionais à banda passante requerida por cada tipo de tráfego.

$n_1$	$n_2$	.....	$n_N$	$L_1(\vec{n})$	$L_2(\vec{n})$	.....	$L_N(\vec{n})$
$\geq 0$	$\geq 0$	.....	$\geq 0$	$T_1$	$T_2$	.....	$T_N$

Tabela III.2. Tabela LDS para algoritmo DTS

No caso em que, o número de células em uma fila seja menor que a quantidade de células que devem ser transmitidas nessa fase, DTS começa a atender imediatamente a fase seguinte. LDS emula corretamente esta situação, devido ao ponto 6 da operação do meta-algoritmo.

### III.4 Algoritmos de Escalonamento Propostos Nesta Tese

Como foi mencionado anteriormente, além de emular outros algoritmos, LDS pode ser programado para que execute novas propostas de algoritmos. A modo de exemplo, a seguir são propostas quatro tabelas de operação para LDS (Tabelas III.3, III.4, III.5 e III.6), que são aproximações de algoritmos tipo PS e GPS.

#### III.4.1 Transmissão proporcional

Na Tabela III.3, é apresentada a tabela LDS para o caso de um multiplexador que serve dois tipos de tráfego: um com restrições de tempo real (tráfego tipo 1) e outro sem restrições de tempo real (tráfego tipo 2). O algoritmo corresponde a uma aproximação do algoritmo PS (Processor Sharing), e é denominado Algoritmo Proporcional. Nesse algoritmo, em cada fase de um ciclo, o número de células transmitidas é proporcional ao número de células desse tipo que contém o multiplexador, ao início do ciclo. Não obstante, com o objetivo de manter limitado o retardo experimentado pelo tráfego de tempo real, se o número de células desse tipo de tráfego supera um determinado limiar (denominado  $u$ ), o multiplexador somente atende ao tráfego até que a ocupação desse buffer fique abaixo do limiar.

Na notação utilizada na Tabela III.3,  $T$  corresponde ao tamanho de um ciclo,  $n_1$  corresponde ao número de células de tipo 1 que estão no buffer correspondente, e  $n_2$  corresponde ao número de células de tipo 2.

$n_1$	$n_2$	$L_1(\bar{n})$	$L_2(\bar{n})$
$n_1 \leq u$	$\geq 0$	$T \left( \frac{n_1}{\sum_{v_i} n_i} \right)$	$T \left( \frac{n_2}{\sum_{v_i} n_i} \right)$
$n_1 > u$	$\geq 0$	$n_1 - u$	0

Tabela III.3. Tabela LDS para Algoritmo Proporcional

As Tabelas III.4 e III.5 são análogas àquela apresentada anteriormente. Isto é, correspondem a métodos de escalonamento que aproximam-se de PS, atribuindo a cada fila uma banda passante proporcional à carga dessa fila. Também é considerado um multiplexador que serve dois tipos de tráfego: um de tempo real e outro sem restrições de tempo real. Em ambas tabelas é especificado que quando o buffer destinado ao tráfego de tempo real supera um determinado limiar, apenas são transmitidas as células pertencentes a esse tipo de tráfego. A diferença entre a Tabela III.3, que estabelece a proporção de banda passante atribuída a cada fila, segundo o estado do multiplexador ao início do ciclo, e as Tabelas III.4 e III.5 é que as últimas estabelecem que a proporção de banda passante atribuída a cada fila considere também as células que chegam durante o ciclo. Pode-se então dizer que os algoritmos descritos pelas Tabelas III.4 e III.5 são preditivos. Isto significa que, mediante alguma heurística, pode-se prever (aproximadamente) a quantidade de células que chegarão durante o ciclo que começa e, com base nessa informação, tomar a decisão de quantas células de cada tipo devem ser transmitidas no ciclo.

As Tabelas III.4 e III.5 diferenciam-se no modo em que é feita a predição da chegada de células durante um ciclo.

### III.4.2 Algoritmo Preditivo Baseado na Estatística das Chegadas

Na Tabela III.4, é feita a suposição que a quantidade de células de tipo de tráfego  $i$  que chegam durante um ciclo é igual à taxa média de chegada de células  $\lambda_i$ , multiplicada pelo tamanho do ciclo,  $T$ .

A linha 1 da Tabela III.4 mostra que, quando a ocupação do buffer, destinado ao tráfego de tempo real, está abaixo de um determinado limiar (denominado  $u$ ), não é atribuída prioridade aos tráfegos. O sistema aproxima-se, assim, da operação de PS. Para o algoritmo, isto significa que o serviço somente está determinado pelas taxas de chegada de cada tipo de

tráfego, isto é, quanto maior a taxa de chegada do tráfego tipo  $i$ , tanto maior a quantidade de células desse tráfego que serão transmitidas em um ciclo. O objetivo do algoritmo é fazer com que a quantidade de células de tipo  $i$  transmitidas durante a fase  $i$  seja proporcional, não apenas à quantidade de células desse tipo presentes no buffer, mas também à velocidade de chegada desse tipo de células.

$n_1$	$n_2$	$L_1(\vec{n})$	$L_2(\vec{n})$
$n_1 \leq u$	$\geq 0$	$T \left( \frac{n_1 + \lambda_1 t}{\sum_{\forall i} (n_i + \lambda_i t)} \right)$	$T \left( \frac{n_2 + \lambda_2 t}{\sum_{\forall i} (n_i + \lambda_i t)} \right)$
$n_1 > u$	$\geq 0$	$n_1 - u$	0

Tabela III.4. Tabela LDS para um Algoritmo Preditivo baseado na estatística das chegadas

Na segunda linha da Tabela III.4, mostra-se a operação do algoritmo quando a ocupação do buffer do tráfego de tempo real supera um limiar  $u$ . Nesta situação, o algoritmo somente serve as células do tráfego de tempo real até que o buffer tenha menos células que o limiar  $u$ .

### III.4.3 Algoritmo Preditivo baseado na chegada do ciclo anterior

A Tabela III.5 mostra a operação do mesmo algoritmo descrito na Tabela III.4, exceto que neste caso é feita a suposição que a chegada de células de tipo  $i$  do ciclo atual é igual ao número de células que chegaram no ciclo anterior (denotada por  $c_i$ ).

$n_1$	$n_2$	$L_1(\vec{n})$	$L_2(\vec{n})$
$n_1 \leq u$	$\geq 0$	$T \left( \frac{n_1 + c_1}{\sum_{\forall i} (n_i + c_i)} \right)$	$T \left( \frac{n_2 + c_2}{\sum_{\forall i} (n_i + c_i)} \right)$
$n_1 > u$	$\geq 0$	$n_1 - u$	0

Tabela III.5. Tabela LDS para um Algoritmo Preditivo baseado nas chegadas do ciclo anterior.

### III.4.4 Algoritmo que aproxima a GPS

Na Tabela III.6, é apresentada a tabela LDS para um algoritmo de Escalonamento que aproxima a operação de GPS (*Generalized Processor Sharing*). Para isto, modifica-se o algoritmo apresentado na Tabela III.4, atribuindo ponderações distintas das ponderações originais. O parâmetro  $\alpha$  ( $0 < \alpha < 1$ ) que multiplica o número de células que deveria ser transmitido segundo a aproximação PS, estabelece as diferentes ponderações para as diferentes filas.

$n_1$	$n_2$	$L_1(\bar{n})$	$L_2(\bar{n})$
$n_1 \leq u$	$\geq 0$	$T \left( \frac{n_1 + \lambda_1 t}{\sum_i (n_i + \lambda_i t)} \right) \alpha$	$T \left( \frac{n_2 + \lambda_2 t}{\sum_i (n_i + \lambda_i t)} \right) (1 - \alpha)$
$n_1 > u$	$\geq 0$	$n_1 - u$	0

Tabela III.6. Tabela LDS para um Algoritmo que aproxima a operação de GPS.

### III.5 Resumo

Neste capítulo foi proposto um Meta-algoritmo, denominado LDS, que permite emular e avaliar os diferentes algoritmos de escalonamento propostos na literatura. O meta-algoritmo LDS permite emular os distintos algoritmos de escalonamento usando tabelas que descrevem o comportamento de um algoritmo em particular. Para avaliar o desempenho de um determinado algoritmo de escalonamento, pode-se então avaliar o desempenho de LDS em função de seus parâmetros de operação, obtendo desta forma as medidas de desempenho para o algoritmo em estudo. Esta avaliação é realizada na Seção IV.3

Ainda, neste Capítulo, foram propostos novos algoritmos de escalonamento, apresentando para cada um deles suas Tabelas LDS.

## Capítulo IV

# Análise de Desempenho de Multiplexadores

### IV.1 Solução usando *Performability*

Nesta seção é apresentado o método matemático genérico desenvolvido em [VAL93], o qual é usado aqui para avaliar diferentes medidas de desempenho, para os multiplexadores estudados nas seções posteriores.

Em primeiro lugar é definido e resolvido um problema particular de “*performability*” [MEY80, MEY92], cuja solução é então aplicada na avaliação de diversas medidas transientes.

Para estabelecer o problema de interesse, a seguir é introduzida a notação que será usada:

- Seja  $Y_n = \{Y(n), n \geq 0\}$  a cadeia de Markov que descreve o comportamento do sistema nos instantes das transições de  $Y_n$ .
- Seja  $E = \{e_i, i=1, \dots, |E|\}$  o espaço de estados finitos associado a  $Y_n$ , onde  $|E|$  corresponde ao número total de estados de  $E$ .
- Seja  $Z$  a matriz de probabilidades de transição entre os estados de  $Y_n$ .
- Seja  $z_{ij}$  a probabilidade de transição de um passo entre os estados  $e_i$  e  $e_j$  da cadeia de Markov  $Y_n$ .
- Seja  $T \subset (E \times E)$  o conjunto de possíveis transições entre estados da cadeia de Markov  $Y_n$ .
- Seja  $t_{ij} \in T$  a transição do estado  $e_i \in E$  para o estado  $e_j \in E$ .
- Seja  $\eta(i,j)$  a função de atribuição de recompensa associada a cada uma das transições  $t_{ij} \in T$ . Esta recompensa corresponde à recompensa ganha pelo sistema cada vez que transita de um estado  $e_i$  para um estado  $e_j$ . Esta função de atribuição de recompensas deve ser definida previamente à avaliação de cada medida de desempenho.
- Seja  $\Theta = \{\theta_k, k = 1, \dots, \theta_\Theta\}$  o conjunto de diferentes recompensas associadas à transições do conjunto  $T$ . Em geral,  $|\Theta| \leq |E|^2$ , devido ao fato que várias transições

podem estar associadas a uma mesma recompensa e que algumas transições entre estados não existem.

- Seja  $RTAD(n)$  a recompensa total que o sistema ganha nas primeiras  $n$  transições da cadeia de Markov  $Y_n$ . O ganho de recompensas somente pode acontecer nas transições, obedecendo uma função de atribuição de recompensas  $\eta(i,j)$ .  $RTAD(n)$  é uma variável aleatória, devido ao fato que, durante um intervalo de observação, a cadeia de Markov pode transitar por diferentes caminhos. Sem perda de generalidade, supõe-se que  $RTAD(0)=0$ .

O problema de “*performability*” de interesse para este trabalho, é a avaliação da função de distribuição cumulativa (FDC) de  $RTAD(n)$ , denominada  $G(n,r)$ , para uma medida de desempenho qualquer. Onde  $r$  é a recompensa a atingir.

Usando as definições anteriores, tem-se:

$$G(n,r) = P[RTAD(n) \leq r], \quad (IV. 1)$$

Para avaliar  $G(n,r)$ , define-se  $G_j(n,r)$  como a probabilidade que  $RTAD(n)$  seja menor ou igual a  $r$  e que o estado da cadeia de Markov, imediatamente após  $n$  transições, seja  $e_j$ . Considerando todos os casos possíveis, é obtida a seguinte expressão:

$$G(n,r) = \sum_{e_j \in E} G_j(n,r) \quad (IV. 1)$$

Para continuar a análise são necessárias algumas definições adicionais:

- Seja  $RTAD_j(n)$  a variável aleatória  $RTAD(n)$  para o caso especial onde a cadeia de Markov está no estado  $e_j$  após  $n$  transições.
- Seja  $\mathfrak{R}_j^n$  o conjunto de todos os possíveis valores que pode tomar  $RTAD_j(n)$ .
- Seja  $w$  um dos valores pertencentes a  $\mathfrak{R}_j^n$ . O valor de  $w$  depende do caminho seguido pela cadeia de Markov. Um mesmo valor de  $w$  pode corresponder a diferentes caminhos.
- Seja  $r_j^n(r)$  um subconjunto de  $\mathfrak{R}_j^n$ , que consiste de todas as recompensas de  $w \in \mathfrak{R}_j^n$  tais que  $w \leq r$ .

Usando as definições anteriores e considerando todos os possíveis valores que  $w$  pode tomar, tem-se que:

$$G_j(n, r) = \sum_{w \in \mathfrak{R}_j^n(r)} P[RTAD_j(n) = w] \quad (\text{IV. 2})$$

A probabilidade  $P[RTAD_j(n) = w]$  pode ser avaliada de forma recorrente no valor de  $n$ . Para  $n=0$ , suponha  $\mathfrak{R}_j^n = \{0\}$ , obtém-se então:

$$P[RTDA_j(0) = w] = \alpha_j(0)I[w = 0], \quad (\text{IV. 3})$$

onde  $\alpha_j(0)$  é a probabilidade que o sistema comece sua operação no estado  $e_j \in E$  e  $I[\text{evento}]$  é uma “função índice”, que possui o valor um se o valor lógico do evento é verdadeiro e zero em caso contrário. O conjunto de probabilidades  $\alpha_j(0)$ ,  $e_j \in E$ , ou pode ser um parâmetro de entrada do sistema ou pode ser avaliado especificamente para cada problema em particular.

Para avaliar  $P[RTAD_j(n) = w]$  para  $n \geq 1$ , condiciona-se na última transição da cadeia de Markov. Portanto:

$$P[RTDA_j(n) = w] = \sum_{e_j \in E} z_{ij} P[RTDA_i(n-1) = w - \eta(i, j)] \quad (\text{IV. 4})$$

As equações (IV.1) a (IV.5) representam uma solução eficiente para a avaliação  $G(n, r)$ , já que podem ser calculadas de forma recorrente. Essas equações podem ser usadas na avaliação de um conjunto de medidas de desempenho, como por exemplo, aquelas que permitam caracterizar o comportamento de um multiplexador durante o congestionamento.

## IV.2 Análise de Desempenho de um Multiplexador para Tráfego de Voz

### IV.2.1 Introdução

A transmissão de voz ainda é e, talvez continue a ser, o serviço mais usado para a comunicação pessoal. Sua implementação em uma rede ATM, tomada aqui como exemplo, pode ser feita usando diversas alternativas. Tal como foi explicado na seção II.4, para tornar mais eficiente o uso da banda passante do canal, uma rede pode transmitir sinais de voz usando supressão de silêncio e multiplexação estatística [ITU97].

Na Figura IV.1 apresenta-se um esquema simplificado de multiplexador usado para transmitir tráfego de voz.

Como as fontes geram células de forma assíncrona, o processo total de chegadas também é um processo assíncrono, devendo o multiplexador acomodar tal tipo de tráfego através do armazenamento temporário das células no buffer de voz, antes de sua transmissão multiplexada.

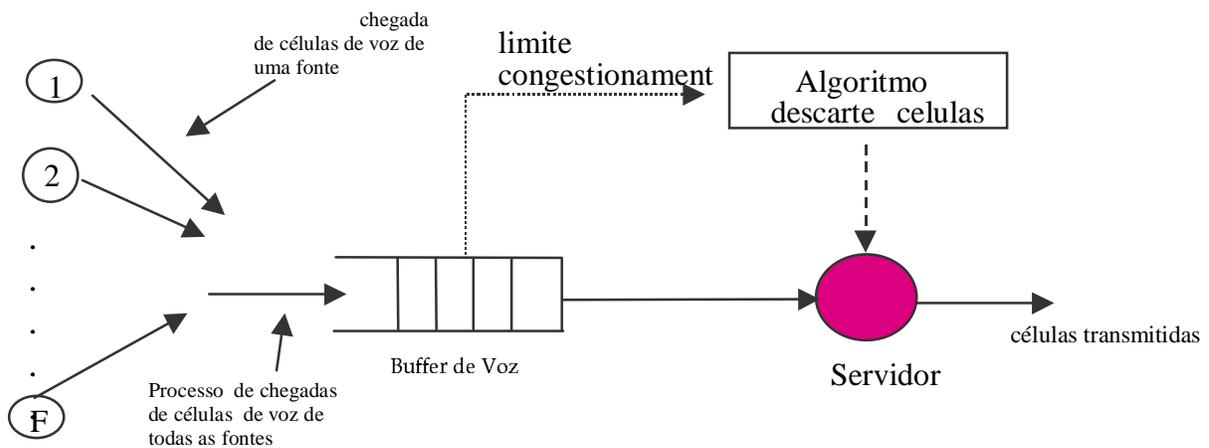


Figura IV.1. Multiplexador ATM para tráfego de voz

Um aspecto fundamental na transmissão de células de voz em uma conversação é o retardo de transferência. Este retardo deve ser menor que uns poucos milisegundos (ver II.4) para manter a interatividade de uma conversação. Por este motivo, não é possível aumentar a capacidade do buffer do multiplexador além de um certo limite, pois as células poderiam gastar um tempo excessivo na fila de transmissão, excedendo o retardo máximo permitido para cumprir a restrição de interatividade. O desempenho de um multiplexador de voz com tamanho de buffer finito é, assim, de considerável interesse, levando em conta a característica de tolerância a perdas deste tipo de tráfego.

Devido a chegada assíncrona de células ao buffer limitado, eventualmente podem ocorrer situações de congestionamento, quando várias fontes de voz coincidem no envio de células, tendo como consequência que a velocidade de chegada de células é maior que a velocidade de saída do multiplexador. Nesse caso, o número de células na memória do multiplexador aumenta progressivamente, situação que pode deteriorar a qualidade do serviço oferecido pela

rede, uma vez que aumenta os retardos de transferência, podendo até provocar a perda de informação por falta de memória de armazenamento. Pode-se então definir um período de congestionamento como aquele intervalo de tempo onde as células são retardadas além de um certo limite pré-definido, não satisfazendo assim a QoS desejável para esse tráfego. Para minorar este efeito, o multiplexador considerado possui um mecanismo de controle de congestionamento pelo descarte de células (ver seção IV.2.2) [SRI91].

Para conhecer com maior profundidade os fenômenos que acontecem em um multiplexador operando com o mecanismo de descarte de células, foi avaliado um conjunto de medidas de desempenho e seus resultados são apresentados nesta tese.

Esta tese amplia algumas das medidas de valor médio obtidas em [SRI91], no sentido de torná-las válidas para qualquer instante de tempo (na referência citada acima, os resultados foram obtidos apenas para os instantes de saída dos pacotes de voz do multiplexador). Essa restrição é irrelevante quando da avaliação da utilização média do buffer do multiplexador. Contudo, no caso da avaliação do retardo médio sofrido pelas células no multiplexador, considerar que seus instantes de chegada só ocorram nos instantes de saída de outras células pode corresponder a uma aproximação que, no mínimo, deve ser olhada com bastante cuidado. Outra contribuição desta tese é apresentar a avaliação de medidas transientes, entre elas a função de distribuição do tempo de retardo dos pacotes no multiplexador. Essa medida, de avaliação extremamente complexa, é muito importante para a determinação dos parâmetros do multiplexador, para a definição de mecanismos de controle de tráfego e congestionamento e para os mecanismos de compensação da variação estatística do retardo, que não podem ser determinados apenas pela avaliação do tempo médio.

Como somente durante os períodos de congestionamento degrada-se o comportamento do multiplexador, outro objetivo deste trabalho foi caracterizar o comportamento do multiplexador durante o período de congestionamento. Um período de congestionamento pode ser definido, em forma alternativa, como o intervalo de tempo transcorrido em que o algoritmo descarta continuamente células, como consequência da chegada de novas células. São assim avaliadas a função de distribuição do comprimento do período de congestionamento, a distribuição dos períodos em que o multiplexador não experimenta

congestionamento, e a distribuição do número de células descartadas durante os períodos de congestionamento. Também é calculado o valor médio dessas medidas, usando um novo método matemático, também contribuição desta tese.

## IV.2.2 Operação do Multiplexador ATM para Tráfego de Voz

A seguir, é dada uma descrição resumida sobre a forma como as células de voz são formadas nas fontes e sobre como opera o algoritmo de controle de congestionamento utilizando descarte de células. Para maiores detalhes deve-se reportar à referência [SRI91]. A forma de codificação do sinal de voz descrita é apenas um exemplo e não faz com que o método de avaliação perca sua generalidade. O mesmo pode-se dizer pelo fato de se usar, como exemplo, a multiplexação de células ATM. O importante apenas é que pacotes tenham tamanho fixo e tenham prioridade de perdas diferentes, como se verá.

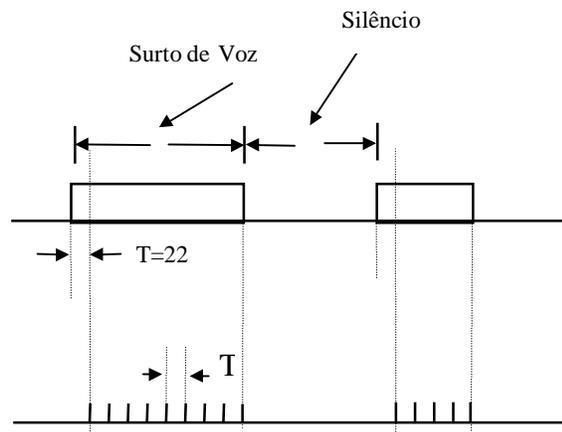


Figura IV.2. Caracterização da voz.

O primeiro passo para transmissão de voz consiste na digitalização do sinal original. No esquema que se quer analisar, cada fonte é amostrada somente durante os períodos de atividade, a uma taxa constante de 8KHz (ver Figura IV.2). As amostras de voz são codificadas utilizando ADPCM embutido - 4 bits por amostra. Estas amostras são empacotadas em pares de células ATM, cada uma das quais compõe-se de 5 bytes de cabeçalho da célula, 4 bytes de cabeçalho AAL2 [AAL297] (para que a informação de voz seja atendida como tráfego de tempo real com taxa de transmissão variável) e 44 bytes de informação (ver Figura IV.3). Para preencher um par de células ATM é necessário obter 176 amostras, as quais são arranjadas como é mostrado na Figura IV.2. Essas 176 amostras

representam 22 mseg de sinal. Os dois bits menos significativos de cada uma das 176 amostras são armazenados, seqüencialmente, em uma célula de baixa prioridade, que se etiqueta como descartável (bit CLP = 1 na célula ATM). Outra célula de alta prioridade contém os dois bits mais significativos de cada uma das 176 amostras, armazenados seqüencialmente, e é identificada como não-descartável (bit CLP=0 na célula ATM). No restante desta tese, o termo “par de células” é usado para se referir ao par de células gerado a partir das 176 amostras.

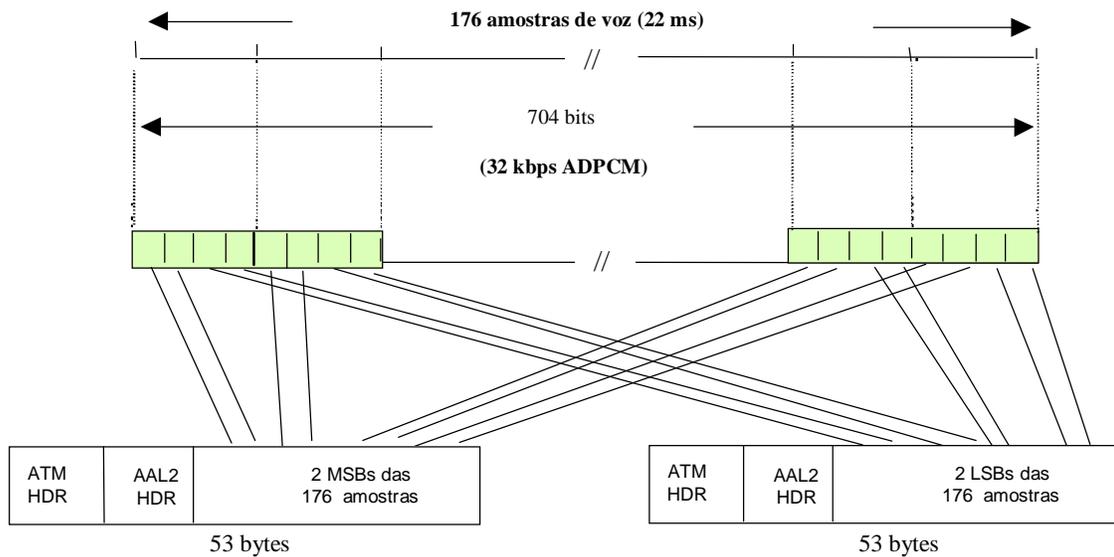


Figura IV.3. Estrutura das células de voz (figura obtida de [SRI91]).

Pelo fato do sinal de voz aceitar um certo percentual de perda, o multiplexador possui um mecanismo para controlar as situações de congestionamento baseado nessa característica. A Figura IV.4 apresenta o algoritmo de descarte de células utilizado pelo multiplexador ATM, proposto por [SRI91].

Estado do buffer ( $e_i$ )	Ação do Controle de Congestionamento
$0 \leq i \leq u$	Não há descarte
$u + 1 \leq i \leq K - 1$	Descarte da célula de baixa prioridade

Figura IV.4 Algoritmo de descarte de células.

Resumidamente, seja  $K$  o número máximo de pares de células que podem ser armazenados no buffer do multiplexador. Define-se, arbitrariamente, que o tráfego de voz está em congestionamento quando a quantidade de células presentes no buffer de voz supera um certo limiar  $u$ . Em consequência, se ao início da transmissão de um par de células há no máximo  $u$  pares de células presentes no buffer, as duas células do par são transmitidas. Em caso contrário e, se ainda houver espaço no buffer, a transmissão é acelerada, descartando-se a célula de baixa prioridade (ou seja, somente a célula de alta prioridade é transmitida). Por último, se o buffer está cheio, descarta-se qualquer par de células que chegue, enquanto a condição de buffer cheio se mantenha.

## IV.2.3 O Modelo Genérico e sua Solução Geral

### IV.2.3.1 O Modelo Genérico

O modelo do multiplexador usado nesta tese foi obtido de [SRI91], porém foi ampliado para ser usado por diferentes processos de chegada. Também, para o modelo em estudo, é apresentada uma forma mais eficiente de calcular as probabilidade em estado estacionário.

A Figura IV.5 apresenta um diagrama temporal que mostra onde são definidos os pontos embutidos, e a cadeia de Markov definida nesses pontos.

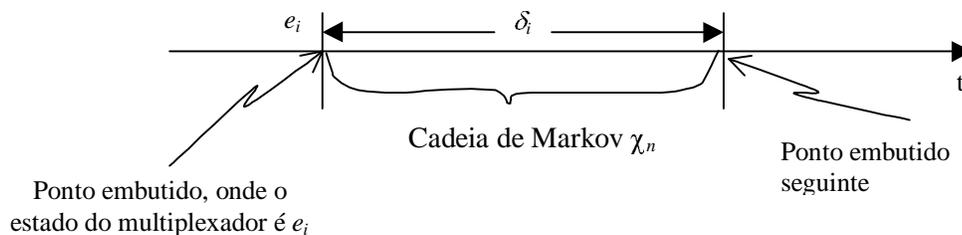


Figura IV.5. Diagrama dos pontos embutidos e cadeia Markov de tempo discreto definida nos pontos embutidos.

Seja  $e_i$  o estado do multiplexador em um determinado instante, onde  $i$  ( $0 \leq i < K$ ) é o número de pares de células armazenadas na memória, incluindo as que estão sendo transmitidas.

Define-se os instantes de término da transmissão dos pares de células como pontos embutidos no tempo. Seja  $\delta_i$  um ciclo de tipo  $i$ , que corresponde ao intervalo de tempo transcorrido a partir do ponto embutido, onde o estado do multiplexador é  $e_i$ , até o seguinte ponto embutido (ver Figura IV.5). O tempo necessário para servir um par de células (transmitir uma ou ambas células) durante  $\delta_i$  é determinístico, mas dependente da carga, devido ao algoritmo de descarte de células. Denotando por  $s_i$  esse tempo de serviço, tem-se que:

$$s_i = \begin{cases} D_2, & i = 0,1,2,\dots,u \\ D_1, & i = u+1,\dots,K-1 \end{cases} \quad (\text{IV. 5})$$

onde  $u+1$  é o número mínimo de pares de células de voz presentes no buffer para o qual se define que o multiplexador está em congestionamento.  $D_2=2D_1$  corresponde ao tempo necessário para transmitir um par de células de voz, quando não há congestionamento e,  $D_1$  o tempo requerido para transmitir somente a célula de maior prioridade, quando há congestionamento.

Seja  $\chi_n$  a cadeia de Markov de tempo discreto definida nos pontos embutidos, cujos estados correspondem ao número de pares de células armazenadas no multiplexador.

Sejam  $a_k$  e  $b_k$  as probabilidades de chegada de  $k$  pares de células de voz ao multiplexador, durante os intervalos de tamanho  $D_2$  e  $D_1$ , respectivamente.

Seja  $\mathbf{P}$  a matriz de probabilidades de transição entre os estados de  $\chi_n$ .

A matriz  $\mathbf{P}$  é então dada por :

$$\begin{array}{c} \begin{matrix} 0 & 1 & 2 & \dots & \dots & u-1 & u & u+1 & \dots & \dots & K-2 & K-1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ \vdots \\ u-1 \\ u \\ u+1 \\ \vdots \\ \vdots \\ K-2 \\ K-1 \end{matrix} \end{array} \left[ \begin{array}{cccccccccccc} a_0 & a_1 & a_2 & \cdot & \cdot & a_{u-1} & a_u & a_{u+1} & \cdot & \cdot & a_{K-2} & 1-\Sigma \\ a_0 & a_1 & a_2 & \cdot & \cdot & a_{u-1} & a_u & a_{u+1} & \cdot & \cdot & a_{K-2} & 1-\Sigma \\ 0 & a_0 & a_1 & a_2 & \cdot & \cdot & a_{u-1} & a_u & a_{u+1} & \cdot & a_{K-3} & 1-\Sigma \\ \cdot & \cdot \\ \cdot & \cdot \\ 0 & 0 & 0 & 0 & a_0 & a_1 & a_2 & a_3 & \cdot & \cdot & a_{K-u} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & a_0 & a_1 & a_2 & \cdot & \cdot & a_{K-u+1} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 & b_0 & b_1 & \cdot & \cdot & a_{K-u-2} & \cdot \\ \cdot & \cdot \\ \cdot & 1-\Sigma \\ \cdot & b_0 & b_1 & 1-\Sigma \\ \cdot & 0 & b_0 & 1-\Sigma \end{array} \right]$$

Uma vez que  $\mathbf{P}$  é uma matriz estocástica, os termos  $(1-\Sigma)$ , na última coluna, correspondem a 1 menos a soma de todos os outros termos da linha.

Cabe salientar que esta matriz  $\mathbf{P}$  é geral. Isto é, as probabilidades  $a_k$  e  $b_k$  podem tomar valores diferentes dependendo do modelo de chegadas de células usado (por exemplo: MMPP [SCH96], modelo de fluidos [SCH96], Poisson). Nessa tese, foi considerado que o processo de chegada de células ao multiplexador é um processo de Poisson, dado que este modelo é adequado ao tipo de multiplexador usado [SRI91]. Esta instanciação do modelo geral é apresentada na Seção IV.3.3.

#### IV.2.3.2 Cálculo do autovetor $\pi$

Seja  $\pi = \langle \pi_0, \pi_1, \pi_2, \dots, \pi_{K-1} \rangle$  o autovetor de  $\mathbf{P}$ . O cálculo de  $\pi$  pode ser realizado, como é usual, resolvendo-se a equação:

$$\pi = \pi \mathbf{P} \quad (\text{IV. 6})$$

Entretanto, como as transições de  $\mathbf{P}$  a partir de um determinado estado  $e_i$ ,  $1 \leq i < K$  só podem ocorrer para estados  $e_j$  com  $j \geq i-1$ ,  $\pi$  pode ser calculado de forma mais eficiente que a tradicional multiplicação de matrizes [ROM90].

A seguir é apresentada uma outra contribuição desta tese (de menor peso), que é uma forma mais eficiente de calcular  $\pi$ , para o tipo de matriz em análise.

Seja  $X_i$  um valor não normalizado de  $\pi_i$ ,  $i=0,1,2,3,\dots,K-1$ ; isto é  $\sum_{i=0}^{K-1} X_i \neq 1$  e  $\frac{X_i}{X_j} = \frac{\pi_i}{\pi_j}$ .

Supõe-se, de forma arbitrária, que:

$$X_0 = 1 \quad (\text{IV. 7})$$

Aplicando a equação de balanceamento global ao estado  $e_0$ , mas usando  $X_0$  e  $X_1$  ao invés de  $\pi_0$  e  $\pi_1$  respectivamente, tem-se que:

$$X_1 = X_0 \frac{(1-a_0)}{a_0} \quad (\text{IV. 8})$$

Analogamente, podem ser obtidas equações recorrentes para o cálculo de  $X_{i+1}$ ,  $2 \leq i+1 \leq K-1$ , aplicando as equações de balanceamento global para os estados  $e_i$ ,  $1 \leq i \leq K-2$ . Então:

$$X_{i+1} = \frac{1}{a_0} \left\{ X_i (1 - a_1) - X_0 a_i - \sum_{k=1}^{i-1} X_k a_{i-k+1} \right\} \quad (\text{IV. 9})$$

$$X_{u+1} = \frac{1}{b_0} \left\{ X_u (1 - a_1) - X_0 a_u - \sum_{k=1}^{u-1} X_k a_{u-k+1} \right\} \quad (\text{IV. 10})$$

$$X_{i+1} = \frac{1}{b_0} \left\{ X_i (1 - b_1) - X_0 a_i - \sum_{k=1}^u X_k a_{i-k+1} - \sum_{k=u+1}^{i-1} X_k b_{i-k+1} \right\} \quad (\text{IV. 11})$$

Seja  $S = \sum_{i=0}^{K-1} X_i$ , pode-se provar que:

$$\pi_i = \frac{X_i}{S}, \quad 0 \leq i \leq K \quad (\text{IV. 12})$$

Assim, o conjunto de equações (IV.7) a (IV.13) permite avaliar  $\pi$  de forma recorrente e, portanto, mais eficiente que a tradicional multiplicação matricial.

### IV.2.3.3 Instanciação do Modelo Usando um Processo de Poisson para o Processo de Chegadas.

Com objetivo de aclarar e para dar base às seções subseqüentes, repete-se nesta subseção o modelo e as análises básicas realizadas em [SRI91].

#### *IV.2.3.3.1 Por que utilizar para o processo de chegadas um Processo de Poisson?*

Em um multiplexador que não usa um método de controle de congestionamento por descarte de células, a chegada de células provenientes da superposição de várias fontes de voz, pode resultar, no caso de alta utilização, que uma quantidade significativa dessas células tenha que ser armazenada na fila do multiplexador. Nesse caso as correlações entre os intervalos de chegada de cada fonte de voz influencia o comportamento da fila, dando como resultados retardos elevados [HEF86, SRI89]. Se, no entanto, é usado um mecanismo de controle de congestionamento pelo descarte de células, o comprimento da fila não adquire grandes tamanhos. Isto acontece porque o mecanismo de descarte de células acelera o serviço nos momentos críticos, impedindo a formação de filas grandes. Com poucas células interagindo

na fila (em qualquer tempo), o intervalo entre chegadas com distribuição exponencial é o que controla o comportamento e o efeito das correlações é insignificante. Como consequência, se há poucas células interagindo na fila, o comportamento tende a imitar o comportamento de uma fila com chegadas de Poisson. Em [SRI86,SRI91] pode ser encontrada uma demonstração formal de que quando é usado o mecanismo de descarte de células, como o descrito na Seção IV.2.2, o processo de chegada de células ao multiplexador, formado pela superposição de células, comporta-se como um processo de Poisson.

Assim, nesta tese, o processo total de chegada de pares de células ao multiplexador foi modelado como um processo de Poisson de parâmetro  $\lambda$ . Para o cálculo deste parâmetro, supõe-se que todas as fontes de voz têm as mesmas características e são independentes entre si. As células são geradas em cada uma das fontes a uma taxa média  $\lambda_f$ , onde  $f$  é uma fonte específica, sendo  $\lambda_1 = \lambda_2 = \lambda_3 = \dots = \lambda_f$ . Logo, a taxa média de chegadas de células de todas as fontes de voz ao multiplexador é dada por  $\lambda = F\lambda_f$ , onde  $F$  é o número total de fontes de voz (ver Figura IV.1).

Como consequência do anterior, um modelo  $M / \tilde{D} / 1 / K$  aproxima o comportamento da fila do multiplexador de maneira bastante exata. Sendo que  $\tilde{D}$  representa o serviço determinístico e dependente do tempo, dado pela equação (IV.6).

Sejam  $a_k$  e  $b_k$  as probabilidades de chegada de  $k$  pares de células de voz ao multiplexador, nos intervalos de tamanho  $D_2$  e  $D_1$ , respectivamente. Devido a suposição que o processo de chegadas tem uma distribuição Poisson, conclui-se que:

$$\begin{aligned} a_k &= \frac{e^{-\lambda D_2} (\lambda D_2)^k}{k!}; k \geq 0 \\ b_k &= \frac{e^{-\lambda D_1} (\lambda D_1)^k}{k!}; k \geq 0 \end{aligned} \tag{IV. 13}$$

Usando o método apresentado em IV.2.3.2, não foram achadas nem dificuldades numéricas nem computacionais para calcular o vetor de probabilidades em estado estacionário  $\pi$ , para os valores de tamanho de buffers de interesse do multiplexador em estudo. Para o cálculo de  $\pi$  foi usada a matriz  $\mathbf{P}$  instanciada com as probabilidades de transição dadas por (IV.14).

## IV.2.4 Avaliação de Medidas de Desempenho

### IV.2.4.1 Número Médio de Pares de Células no Multiplexador

Seja  $N$  o número médio de pares de células de voz no multiplexador;  $N_i$  o número médio de células no multiplexador durante um ciclo  $\delta_i$ ; e  $\omega_i$  a probabilidade em estado estacionário de se estar em um ciclo  $\delta_i$ . Então:

$$N = \sum_{i=0}^{K-1} \omega_i N_i \quad (\text{IV. 14})$$

Pela teoria da renovação [ROS93], observa-se que:

$$\omega_i = \frac{\pi_i B_i}{\sum_{j=0}^{K-1} \pi_j B_j} \quad (\text{IV. 15})$$

onde  $B_i$  é a duração média de um ciclo  $\delta_i$ .

Para o cálculo de  $B_i$  devem ser distinguidos dois casos. O primeiro corresponde a  $B_0$ , quando não existem pacotes ao começo do ciclo. Nesse caso,  $B_0$  tem uma duração igual ao tempo médio de chegada de um pacote ao multiplexador, mais o tempo de serviço do primeiro pacote, isto é:  $B_0 = \frac{1}{\lambda} + D_2$ . No segundo caso, o multiplexador se encontra em algum estado  $e_i$ ,  $i > 0$ , no início do ciclo  $B_i$ . Nesse caso,  $B_i$  depende do tempo de serviço da célula, como é dado pela equação (IV.6). Resumindo ambos os casos, chega-se a:

$$B_i = \begin{cases} \frac{1}{\lambda} + D_2, & i = 0 \\ D_2, & 1 \leq i \leq u \\ D_1, & u + 1 < i < K \end{cases} \quad (\text{IV. 16})$$

O cálculo de  $N_i$  requer um desenvolvimento mais elaborado. Para tanto, é atribuída ao multiplexador um ganho de recompensa igual a  $i$  quando se encontra no estado  $e_i$ . A idéia básica é que o multiplexador ganhe uma recompensa, a uma velocidade unitária, para cada um de seus pares de células. Isto significa que, quando o multiplexador tiver  $i$  pares de células armazenadas, incluindo aquele em transmissão, a cada unidade de tempo que passar nesse estado, ganhará uma recompensa igual a  $i$ . Essa interpretação permite calcular a recompensa total ganha pelo multiplexador, somando-se as contribuições dadas por cada um dos seus pares de células.

Com a atribuição de recompensa assinalada,  $N_i$  corresponde à recompensa média ganha durante um ciclo  $\delta_i$ , dividida pelo tamanho médio desse ciclo. Mais precisamente, se  $R_i$  for definida como a recompensa média acumulada pelo multiplexador em um ciclo  $\delta_i$ , obtém-se que:

$$N_i = \frac{R_i}{B_i}; \quad 0 \leq i < K \quad (\text{IV. 17})$$

Para calcular  $R_i$ , condiciona-se no número de chegadas enquanto se transmite um par de células no ciclo  $\delta_i$ . Seja  $R_{ij}$  a recompensa média acumulada no ciclo  $\delta_i$ , no caso em que chegam  $j$  pares de células; e  $Z_i(j)$  corresponde à probabilidade que cheguem  $j$  pares de células enquanto é transmitido um par de células no ciclo  $\delta_i$ . Assim, observa-se que:

$$R_i = \sum_{j=0}^{K-i-1} Z_i(j)R_{ij} + \sum_{j=K-i}^{\infty} Z_i(j)R_{ij}; \quad 0 \leq i < K \quad (\text{IV. 18})$$

Usando o fato que o processo de chegadas tem uma distribuição Poisson com parâmetro  $\lambda$ , a equação (IV.19) pode ser escrita como:

$$R_i = \sum_{j=0}^{K-i-1} e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} R_{ij} + \sum_{j=K-i}^{\infty} e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} R_{ij}; \quad 0 \leq i < K \quad (\text{IV. 19})$$

onde  $s_i$  é dado pela equação (IV.6).

No primeiro somatório da equação (IV.20) estão agrupados os casos em que a chegada de  $j$  pares de células não é suficiente para esgotar a capacidade de armazenamento do multiplexador. O segundo somatório corresponde a todos os casos para os quais o multiplexador esgota sua capacidade de armazenamento durante o ciclo  $\delta_i$ . Esta separação permite calcular de um modo mais simples  $R_{ij}$ , como se descreve a seguir.

O valor de  $R_{ij}$ , no primeiro somatório de (IV.20), corresponde à área abaixo da curva na Figura IV.6. Como é mostrado na Figura IV.11, a recompensa total acumulada no ciclo  $\delta_i$  tem dois componentes: um devido ao número de pares de células no multiplexador no começo do ciclo, e outro dependente do número de pacotes que chegam ao multiplexador durante  $s_i$ . Para  $i > 0$ , a recompensa total acumulada devido ao primeiro componente é  $is_i$ . Para calcular a recompensa total acumulada graças ao segundo componente, usa-se o fato que a chegada de pares de células tem uma distribuição Poisson. Para simplificar a análise matemática, pode ser usado o teorema que estabelece que: “os instantes de chegada  $j$  eventos, ocorridos com distribuição Poisson em um intervalo  $s_i$ , têm a mesma distribuição conjunta que a estatística

de ordem de  $j$  eventos independentes uniformemente distribuídos no mesmo intervalo” [KAR81]. Então, a recompensa ganha pelos  $j$  pares de células que chegam durante  $s_i$  corresponde à soma do tempo médio de permanência no ciclo  $\delta_i$  de cada um dos  $j$  pares de células. Além do mais, pelo teorema anteriormente enunciado, o processo de chegada de pares de células de voz tem uma distribuição uniforme e independente, pode ser visto que o tempo médio de permanência de cada par de célula durante o ciclo  $\delta_i$  é igual à metade do intervalo.

Isto implica que a recompensa média ganha por  $j$  pares de células que chegam em  $s_i$  é  $j \frac{s_i}{2}$ .

Finalmente, somando-se os dois componentes, obtém-se:

$$R_{ij} = is_i + j \frac{s_i}{2}; \quad i > 0, (i+j) < K \quad (\text{IV. 20})$$

Para calcular  $R_{ij}$  do segundo somatório de (IV.20), com  $i > 0$ , considera-se o fato que ocorreram  $j$  chegadas durante o intervalo  $[0, s_i]$ ,  $(i+j) \geq K$ , como é mostrado na Figura IV.7. Nesse caso, em primeiro lugar, nota-se que são perdidos os pares de células que chegam ao multiplexador no ciclo  $\delta_i$ , quando a memória está saturada. Seja então a variável aleatória  $\tau$  o instante em que a memória satura. Considerando o começo do ciclo  $\delta_i$  com o tempo 0, seja  $A'_{ij}(\tau)$  a recompensa total ganha pelo multiplexador no intervalo  $[0, \tau)$  do ciclo  $\delta_i$  e  $A''_{ij}(\tau)$  a recompensa total ganha pelo multiplexador no intervalo  $[\tau, s_i]$  do mesmo ciclo (veja Figura IV.7). Aplicando o teorema de probabilidades totais, tem-se que a recompensa total acumulada pelo multiplexador durante o ciclo  $\delta_i$  é:

$$R_{ij} = \int_{\tau=0}^{\infty} (A'_{ij}(\tau) + A''_{ij}(\tau)) f(\tau) d\tau, \quad i > 0, i+j \geq K \quad (\text{IV. 21})$$

onde  $f(\tau)$  corresponde a função de densidade de probabilidade de  $\tau$ .

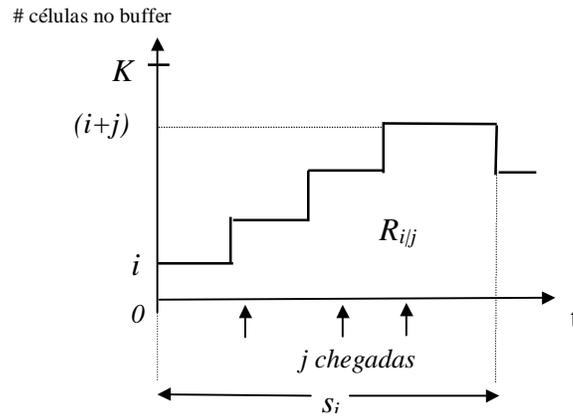


Figura IV. 6. Recompensa acumulada no ciclo  $\delta_i$  ( $i > 0$ ) quando a memória do multiplexador não está saturada.

Para calcular  $A'_{ij}(\tau)$ , procede-se em forma análoga ao caso em que a memória não está saturada. A diferença é que aqui se deve substituir  $s_i$ , da equação (IV.21), por  $\tau$ , uma vez que este é o tempo que a memória demora para saturar. Portanto, notando que chegam  $K-i-1$  pares de células no intervalo  $[0, \tau)$ , chega-se a:

$$A'_{ij}(\tau) = \frac{K-1+i}{2} \tau; \quad i > 0, i+j \geq K \quad (\text{IV. 22})$$

Por sua vez,  $A''_{ij}(\tau)$  corresponde, simplesmente, a recompensa ganha pelo multiplexador devido à permanência de  $K$  pares de células durante o intervalo  $[\tau, s_i]$ . Assim:

$$A''_{ij}(\tau) = K(s_i - \tau); \quad i > 0, i+j \geq K \quad (\text{IV. 23})$$

Substituindo os resultados de (IV.23) e (IV.24) em (IV.22):

$$R_{ij} = \int_{\tau=0}^{\infty} \left\{ \frac{K-1+i}{2} \tau + K(s_i - \tau) \right\} \cdot f(\tau) d\tau; \quad i > 0 \quad (\text{IV. 24})$$

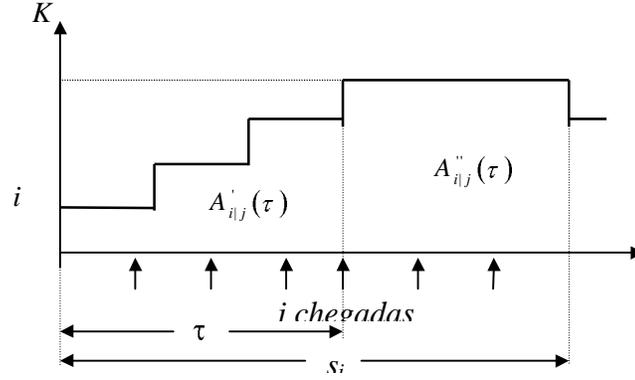


Figura IV.7 : Recompensa acumulada no ciclo  $\delta_i$  ( $i > 0$ ), quando a memória do multiplexador está saturada.

Para calcular  $f(\tau)$ , pode ser usado o fato que  $\tau$  corresponde à  $(K-i)$ -ésima estatística de ordem de  $j$  variáveis aleatórias independentes e identicamente distribuídas no intervalo  $[0, s_i]$  [KAR81]. Portanto:

$$f(\tau) = \binom{j}{K-i-1} \left[ \frac{\tau}{s_i} \right]^{K-i-1} (j-K+i+1) \frac{1}{s_i} \left[ \frac{s_i - \tau}{s_i} \right]^{j-K+i} \quad (\text{IV. 25})$$

Substituindo (IV.26) em (IV.25):

$$R_{ij} = s_i^{-j} (j-K+i+1) \int_{\tau=0}^{s_i} \left\{ \frac{K-1+i}{2} \tau + K(s_i - \tau) \right\} \binom{j}{K-i-1} \tau^{K-i-1} (s_i - \tau)^{j-K+i} d\tau$$

Aplicando o teorema do binômio à expressão  $(s_i - \tau)^{j-K+i}$ , trocando a ordem do somatório com a integral e simplificando, obtém-se:

$$R_{ij} = s_i \binom{j}{K-i+1} (j-K+i+1) \sum_{l=0}^{j-K+i} \binom{K-i+1}{j} (-1)^l \left\{ \frac{K}{l+(K-i)} - \frac{\frac{(K-i+1)}{2}}{l+(K-i+1)} \right\} \quad (\text{IV. 26})$$

Usando em (IV.27) o fato que:  $\sum_{l=0}^m \frac{\binom{m}{l} (-1)^l}{(l+L)} = \left\{ L \binom{L+m}{L} \right\}^{-1}$  e simplificando, chega-se a:

$$R_{ij} = s_i \binom{K-i}{j+1} \cdot \left( \frac{K-1+i}{2} \right) + s_i \left( \frac{j-K+i+1}{j+1} \right) K \quad (\text{IV. 27})$$

Considerando que a forma de  $A'_{ij}(\tau)$  é retangular (ver Figura IV.7), da equação (IV.28) se deduz que o valor médio da base é dado por  $s_i \left( \frac{(K-i)}{(j+1)} \right)$ , e que o valor médio da altura é dado

por  $\frac{\binom{m}{j}^{-1}}{\sum_{i=0}^{m-1} \binom{m}{i}^{-1}}$ . Analogamente,  $X_{ni} = \frac{1}{b} \left\{ X_{b+1-i} - X_{nb} - \sum_{k=1}^{ni} a_{kni} \right\}$  corresponde ao tamanho médio da base de  $A''_{i|j}(\tau)$ , sendo  $K$  a altura média de  $A''_{i|j}(\tau)$ .

Assim, para obter  $R_i$ ,  $i > 0$ , os resultados obtidos em (IV.21) e (IV.28) são substituídos em (IV.20):

$$R_i = \sum_{j=0}^{K-i-1} \left[ i s_i + j \frac{s_i}{2} \right] e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} + \sum_{j=K-i}^{\infty} s_i \left[ \left( \frac{K-i}{j+1} \right) \left( \frac{K-1+i}{2} \right) + K \left( \frac{j-K+i+1}{j+1} \right) \right] e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} \quad (\text{IV. 28})$$

Resta somente obter  $R_0$ , para isto note-se que, no ciclo  $\delta_0$ , o multiplexador somente começa a ganhar recompensas quando chega o primeiro par de células. A partir desse instante, o comportamento passa a ser igual ao do início do ciclo  $\delta_i$ . Portanto, como  $u \geq 1$ , então  $R_0 = R_1$ . Finalmente, substituindo (IV.17) e (IV.29) em (IV.18), e posteriormente (IV.18) em (IV.15), conclui-se que o número médio de pares de células no multiplexador,  $N$ , é:

$$N = \omega_0 \frac{D_2}{D_2 + \frac{1}{\lambda}} \left\{ \sum_{j=0}^{K-2} \left[ 1 + \frac{j}{2} \right] e^{-\lambda D_2} \frac{[\lambda D_2]^j}{j!} + \sum_{j=K-1}^{\infty} \left[ \left( \frac{K-1}{j+1} \right) \frac{K}{2} + K \left( \frac{j-K+2}{j+1} \right) \right] e^{-\lambda D_2} \frac{[\lambda D_2]^j}{j!} \right\} + \sum_{i=1}^{K-1} \omega_i \left\{ \sum_{j=0}^{K-i-1} \left[ i + \frac{j}{2} \right] e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} + \sum_{j=K-i}^{\infty} \left[ \left( \frac{K-i}{j+1} \right) \left( \frac{K-1+i}{2} \right) + K \left( \frac{j-K+i+1}{j+1} \right) \right] e^{-\lambda s_i} \frac{[\lambda s_i]^j}{j!} \right\} \quad (\text{IV. 29})$$

onde  $\omega_i$ ,  $0 \leq i < K$ , é dado pela equação (IV.16).

#### IV.2.4.2 Cálculo do Tempo Médio de Permanência no Multiplexador, Tempo Médio na Fila e Número Médio de Pares de Células na Fila do Multiplexador.

Como a memória do multiplexador é finita, nem todos os pares de célula que chegam são enfileirados. A vazão do sistema é dado por  $\mu(1-p_0)$ , onde  $p_0$  é a probabilidade que o multiplexador esteja vazio em um tempo qualquer e  $\mu$  é a taxa média de transmissão quando o multiplexador não está ocioso.

Para calcular  $p_0$ , note que os instantes durante o qual o sistema está vazio são pontos de renovação do seu comportamento estocástico. Partindo desde um ponto de renovação, o sistema permanece algum tempo vazio, denominado período vazio. A partir do primeiro par

de células que chega, o sistema permanece ocupado, denominado período ocupado, onde passa a atender todos os pares de células que chegam até ficar vazio novamente. Seja  $I$  o valor médio do período vazio e seja  $B$  o valor médio de um período ocupado. Pela teoria da renovação tem-se que:  $p_0 = \frac{I}{I+B}$ . Como a chegada de pares de células é caracterizada por

um processo de Poisson com taxa  $\lambda$ , tem-se que  $I = \frac{1}{\lambda}$ . Por outro lado, usando um argumento

estacionário (ver apêndice A), obtém-se que  $B = \left(\frac{1}{\pi_0}\right)\left(\frac{1}{\mu}\right)$ , onde  $\frac{1}{\pi_0}$  corresponde ao número

médio de pares de células transmitidas durante um período ocupado e  $\frac{1}{\mu}$  é o tempo médio

de transmissão dos pares de células. Por definição:  $\left(\frac{1}{\mu}\right) = \sum_{i=0}^{K-1} \pi_i s_i$ . Então:

$$\mu = \left[ D_2 \sum_{i=0}^u \pi_i + D_1 \sum_{i=u+1}^{K-1} \pi_i \right]^{-1} \quad (\text{IV. 30})$$

Logo, substituindo  $\frac{1}{\mu}$  em  $B$ , e depois os valores de  $I$  e  $B$  em  $p_0$ , obtém-se que:

$$p_0 = \frac{\pi_0}{\pi_0 + \rho} \quad (\text{IV. 31})$$

onde  $\rho = \frac{\lambda}{\mu}$ .

Seja  $T$  o tempo médio de permanência dos pares de células no multiplexador. Usando a Lei de Little [KLE78] obtém-se:

$$T = \frac{N}{\mu(1-p_0)} \quad (\text{IV. 32})$$

Substituindo (IV.31) e (IV.32) em (IV.33), conclui-se que  $T$  é dado por:

$$T = \left( \frac{\pi_0 + \rho}{\rho} \right) N \left[ D_2 \sum_{i=0}^u \pi_i + D_1 \sum_{i=u+1}^{K-1} \pi_i \right] \quad (\text{IV. 33})$$

Considerando  $W$  como o tempo médio que um par de células gasta na fila do multiplexador, então:

$$W = T - \frac{1}{\mu} \quad (\text{IV. 34})$$

Usando (IV.31) e (IV.32) em (IV.33), é obtido:

$$W = \left[ D_2 \sum_{i=0}^u \pi_i + D_1 \sum_{i=u+1}^{K-1} \pi_i \right] \left( \frac{\pi_0 + \rho}{\rho} N - 1 \right) \quad (\text{IV. 35})$$

Seja  $L_q$  o número médio de pacotes na fila. Aplicando a Lei de Little  $L_q = \mu(1 - p_0)W$ . Então, usando (IV.30), (IV.32) e (IV.36), tem-se que:

$$L_q = N - \frac{\rho}{\pi_0 + \rho} \quad (\text{IV. 36})$$

#### **IV.2.4.3 Avaliação de Medidas de Desempenho que Caracterizam um Período de Congestionamento**

Usando os conceitos da Seção IV.1.1, nesta sub-seção é avaliado um conjunto de novas medidas transientes que permitem caracterizar o comportamento do multiplexador durante os períodos de congestionamento. Essas medidas são a Função de Distribuição Cumulativa (FDC) do tempo de residência de um par de células no multiplexador; a FDC do comprimento de um período de congestionamento; a FDC das células descartadas durante o período de congestionamento; a FDC do número de células perdidas durante o período de congestionamento e a FDC do comprimento de um período sem congestionamento.

##### ***IV.2.4.3.1 FDC do Tempo de Residência de um Par de Células no Multiplexador***

Um dos aspectos críticos na operação das redes com QoS é a capacidade de satisfazer certos requisitos de tempo real. Um destes requerimentos é o retardo de transferência de células. Para avaliar o retardo completo, não é suficiente apenas avaliar o tempo de residência médio das células no multiplexador, também é necessário conhecer a função de distribuição deste tempo. Quando o multiplexador opera com a política de descarte de células, a avaliação dessa distribuição é uma tarefa difícil, porque o tempo de residência de um determinado par de células depende não somente dos pares de células que estão no multiplexador antes que dito

par chegue, mas também dos pares de células que chegam depois que ele chegou, mas antes de sua transmissão.

A partir de agora, para efeito de análise, o par de célula sob estudo será chamado de *par de células marcado, M*.

Seja  $t$  a variável aleatória que corresponde ao tempo em que  $M$  permanece no multiplexador. Suponha que o multiplexador sirva sua fila usando uma política FIFO. Assim, o valor de  $t$  é dado por:

$$t = \xi + \phi \quad (\text{IV. 37})$$

onde  $\xi$  é igual ao tempo necessário para atender a todos os pares de células que esperam na memória do multiplexador quando  $M$  chega, incluindo o tempo de serviço de  $M$ ; e  $\phi$  é o tempo residual do serviço do par de células que está sendo atendido no momento em que  $M$  chega.

De (IV.38), pode ser deduzido que a variável aleatória  $t$  é limitada por  $\xi \leq t \leq \xi + D_2$ , dado que o valor da variável  $\phi$  varia entre 0 e  $D_2$ . Desta forma, a FDC de  $t$ ,  $P[t \leq r]$ , pode ser limitada por:

$$P[\xi \leq r - D_2] \leq P[t \leq r] \leq P[\xi \leq r],$$

onde  $P[\xi \leq r]$  é a FDC da variável aleatória  $\xi$ .

Para calcular  $P[\xi \leq r]$ , o comportamento da fila do multiplexador nos pontos embutidos é representado pela cadeia de Markov  $\mathbf{Z}_r$ , dada na Figura IV.8.

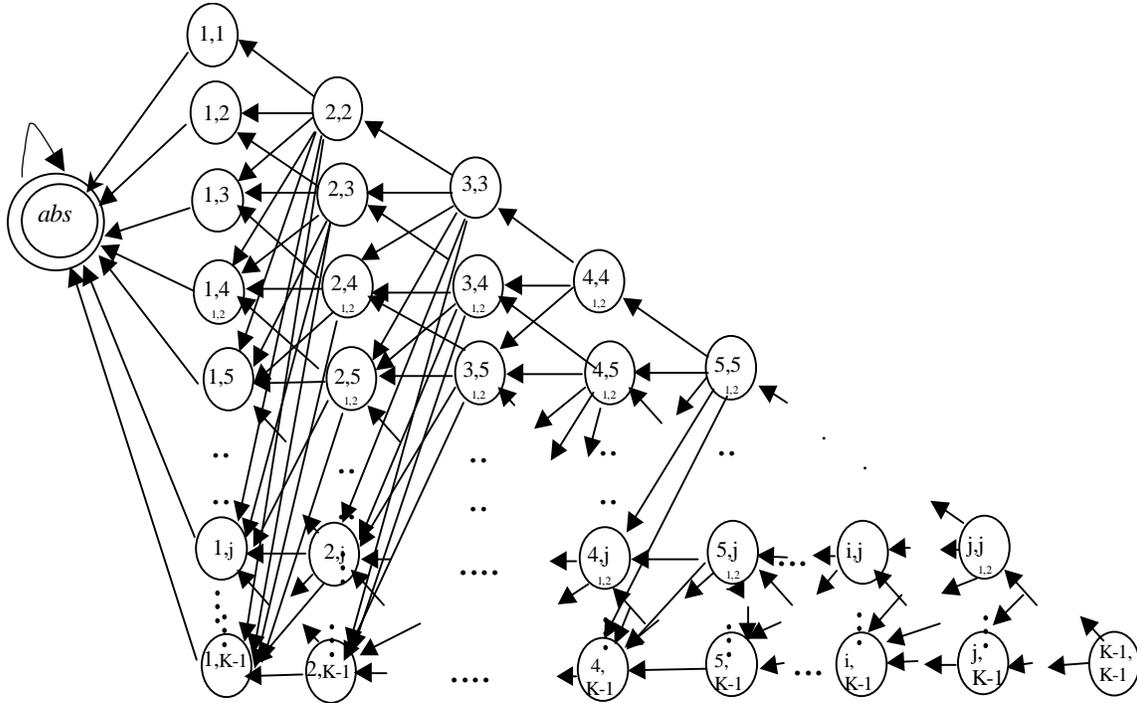


Figura IV.8. Cadeia de  $\mathbf{Z}_r$  usada para modelar o tempo de residência no multiplexador

O estado  $(i,j)$  desta cadeia de Markov significa que, ao término da transmissão de um par de células (ponto embutido), o comprimento da fila é de  $j$  pares de células e a posição de  $M$  é  $i$ . O estado “*abs*” é um estado absorvente, usado para modelar o fim da transmissão de  $M$ . A matriz de probabilidade de transição de  $\mathbf{Z}_r$  é calculado como é apresentado na Figura IV.9, onde as probabilidades  $a_k$  e  $b_k$  são dadas por (IV.2).

Para modelar o tempo de residência das células marcadas, associa-se, adicionalmente, uma recompensa específica a cada uma das transições da cadeia de  $\mathbf{Z}_r$ . A idéia básica é associar, a cada uma das transições de saída de um certo estado, uma recompensa igual ao tempo que o sistema permanece no estado. O sistema começa ganhando recompensa no momento que  $M$  chega, e pára de ganhar recompensa quando este par de células for transmitido (nesse caso o sistema cai para o estado absorvente).

Estado Inicial	Estado final	Probabilidade	Limites
$(abs)$	$(abs)$	$1$	
$(1,j)$	$(abs)$	$1$	$1 \leq j \leq K-1$
$(i,j)$	$(i-1, j-1)$	$a_0$	$1 < i \leq j \leq u$
$(i,j)$	$(i-1, j-1)$	$b_0$	$u < j \leq K-1$ $1 < i \leq j$
$(i,j)$	$(i-1, k)$	$a_{k-j+1}$	$1 < i \leq j \leq u$ $j \leq k < K-1$
$(i,j)$	$(i-1, K-1)$	$1 - \sum_{n=0}^{K-1-j} a_n$	$1 < i \leq j \leq U$
$(i,j)$	$(i-1, k)$	$b_{k-j+1}$	$u < j \leq K-1$ $1 < i \leq j, j \leq k < K-1$
$(i,j)$	$(i-1, K-1)$	$1 - \sum_{n=0}^{K-1-j} b_n$	$u < j \leq K-1$ $1 < i \leq j$

Figura IV.9. Probabilidades de transição da cadeia de Markov que modela o tempo de residência do par de células no multiplexador de voz

Como consequência dessa modelagem, a avaliação da função de distribuição cumulativa do tempo de residência do par de células é equivalente a avaliação da função de distribuição da recompensa que o multiplexador já ganhou.

Mais precisamente, a função  $\eta_r(i, j)$ , que associa recompensas às transições de saída do estado  $(i, j)$ ,  $1 \leq i \leq j < K$ , a qualquer outro estado, é definida como:

$$\eta_r(i, j) = \begin{cases} D_2, & j = 0, 1, 2, \dots, u; & 1 \leq i \leq j \\ D_1, & j = u + 1, \dots, K - 1; & 1 \leq i \leq j \\ 0 & \text{outros casos} \end{cases} \quad (\text{IV. 38})$$

Para avaliar a função de distribuição cumulativa do tempo de residência no multiplexador do par de células marcadas, aplica-se o procedimento apresentado na Seção IV.1.1. A seguir e,

para maior clareza, é deduzida, novamente, parte da Seção IV.1.1., mas, nesse caso, para um estado bidimensional. Para tanto, define-se  $G(r)$  como a função de distribuição cumulativa do tempo de residência:

$$G(r) = P[\xi \leq r] \quad (\text{IV. 39})$$

onde  $\xi$ , definido em (IV.38), é uma variável aleatória que corresponde ao tempo de residência de um par de células no multiplexador.

Como o tempo de residência do par de células no multiplexador termina quando o sistema entra no estado  $abs$  e como o número máximo de transições que  $M$  pode realizar até atingir o estado absorvente é  $K-1$  (ver Figura IV.9), tem-se que:

$$G(r) = G_{abs}(K-1, r) \quad (\text{IV. 40})$$

A função  $G_{abs}$  é avaliada após  $K-1$  transições, que é o máximo número de transições que o sistema suporta antes de cair no estado absorvente  $abs$  (a partir desse momento, a recompensa que o sistema ganha permanece constante), independentemente do estado em que se encontra na chegada do par de células marcadas.

Para avaliar  $G_{abs}(K-1, r)$ , deve ser avaliado  $P[RTAD_{abs}(K-1) = w]$  para diferentes valores de  $w$ . Isto pode ser feito através de um procedimento recorrente, como se segue:

$$P[RTDA_{(i,j)}(n) = w] = \sum_{(k,l)} z_{r(k,l)(i,j)} P[RTDA_{(k,l)}(n-1) = w - \eta_r(k,l)] \quad (\text{IV. 41})$$

onde a condição inicial ( $n=0$ ) é dada por  $\mathfrak{R}_{(i,j)}^0 = \{0\}$ . Dessa forma, tem-se que:

$$P[RTAD_{(i,j)}(0) = w] = \alpha_{(i,j)}(0) I[w = 0] \quad (\text{IV. 42})$$

onde  $\alpha_{(i,j)}(0)$  é a probabilidade do sistema entrar no estado  $(i,j)$  quando chega o par de células marcadas.

Para resolver  $G(r)$  é necessário avaliar somente  $\alpha_{(i,j)}(0)$ . A avaliação exata de  $\alpha_{(i,j)}(0)$  não é fácil, principalmente devido ao tempo residual do par de células que está sendo transmitido quando  $M$  chega. Para superar esta dificuldade, avalia-se uma aproximação de  $\alpha_{(i,j)}(0)$ . Note que durante o serviço de um par de células, a probabilidade que somente chegue um par de células é maior que a probabilidade que cheguem dois ou mais pares (porque o tempo de transmissão de um par é menor que o tempo entre chegadas de pares de células). Supondo-se

então que quando chega um par de células não chega nenhum outro par no mesmo intervalo, obtém-se que:

$$\alpha_{(i,j)}(0) \approx \begin{cases} \pi_0 + \pi_1, & i = j = 1 \\ \pi_i, & i = j > 1 \\ 0, & i \neq j \end{cases} \quad (\text{IV. 43})$$

As equações (IV.39) a (IV.44) representam uma solução recorrente para calcular  $G(r)$ .

#### IV.2.4.3.2 FDC do Comprimento de um Período de Congestionamento

$$\begin{array}{c} \begin{array}{c} und \\ u+1 \\ u+2 \\ \vdots \\ K-2 \\ K-1 \end{array} \left[ \begin{array}{ccccccc} \begin{array}{c} und \\ u+1 \\ u+2 \\ \vdots \\ K-2 \\ K-1 \end{array} & \begin{array}{c} und \\ u+1 \\ u+2 \\ \vdots \\ K-2 \\ K-1 \end{array} & \begin{array}{c} und \\ u+1 \\ u+2 \\ \vdots \\ K-2 \\ K-1 \end{array} & \begin{array}{c} und \\ u+1 \\ u+2 \\ \vdots \\ K-2 \\ K-1 \end{array} & \dots & \begin{array}{c} und \\ u+1 \\ u+2 \\ \vdots \\ K-2 \\ K-1 \end{array} & \begin{array}{c} und \\ u+1 \\ u+2 \\ \vdots \\ K-2 \\ K-1 \end{array} \\ \begin{array}{c} und \\ u+1 \\ u+2 \\ \vdots \\ K-2 \\ K-1 \end{array} & \begin{array}{c} 1 \\ b_0 \\ b_0 \\ \vdots \\ b_0 \\ b_0 \end{array} & \begin{array}{c} 0 \\ b_1 \\ b_1 \\ \vdots \\ b_1 \\ b_1 \end{array} & \begin{array}{c} 0 \\ b_2 \\ b_1 \\ \vdots \\ b_1 \\ b_1 \end{array} & \begin{array}{c} \cdot \\ \cdot \\ \cdot \\ \cdot \\ b_0 \\ b_0 \end{array} & \begin{array}{c} 0 \\ b_{K-u-1} \\ b_{K-u-2} \\ \vdots \\ b_1 \\ b_0 \end{array} & \begin{array}{c} 0 \\ 1-\Sigma \\ 1-\Sigma \\ \vdots \\ 1-\Sigma \\ 1-\Sigma \end{array} \end{array} \right] \end{array}$$

Figura IV.10. Matriz de probabilidades de transição  $\mathbf{Z}_0$  da cadeia de Markov que modela o comportamento do sistema durante um período de congestionamento.

A Figura IV.10 mostra a matriz de probabilidades de transição  $\mathbf{Z}_0$  da cadeia de Markov que modela o comportamento do sistema durante um período de congestionamento. Nessa matriz, o estado  $i$ ,  $u < i < K$  corresponde ao número de pares de células no multiplexador em um ponto embutido, durante um período de congestionamento. As probabilidades de transição  $b_i$ ,  $i \geq 0$  foram avaliadas em (IV.2). O estado *und* é um estado absorvente usado para modelar o fim do período de congestionamento.

O comprimento do período de congestionamento é uma variável aleatória,  $CP$ , que corresponde ao tempo que o multiplexador permanece no conjunto de estado:  $u+1 \dots, K-1$ . Para calcular a FDC de  $CP$ , associa-se uma recompensa a cada uma das transições que ocorre entre os estados de  $\mathbf{Z}_0$ . Especificamente, a função de atribuição de recompensas para este caso é dada por:

$$\eta_o(i, f) = \begin{cases} D_1, & u < i, f < K \\ 0, & i = und; u < f < K \end{cases} \quad (\text{IV. 44})$$

Note que atribuição de recompensas definida em (IV.45) depende somente do estado em que está a cadeia de Markov antes da transição.

Com a atribuição de recompensas de (IV.45), a recompensa acumulada pelo sistema, após um certo número de transições (que será explicado mais adiante), é equivalente ao comprimento do período de congestionamento, já que cada vez que o sistema visita o estado  $i$ ,  $u < i < K$ , ele ganha uma recompensa igual ao tempo que passa nesse estado e, quando entra ao estado absorvente  $und$ , pára de ganhar recompensas.

Seja  $n_o$  o número de transições feitas pela cadeia de Markov. Se  $n_o = \lceil r/D_1 \rceil$  significa que após  $n_o$  transições a recompensa acumulada pelo sistema é maior que  $r$ , em qualquer caminho que termine em algum estado  $f$ ,  $u < f < K$ . Isto implica que a recompensa acumulada pelo sistema somente pode ser menor ou igual a  $r$  após  $n_o$  transições, se em alguma transição antes da  $n_o$ -ésima a cadeia entra no estado absorvente  $und$ . Seja  $P[CP \leq r]$  a probabilidade que a duração do período de congestionamento seja menor ou igual a  $r$  unidades de tempo. Observa-se então que:

$$P[CP \leq r] = G_{und}(n_o, r), \quad (\text{IV. 45})$$

onde  $G_{und}(n_o, r)$  é recorrentemente avaliada usando (IV.5) a (IV.7).

Como foi estabelecido em (IV.6), para avaliar  $G_{und}(n_o, r)$  resta calcular a distribuição de probabilidade do estado inicial do período de congestionamento, ou seja  $\alpha_f(0)$ ,  $u < f < K$ . Isto pode ser feito notando que todos os períodos de congestionamento começam imediatamente após um período de não congestionamento. Então, condicionando no estado não-congestionado (e a partir do qual inicia-se um período congestionado), aplicando o teorema das probabilidades totais e usando a matriz  $\mathbf{P}$  da Seção IV.2.1, chega-se a:

$$\alpha_f(0) = \frac{\pi_0 a_f + \sum_{i=1}^u \pi_i a_{f-i+1}}{\pi_0 \sum_{k=u+1}^{K-1} a_k + \sum_{i=1}^u \pi_i \sum_{k=u+1}^{K-1} a_{k-i+1}}; u < f < K \quad (\text{IV. 46})$$

Finalmente, (IV.5), (IV.6), (IV.7) e (IV.47) permitem a avaliação recorrente de  $P[CP \leq r]$ .

#### **IV.2.4.3.3 FDC das Células Descartadas Durante o Período de Congestionamento.**

Para avaliar a função de distribuição cumulativa do número de células descartadas durante o período de congestionamento, o sistema foi modelado usando uma cadeia de Markov com a matriz de probabilidade de transição da Figura IV.10.

Seja a variável aleatória  $CD$  o número de células descartadas em um período de congestionamento. Para calcular a FDC de  $CD$ , associa-se uma recompensa a cada uma das transições entre estados da cadeia de Markov.

Para modelar o número de células descartadas em cada um dos estados, a função de recompensa apropriada, relacionada à transição de  $Z_0$ , é dada por:

$$\eta_D(i, f) = \begin{cases} 0 & i = und \\ 1 & u + 1 \leq i \leq K - 1 \end{cases} \quad (IV. 47)$$

Seja  $P[CD \leq r]$  a probabilidade que as células descartadas durante o período de congestionamento seja menor o igual a um certo  $r$ . O número de transições é  $n_D = \lceil r \rceil$ . Então, analogamente ao que foi feito para o cálculo de  $P[CP \leq r]$ , obtém-se que:

$$P[CD \leq r] = G_{und}(n_D, r) \quad (IV. 48)$$

Para a avaliação de  $G_{und}(n_D, r)$ , são usadas as equações (IV.5) a (IV.7), com o conjunto de probabilidades  $\alpha_f$ ,  $u < f < K$ , dado por (IV.47).

#### **IV.2.4.3.4 FDC do Número de Células Perdidas Durante o Período de Congestionamento.**

Para avaliar a função de distribuição cumulativa do número de células perdidas durante um período de congestionamento, modela-se o sistema usando uma cadeia de Markov que possui a matriz de probabilidades de transições dada pela Figura IV.11. Define-se como células perdidas as células que são descartadas devido a que ultrapassam o limiar de

congestionamento e aos pares de células que são descartadas devido que ao chegar acham o buffer saturado. Os estados  $u+1, \dots, K-1$  correspondem ao número de pares de células que se encontram na memória do multiplexador, em um ponto embutido, durante um período de congestionamento. O estado  $und$  é um estado absorvente, usado para modelar o fim deste período. Os estados  $K_h, 1 \leq h \leq e$ , são estados virtuais usados para modelar a perda de  $h$  pares de células. Ou seja, quando o sistema entra no estado  $K_h$ , supõe-se que o buffer está cheio e que  $h$  pares de células (virtuais) que foram perdidas por esse motivo. Note que somente um par de células é servido em um período de tempo  $D_1$  (entre dois pontos embutidos adjacentes), devido ao fato que, do par de células, somente a célula de alta prioridade é transmitida quando o sistema está congestionado. Como consequência, quando o sistema entra no estado  $K_h$ ,  $2h+1$  células foram perdidas. Por simplificação, assume-se que as perdas ocorrem nas transições de saída do estado  $K_h$ .

	$und$	$u+1$	...	$K-2$	$K-1$	$K_1$	...	$K_{e-1}$	$K_e$
$und$	1	0	...	0	0	0	...	0	0
$u+1$	$b_0$	$b_1$	...	$b_{K-u-2}$	$b_{K-u-1}$	$b_{K-u}$	...	$b_{K-u+e}$	$1-\Sigma$
...			...				...		
$K-2$			...	$b_1$	$b_2$	$b_3$	...	$b_{e+1}$	$1-\Sigma$
$K-1$			...	$b_0$	$b_1$	$b_2$	...	$b_e$	$1-\Sigma$
$K_1$			...	0	0	$b_1$	...	$b_{e-1}$	$1-\Sigma$
...			...				...		
$K_{e-1}$			...	0	0	0	...	$b_1$	$1-\Sigma$
$K_e$			...	0	0	0	...	0	1

Figura IV.11. Matriz de probabilidade de Transição  $\mathbf{Z}_L$  da cadeia de Markov que modela o número de células perdidas durante um período de congestionamento.

Seja a variável aleatória  $CL$  o número de células perdidas em um período de congestionamento. Para calcular a FDC de  $CL$ , associa-se uma recompensa a cada uma das transições entre estados os da cadeia de Markov. A função de atribuição de recompensas, apropriada às transições de  $\mathbf{Z}_L$ , é dada por:

$$\eta_L(i, f) = \begin{cases} 0 & i = und \\ 1 & u+1 \leq i \leq K-1 \\ 2h+1 & i = K_h; \end{cases} \quad (\text{IV. 49})$$

Note que cada vez que o sistema visita um estado  $i$ ,  $u+1 \leq i \leq K-1$ , ele ganha uma recompensa igual a 1, modelando a célula de baixa prioridade perdida nesses estados. Quando o sistema transita aos estados  $K_h$ ,  $1 \leq h \leq e$ , ele ganha uma recompensa igual a  $2h + 1$ , que modela a célula de baixa prioridade perdida pelo mecanismo de descarte, mais as  $2h$  células perdidas devido ao buffer estar completo quando chegam os  $h$  pares de células. Finalmente, porque o sistema não perde células no estado absorvente, nesse estado ele ganha uma recompensa igual a 0.

Seja  $P[CL \leq r]$  a probabilidade que as células descartadas durante o período de congestionamento seja menor ou igual a um determinado número  $r$  e  $n_L = \lceil r \rceil$ .

Então, observa-se que:

$$P[CL \leq r] = G_{und}(n_L, r) \quad (\text{IV. 50})$$

A avaliação de  $G_{und}(n_L, r)$  é feita usando (IV.5) a (IV.7) e o conjunto de probabilidades  $\alpha_f$ ,  $u < f < K$ , dado por (IV.47).

#### ***IV.2.4.3.5 FDC do Comprimento do Período em que o Sistema não está em Congestionamento.***

A Figura IV.12 mostra a matriz de probabilidades de transição  $\mathbf{Z}_u$  da cadeia de Markov que modela o sistema durante um período de não-congestionamento. Nessa cadeia, os estados  $i$ , com  $0 < i \leq u$ , correspondem ao número de pares de células que contém o multiplexador, nos pontos embutidos. O estado  $ov$  é um estado absorvente, usado para modelar o final de um período de não-congestionamento.

$$\begin{array}{c}
\mathbf{0} \\
\mathbf{1} \\
\mathbf{2} \\
\vdots \\
\mathbf{u-1} \\
\mathbf{u} \\
\mathbf{ov}
\end{array}
\begin{bmatrix}
\mathbf{0} & \mathbf{1} & \mathbf{2} & \dots & \mathbf{u-1} & \mathbf{u} & \mathbf{ov} \\
a_0 & a_1 & a_2 & \dots & a_{u-1} & a_u & 1-\Sigma \\
a_0 & a_1 & a_2 & \dots & a_{u-1} & a_u & 1-\Sigma \\
0 & a_0 & a_1 & \dots & a_{u-2} & a_{u-1} & 1-\Sigma \\
& & & \dots & & & 1-\Sigma \\
& & & & \dots & a_1 & a_2 & 1-\Sigma \\
& & & & & \dots & a_0 & a_1 & 1-\Sigma \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

Figura IV.12: Matriz de probabilidades de transição  $\mathbf{Z}_u$  da cadeia de Markov que modela o comportamento do sistema durante um período não congestionado.

O comprimento de um período não congestionado é uma variável aleatória, NCP, que corresponde ao tempo que o sistema permanece nos estados  $0, 1, \dots, u$ . Para calcular a FDC de NCP, associa-se uma recompensa a cada uma das transições que acontece entre esses estados. A função de atribuição de recompensas para esse caso é:

$$\eta_u(i, j) = \begin{cases} D_2 + \frac{1}{\lambda}; & i = 0 \\ D_2; & 0 < i \leq u \\ 0; & i = ov \end{cases} \quad (\text{IV. 51})$$

Essa função de atribuição de recompensas estabelece que o sistema ganha uma recompensa igual a  $D_2 + 1/\lambda$  em cada uma das transições que começam no estado  $i=0$ ; uma recompensa igual a  $D_2$  em cada uma das transições que começam em qualquer outro estado não congestionado; e uma recompensa igual a 0 quando a cadeia de Markov entra em congestionamento (cai ao estado absorvente  $ov$ ). Seja  $P[NCP \leq r]$  a probabilidade que um período não congestionado seja menor ou igual a  $r$  unidades de tempo e,  $n_u = \lceil r/D_2 \rceil$ . Chega-se a:

$$P[NCP \leq r] = G_{ov}(n_u, r), \quad (\text{IV. 52})$$

onde  $G_{ov}(n_u, r)$  é avaliado usando o método da Seção IV.1. Pelo princípio que todos os períodos não congestionados começam em  $u$ , tem-se que:

$$\alpha_i(0) = \begin{cases} 0 & 0 \leq i < u \\ 1 & i = u \end{cases} \quad (\text{IV. 53})$$

#### IV.2.4.3.6 Caracterização de Valor Médio de um Período de Congestionamento

Até o momento, a caracterização dos períodos de congestionamento foi obtida usando funções de distribuição. Não obstante, também é útil conhecer os valores médios dessas medidas. O valor médio pode ser calculado pela forma tradicional, usando as funções de distribuição recém obtidas. Contudo, nesta sub-seção, os valores médios das medidas de congestionamento são obtidos usando um método mais eficiente. Este método faz a avaliação de forma recorrente, o que permite sua avaliação direta, isto é, sem utilizar as funções de distribuição.

Em primeiro lugar, é estabelecido e resolvido um problema matemático geral. A solução obtida é então utilizada para avaliar os diferentes valores médios de interesse. Com esse propósito, considera-se a função de atribuição de recompensas  $\eta(i,j)$ . Seja  $R$  a recompensa média ganha pelo sistema durante um período de congestionamento, e  $R_i$  a recompensa média ganha pelo sistema desde o instante em que a cadeia de Markov entra no estado  $i$ ,  $u < i < K$ , até o instante no qual a cadeia de Markov transita, pela primeira vez, para o estado  $(i-1)$ .

Começando no estado  $i$ , o sistema ganha uma recompensa média igual a  $R_i$  até que transita pela primeira vez para o estado  $(i-1)$ . Nesse estado  $(i-1)$ , ganha a recompensa  $R_{i-1}$ , até transitar pela primeira vez ao estado  $(i-2)$  e, assim, sucessivamente. Finalmente, quando o sistema transita pela primeira vez ao estado  $(j-1)$ , ele ganha  $R_j$ .

Seja  $R_{i,j}$ ,  $u < j \leq i < K$ , a recompensa média ganha pelo sistema, em um período de congestionamento que começa no estado  $i$  e que acaba quando a cadeia de Markov transita, pela primeira vez, para o estado  $(j-1)$ .

Pela definição recém realizada de  $R_i$  e  $R_{i,j}$ , obtém-se que:

$$R_{i,j} = \sum_{l=i}^j R_l; \quad u < j \leq i < K \quad (\text{IV. 54})$$

Como anteriormente,  $\alpha_i(0)$  é a probabilidade que a cadeia de Markov comece um período de congestionamento no estados  $i$ ,  $u < i < K$ . Condicionando no estado inicial do período congestionado, chega-se a:

$$R = \sum_{i=u+1}^{K-1} \alpha_i(0) R_{i,u+1} \quad (\text{IV. 55})$$

Aplicando (IV.55) e alterando a ordem dos somatórios, obtém-se:

$$R = \left( \sum_{l=u+1}^{K-1} R_l \right) \sum_{i=l}^{K-1} \alpha_i(0), \quad (\text{IV. 56})$$

onde  $\alpha_i(0)$ ,  $u < i < K$ , é dada por (IV.17).

Para completar a solução de  $R$ , é necessário encontrar uma expressão para  $R_l$ ,  $u < l < K$ . Isso pode ser feito de forma recorrente no valor de  $l$ , iniciando em  $u = K-1$  e chegando até  $u = 1$ , como se descreve a seguir. Primeiro, para avaliar  $R_{K-1}$ , condiciona-se na primeira transição da cadeia de Markov, que possui matriz de probabilidades de transição de  $\mathbf{Z}_0$ , onde:

$$R_{K-1} = \eta(K-1, K-1) + (1 - b_0) R_{K-1} \quad (\text{IV. 57})$$

Simplificando (IV.58):

$$R_{K-1} = \frac{\eta(K-1, K-1)}{b_0} \quad (\text{IV. 58})$$

Para  $u < l < K$ , condiciona-se novamente à primeira transição da cadeia de Markov:

$$R_l = \eta(l, l) + \sum_{m=l}^{K-2} b_{m-l+1} R_{m,l} + \left( 1 - b_0 - \sum_{m=l}^{K-2} b_{m-l+1} \right) R_{K-1,l}$$

Usando (IV.55) e após algumas manipulações algébricas, chega-se a:

$$R_l = \frac{1}{b_0} \left\{ \eta(l, l) + \sum_{n=l+1}^{K-1} R_n \left( 1 - \sum_{k=0}^{n-l} b_k \right) \right\}, \quad u < l < K \quad (\text{IV. 59})$$

Resumindo, (IV.57), (IV.60) e (IV.46) permitem avaliar  $R$ , a recompensa média ganha pelo sistema durante um período de congestionamento. Em outras palavras, as equações anteriores permitem avaliar distintas medidas de valor médio, sempre que se tenha definida a função de atribuição de recompensas adequada a cada caso.

Para calcular os valor médio do comprimento de um período de congestionamento, utiliza-se o método recorrente recém derivado e a função de atribuição de recompensas dada em (IV.45).

De modo similar, para calcular o número médio de células descartadas durante um período de congestionamento, usa-se o método recorrente e a função de atribuição de recompensas dada em (IV.48).

## IV.2.5 Exemplos Numéricos

Nesta seção são mostrados alguns exemplos numéricos usando os modelos e a resolução analítica da seção anterior. Para os exemplos desta seção, foram feitas as seguintes suposições:

- fator de atividade de voz (surto de voz) de 42%.
- canal T1, com uma taxa nominal de 1.544 Mbps.
- banda passante usada por ATM, sinal DS1, é de 1.536 Mbps.
- comprimento do buffer do multiplexador de  $K$  pares de células de voz. Este valor é mostrado nos gráficos que se seguem. Para muitos dos exemplos  $K=30$  pares de células. Este tamanho de buffer foi escolhido para não introduzir retardos desnecessários ao tráfego de voz devido ao enfileiramento.
- Para muitos dos exemplos, o limiar de descarte de células de voz é igual a 7.15 mseg, que corresponde a  $u=13$  pares de células.
- $K$  e  $u$  são parâmetros do sistema expressos na forma  $(u,K)$  e, quando requerido, seus valores são escritos nas figuras que ilustram as avaliações.

### IV.2.5.1 Exemplos de FDC

A Figura IV.13 ilustra a FDC do tempo de residência de um par de células no multiplexador. O eixo do tempo está normalizado pelo tempo necessário para transmissão de um par de células ( $D_2$ ). Para um canal T1, o par de células é transmitido aproximadamente em  $D_2=0.55$  mseg. As curvas são feitas para diferentes percentuais de ocupação do canal T1. Os valores do limiar de descarte são mostrados nos gráficos que serão apresentados.

Como pode ser visto na Figura IV.13, para uma ocupação do canal entre 50% e 70%, as células experimentam, com uma alta probabilidade, um tempo de residência baixo no multiplexador. Por outro lado, para uma ocupação de canal de 100%, observa-se que, com alta probabilidade, as células experimentam um tempo de residência no multiplexador maior do que para o caso de baixa carga. Por exemplo, para uma probabilidade 0.83 e uma ocupação do canal de 50%, o tempo de residência das células no buffer é menor ou igual ao tempo necessário para transmitir um par de células. Para essa mesma probabilidade e para uma ocupação do canal de 100%, o tempo de residência é menor ou igual ao tempo necessário para transmitir 10 pares de células. Pode ser observado que, nesse caso, o mecanismo de descarte de células tem uma maior influência no retardo, porque o efeito global suaviza o incremento do tempo de residência das células quando a carga aumenta.

Também pode ser notado, na Figura IV.13, que as curvas apresentam características diferentes para carga baixa (entre 50% e 70%) e para carga média a alta (entre 80% e 100%). No primeiro caso, é verificado que as curvas mostram saltos significativos. Cada salto corresponde, aproximadamente, à probabilidade que o par de células encontre o sistema em um certo estado. Por exemplo, para 50% de carga, o par de células em estudo encontrará, com alta probabilidade, o buffer vazio (esse caso corresponde ao primeiro salto vertical na curva) e, portanto, permanecerá no sistema durante um tempo igual a  $D_2$ . Observa-se também que, com alta probabilidade, o par de células que chega encontrará somente um par de células no buffer, permanecendo no sistema um tempo igual a  $2 * D_2$ .

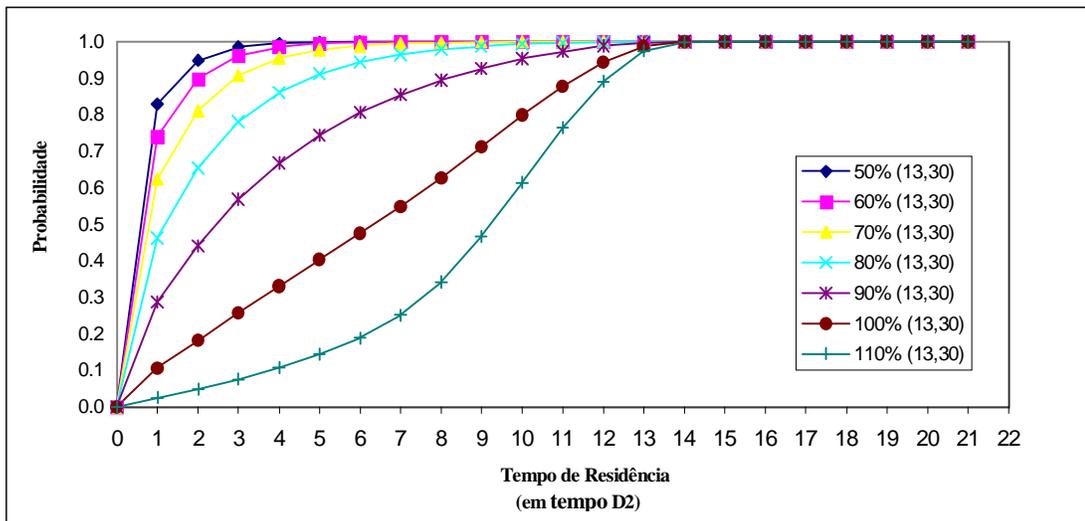


Figura IV.13. Função de Distribuição Cumulativa do tempo de residência.

Para os casos de carga média a alta, verifica-se que as curvas são mais suaves que no caso de carga baixa. Este fenômeno é devido à ação do mecanismo de descarte de células, que acelera a transmissão das células quando o estado do buffer está acima do limiar de descarte, ou equivalentemente, quando o tempo de residência das células é alto.

A Figura IV.14 mostra a influência do valor do limiar de congestionamento sobre a FDC do tempo de residência. Os gráficos foram feitos para uma ocupação de 80% e 100% de um canal T1 e para diferentes valores do limiar de congestionamento ( $u=7, 10, 13$  e  $16$ ). Pode ser visto que o tempo de residência é menor para valores baixos do limiar de congestionamento. Esta influência é ainda mais significativa para uma carga de canal de 100%, onde, por exemplo, para uma probabilidade 0,96 e  $u=7$ , o tempo de residência é menor ou igual a 7 pares de células (ou 3.86 msec). Não obstante, para  $u=16$ , esse tempo de residência é menor ou igual ao tempo usado para transmitir 15 pares de células (ou, equivalentemente, 8.28 msec). Também pode ser observado que, para cargas altas, o tempo de residência é menor ou igual ao limiar de descarte  $u$ , com uma probabilidade igual a 1.

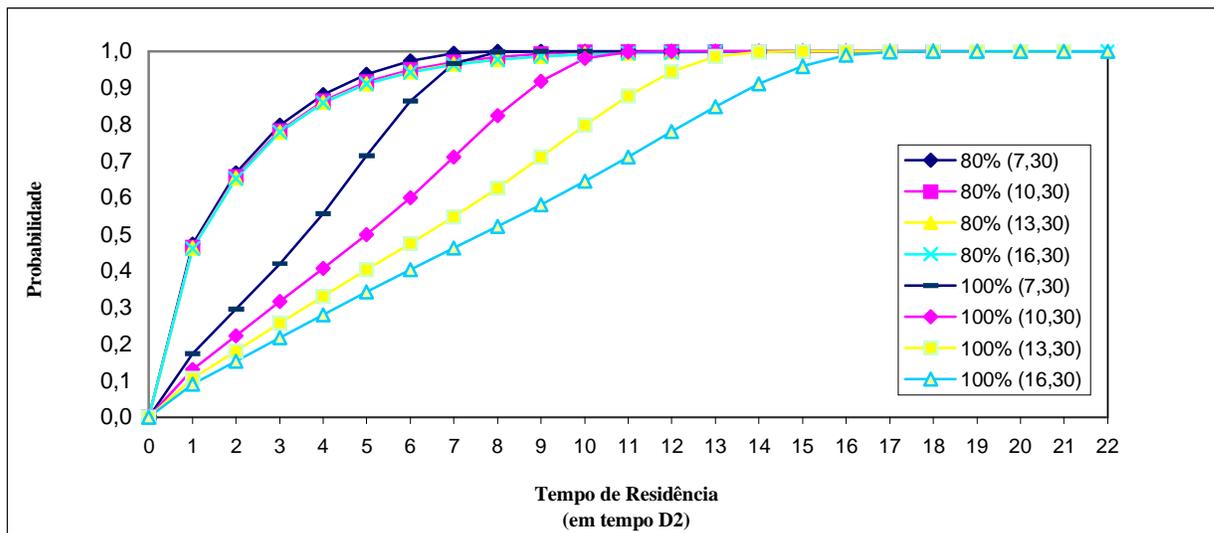


Figura IV.14. FDC do tempo de residência para diferentes valores do limiar de descarte.

A Figura IV.15 mostra a FDC do comprimento do período de congestionamento. O eixo do tempo é normalizado pelo tempo necessário para transmissão de uma célula ( $D_1$ ). Em um canal T1 esse tempo é equivalente a  $D_1=0.275$  msec. As curvas são exibidas para diferentes

percentuais de ocupação do canal T1. O limiar de descarte usado ( $u$ ) é de 13 pares de células e o tamanho do buffer ( $K$ ) é de 30 pares de células.

Como mostrado na Figura IV.15, para uma ocupação de canal de 50%, que representa uma carga baixa para um canal T1, o comprimento do período de congestionamento é menor ou igual ao tempo necessário para transmitir 2.25 células (ou aproximadamente 0.62 mseg), com probabilidade igual a 0.9. Por outro lado, a curva que corresponde a 100% de carga, mostra que o comprimento do período de congestionamento é menor ou igual ao tempo necessário para transmitir 5 células. Em geral, a Figura IV.15 mostra que, com probabilidade menor ou igual a 0.5, os períodos de congestionamento são de um comprimento igual, no máximo, ao tempo necessário para transmitir 2 células, para qualquer carga abaixo de 100% de ocupação do canal T1.

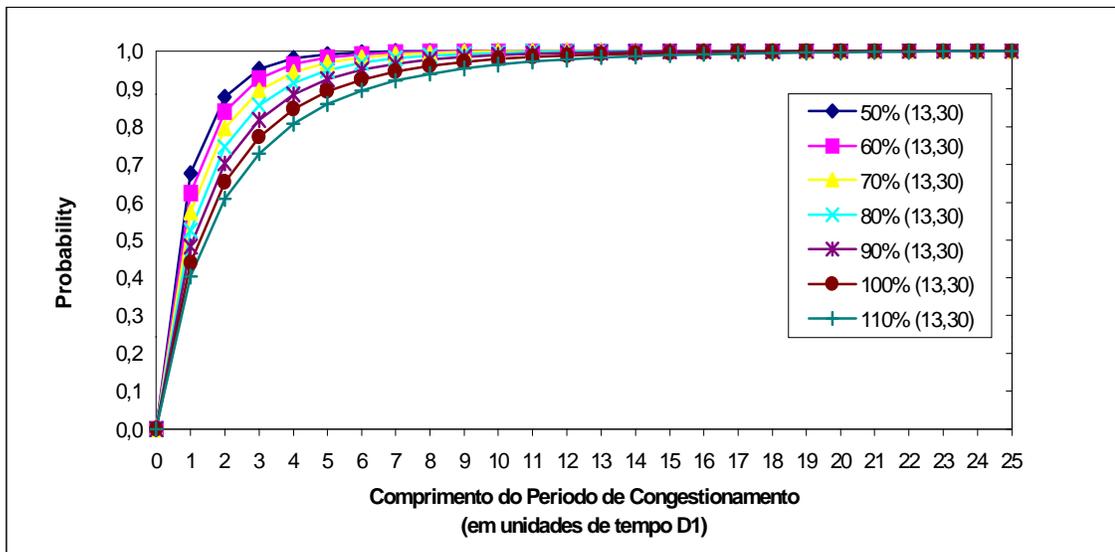


Figura IV.15. Função de Distribuição cumulativa do comprimento do período de congestionamento.

A influência da carga é significativa na determinação da probabilidade do período de congestionamento durar entre 2 a 10 unidades de tempo de transmissão de uma célula, mas para qualquer carga razoável (abaixo de 100%), a probabilidade que o período congestionado dure mais que 10 células é baixo (menor que 0.1). Esse fato mostra a eficiência do mecanismo de descarte de células.

O gráfico da Figura IV.16 mostra a influência do limiar de descarte sobre a FDC do comprimento do período congestionado. Como nos casos prévios, o eixo do tempo está normalizado pela unidade de tempo de transmissão de uma célula,  $D_1$ . As curvas foram feitas para diferentes percentuais de ocupação de um canal T1. No gráfico, o limiar de descarte de células e o tamanho de buffer são mostrados como um par ordenado  $(u, K)$ .

Pode-se notar que, para uma mesma carga, os limiares de descarte selecionados não têm influência na FDC do comprimento de um período congestionado. Esta é a razão pela qual, para cada uma dessas cargas (50%, 90%, 110%), as curvas que correspondem aos diferentes limiares não são diferenciadas.

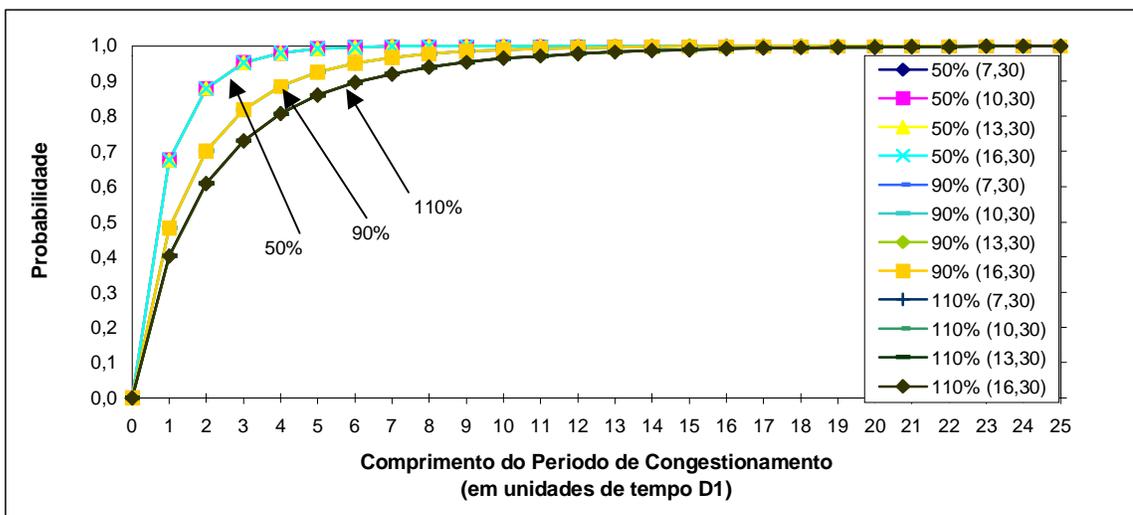


Figura IV.16. Função de Distribuição Cumulativa do Comprimento de um Período Congestionado para diferentes limiares de descarte de células.

A Figura IV.17 mostra a FDC do número de células descartadas para diferentes cargas de canal T1. Em geral, com uma probabilidade 0,5, para qualquer carga, o número de células descartadas durante o período em congestionamento é no máximo de 2 células.

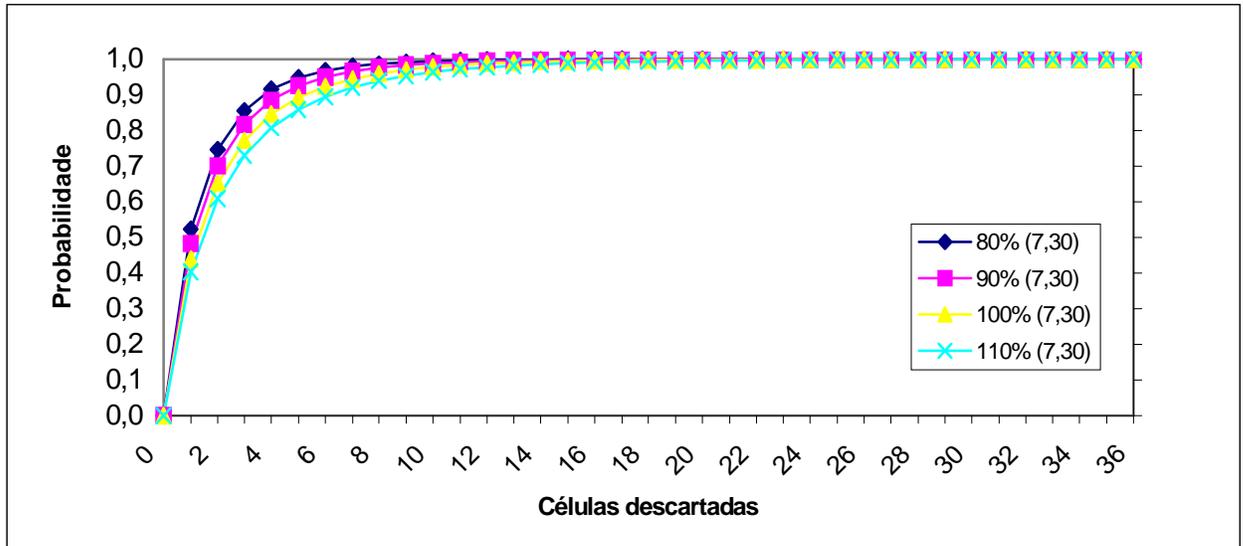


Figura IV.17. Função de Distribuição Cumulativa do Número de Células Descartadas.

#### IV.2.5.2 Exemplos de Medidas de Valor Médio

A Figura IV.18 mostra o número médio de células no buffer do multiplexador versus o percentual de ocupação do canal T1. As curvas estão feitas para um tamanho máximo de buffer de 30 pares de células.

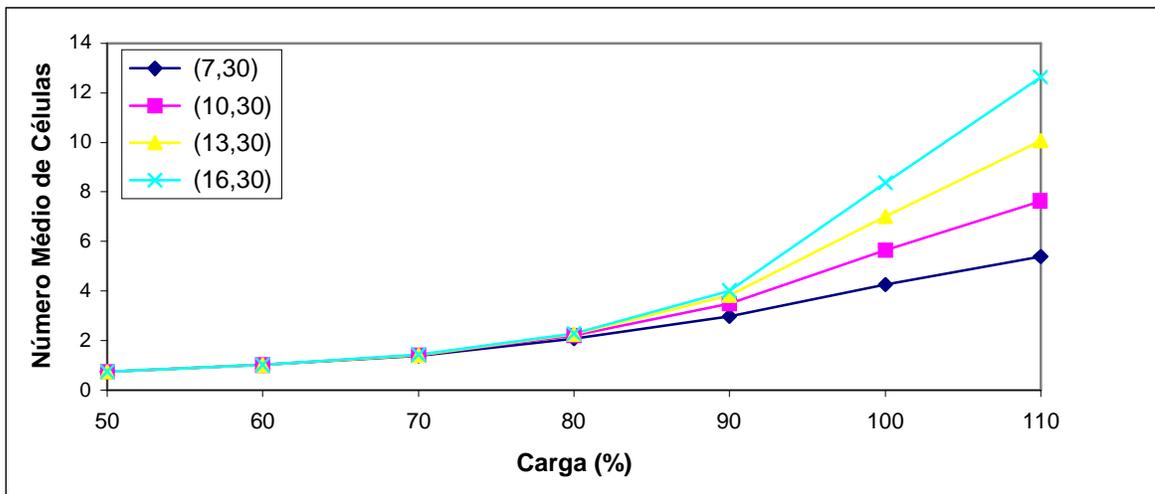


Figura IV.18. Número Médio de células no Multiplexador.

Na Figura IV.18 pode ser visto que, para qualquer carga, o número médio de células no sistema é independente do tamanho do buffer. Também pode ser observado que o número médio de células é baixo para cargas menores ou iguais a 80%. Para cargas entre 80% e 110%, o sistema tem um número médio de células menor que o limiar de descarte  $u$ , devido ao sistema alternar entre estados congestionados e estados não congestionados. Nos estados congestionados, o mecanismo de descarte de células elimina células de baixa prioridade, mantendo o número de células no buffer próximo do limite de descarte.

A Figura IV.19 mostra o tempo médio gasto na fila pelas células versus o percentual de ocupação de um canal T1. As curvas são feitas para diferentes limiares de descarte, que são mostrados no gráfico.

Pode ser observado que para cargas menores que 80%, o valor do limiar não tem qualquer influência sobre o tempo médio de residência na fila. Note também que o tempo gasto na fila está abaixo de 1 mseg.

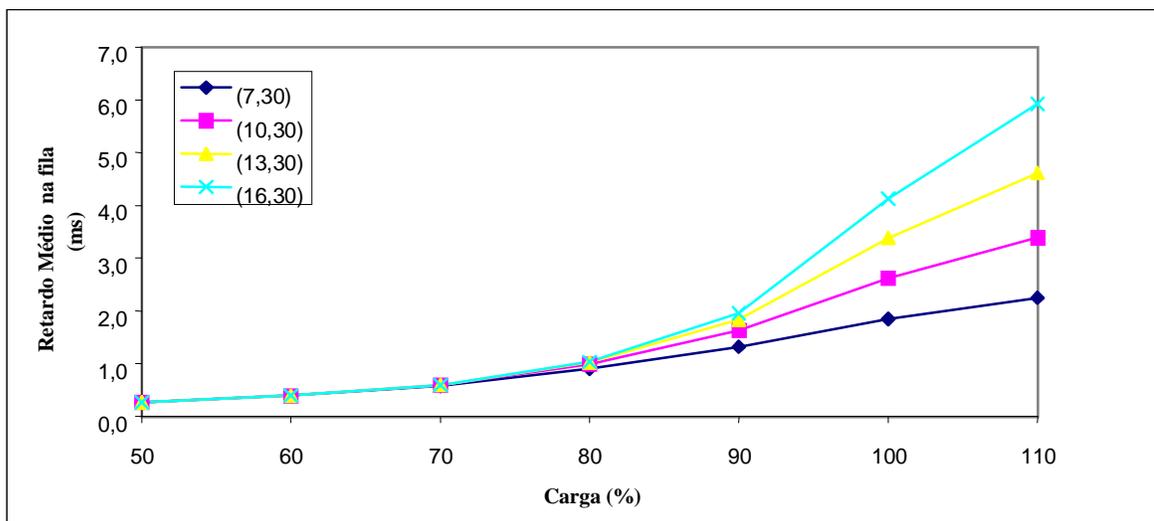


Figura IV.19. Tempo Médio na Fila

Para cargas maiores que 80%, pode ser melhor observada a influência do mecanismo de descarte. Nesse caso, para um limite de descarte igual a 7 pares de células, o tempo médio na fila é baixo, menor que 2 pares de células. Isso se deve ao fato que, se o número de células no buffer do multiplexador exceder o limite  $u$  (sistema é considerado congestionado), o mecanismo de descarte de células acelera o tempo de serviço das células restantes.

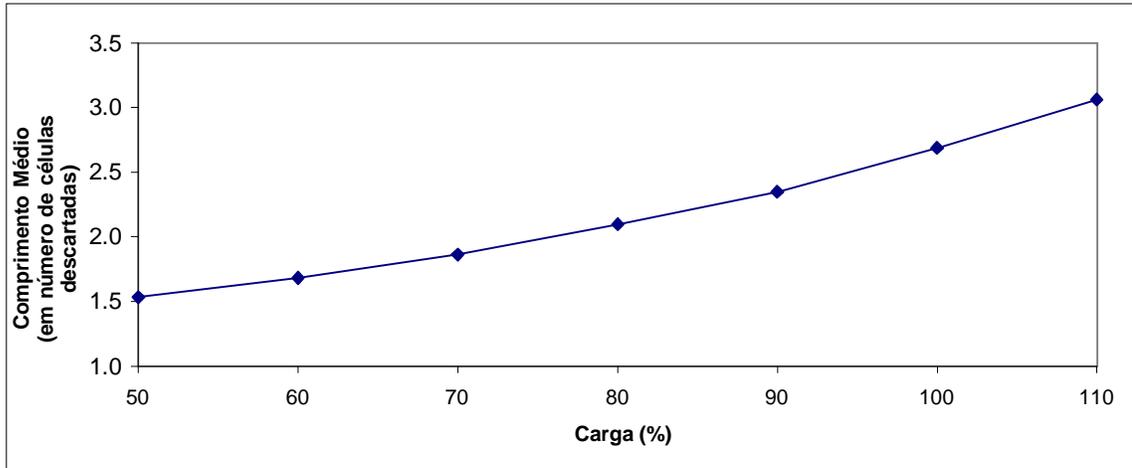


Figura IV.20. Tamanho médio de um período congestionado.

O gráfico da Figura IV.20 mostra o comprimento médio de um período congestionado versus o percentual de ocupação de um canal T1. Pode se notar que, quando é transmitida somente uma célula (àquela de alta prioridade), o percentual de ocupação do canal pode ser maior que 100%. Na figura, o comprimento médio de um período de congestionamento é expresso em termos do número de células que descartadas durante o período congestionado.

Ainda na figura, pode ser visto que o limiar de descarte não tem influência sobre o comprimento do período de congestionamento. Outro fato (intuitivo) que pode ser visto na Figura IV.20 é que, na medida que a carga aumenta, o comprimento médio do período de congestionamento também aumenta.

A Figura IV.21 ilustra o número médio de células descartadas durante o período de congestionamento para diferentes cargas de um canal T1. Os valores do tamanho máximo de buffer e do limiar de congestionamento são mostrados na figura.

Pode ser observado que o número de células descartadas aumenta quando a carga é incrementada. Entretanto, o número de células descartadas é independente do valor do limiar de descarte.

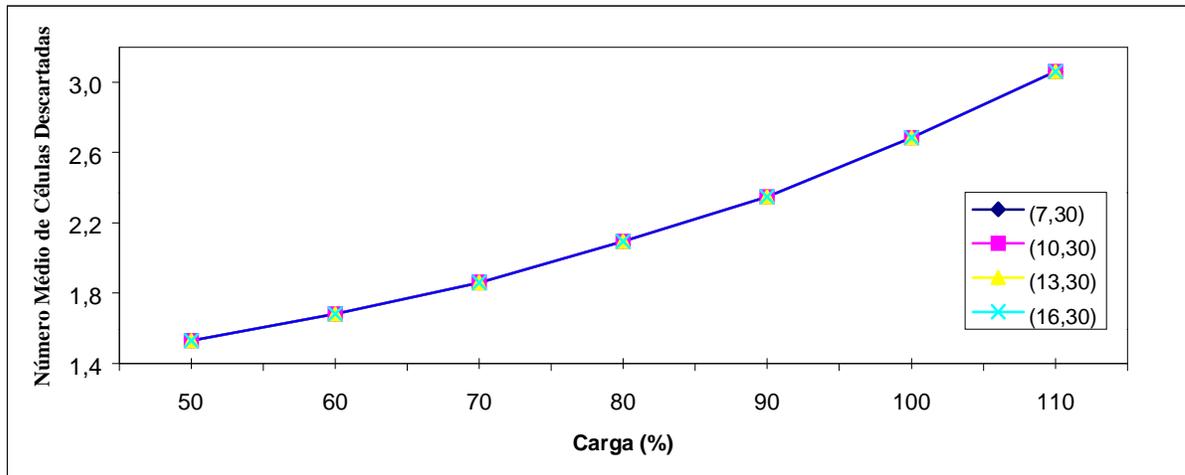


Figura IV.21: Número Médio de Células Descartadas

#### IV.2.6 Resumo da seção IV.2

A análise de desempenho de multiplexadores de células, sob algoritmos de controle de congestionamento não é uma tarefa trivial. A contribuição mais importante, entretanto, foi o desenvolvimento de um novo método para a obtenção de novas medidas de desempenho que caracterizam o comportamento do multiplexador durante os períodos de congestionamento, como apresentado na Seção IV.2.3. Esse método foi usado para avaliar a FDC do tempo de residência das células no multiplexador, uma tarefa muito complexa, pois essa distribuição depende não apenas das células que já estavam no multiplexador no instante de chegada do par de células sob análise, mas também daquelas que chegam durante o tempo de permanência daquele par de células. O método foi aplicado a avaliação de diversas medidas de desempenho durante períodos de congestionamento. Entre elas a FDC do comprimento do período de congestionamento, a FDC das células descartadas no período de congestionamento, a FDC do número de células perdidas durante o período de congestionamento. Os valores médios destas medidas também foram calculados.

Todas as medidas avaliadas permitem caracterizar em grande detalhe o comportamento de um multiplexador durante os períodos de congestionamento e, como consequência, ajudam a entender melhor seu comportamento e a projetar mais eficientemente esses multiplexadores.

Além de analisar a rede para tráfego de voz, também é importante avaliar a rede quando estão presentes nela diversos tráfegos. Para tanto, é importante analisar algoritmos de escalonamento, um dos pontos sensíveis na adequada provisão de QoS. Este é o tópico da Seção IV. 3. no qual é apresentada uma nova metodologia, proposta nesta tese, para a análise de desempenho de algoritmos de escalonamento de banda passante usando o Meta-Algoritmo LDS proposto na Seção III.

### **IV.3 Metodologia Genérica para a Análise de Desempenho do Meta-Algoritmo LDS.**

Nesta seção é proposta uma metodologia geral para a análise de algoritmos de escalonamento de banda passante. Para atingir este objetivo, utiliza-se o Meta-algoritmo LDS, proposto na Seção III, e avalia-se seu desempenho em forma global. Posteriormente, para avaliar um determinado algoritmo de escalonamento, é utilizada a Tabela LDS do algoritmo em análise. Desta maneira, as medidas de desempenho que são obtidas para LDS correspondem, então, às medidas de desempenho do algoritmo em particular.

A seguir, como exemplo, é feita a análise para um multiplexador ATM, com descarte seletivo de células, que compartilha um canal comum de saída entre tráfegos de voz e de dados textuais.

#### **IV.3.1 Multiplexador ATM para Tráfegos de Voz e Dados com Controle de Congestionamento**

A Figura IV.22 apresenta um esquema simplificado do multiplexador genérico da Figura III.1, para um multiplexador capaz de atender tráfego de tempo real (voz) e tráfego sem restrição de tempo real (dados textuais). Para atribuir dinamicamente a banda passante do canal de saída aos dois tipos de tráfego, o multiplexador usa o meta-algoritmo de escalonamento LDS, descrito na seção III.2.

O multiplexador também possui um mecanismo de descarte de células de voz, que é usado em eventuais situações de congestionamento.

Tanto a forma como a informação de voz é empacotada em células ATM quanto a descrição do mecanismo de descarte de células é a mesma feita na Seção IV.2.2.

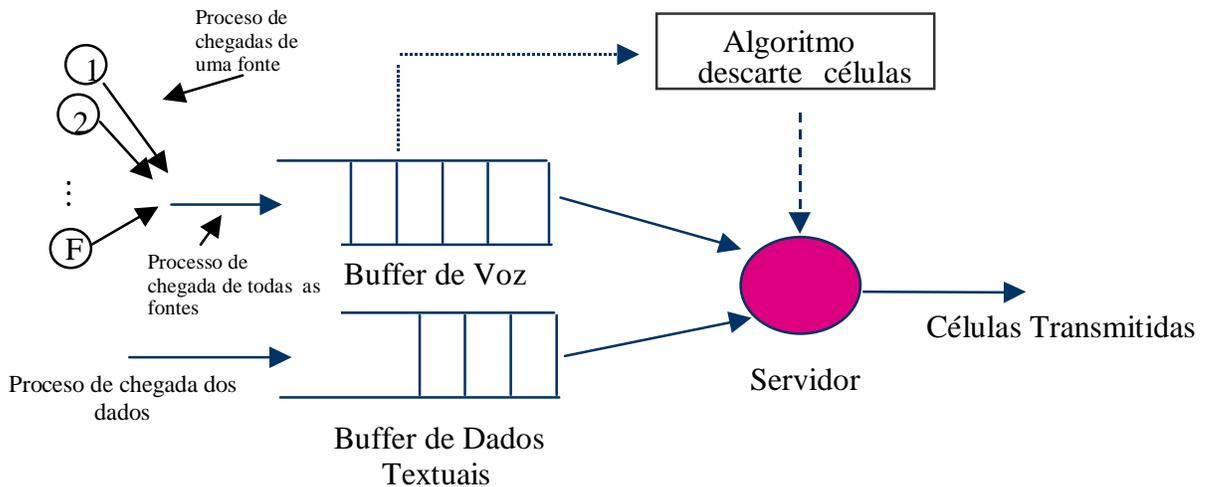


Figura IV.22. Multiplexador ATM para Tráfegos de Voz e Dados.

Note que a relação existente entre a política de escalonamento e o controle de congestionamento é a seguinte. A política de escalonamento especifica a duração de cada fase. Por outro lado, o controle de congestionamento especifica que, enquanto durar a fase de voz, se este tipo de tráfego entrar em congestionamento as células de voz de baixa prioridade serão descartadas.

### IV.3.2 Avaliação de Desempenho do Multiplexador

As medidas de desempenho avaliadas são: o número médio de pares de células de voz presentes no buffer de voz, o número médio de células de dados armazenadas no buffer de dados, o retardo médio experimentado por cada tipo de célula e a probabilidade de perda de células de voz e de dados. A avaliação destas medidas é relevante porque permite estabelecer se o multiplexador pode ou não cumprir com a QoS que cada tipo de tráfego requeira.

Para avaliar as medidas de desempenho mencionadas acima, na seção seguinte descreve-se o modelo que representa a operação do multiplexador. O modelo é composto pela representação da chegada dos tráfegos de voz e dados textuais, a capacidade de armazenamento que possui o multiplexador para ambos tipos de tráfego, a disciplina de serviço que cada tipo de tráfego recebe por parte do multiplexador, o número de processadores do multiplexador e a velocidade de processamento.

### **IV.3.3. Modelo de Operação do Multiplexador**

#### **IV.3.3.1 Processos de Chegada**

O processo de chegada de células de voz ao multiplexador é o mesmo que foi descrito em detalhes na Seção IV.2.3.3.

Supõe-se que o tráfego de dados é sem conexão e, que o processo de chegada das células de dados textuais, é caracterizado também como um processo de Poisson [HEF86,SRI86].

Para efeito de exemplificação nesta tese, modela-se a chegada de pares de células de voz e de células de dados por meio de processos de Poisson independentes entre si, com parâmetro  $\lambda_v$  e  $\lambda_d$ , respectivamente.

#### **IV.3.3.2 Serviço das Células**

Para atender o tráfego de voz, o multiplexador mantém um buffer com um tamanho de  $K_v$  pares de células de voz. O serviço dos pares de células de voz é realizado somente durante a fase de voz (durante a fase de dados o multiplexador não serve células de voz, somente as recebe); a disciplina de serviço neste buffer é FIFO e o tempo que o multiplexador demora para atender um par de células de voz é determinístico e dependente da carga [SRI89].

Seja  $s_{n_v}$  o tempo necessário para transmitir um par de células de voz quando há  $n_v$  pares de células de voz no multiplexador. Então, devido à política de descarte de células (ver Seção IV.3.1), obtém-se que:

$$s_{n_v} = \begin{cases} D_2 & n_v = 1, 2, \dots, u \\ D_1 & n_v = u + 1, \dots, K_v \end{cases} \quad (\text{IV.61})$$

onde  $u+1$  é o mínimo número de pares de células de voz presentes no buffer para o multiplexador ser considerado em congestionamento (ver Seção IV.2.2), e  $D_2$  ( $=2 D_1$ ) é o tempo necessário para transmitir um par de células de voz quando não há congestionamento.

Para o tráfego de dados, o multiplexador possui um buffer de tamanho  $K_d$  células de dados. O serviço das células de dados somente é realizada durante a fase de dados (durante a fase de voz o multiplexador não atende células de dados, só as recebe). A disciplina de atenção neste buffer é FIFO e o tempo que o multiplexador demora para atender uma célula de dados é determinístico e igual a  $D_1$ .

#### IV.3.3.3. Modelos de Filas do Multiplexador

Dado que a chegada de pares de células de voz é modelada com um processo Poisson, que o tempo que o multiplexador demora em atender uma célula é determinístico e dependente da carga, que existe um único processador que serve as células, que o buffer de voz é de tamanho finito igual a  $K_v$  e que durante a fase de dados não se atende tráfego de voz, o modelo utilizado nesta tese para o tráfego de voz corresponde a uma fila  $M / \tilde{D} / 1 / K_v$  com férias. Onde as férias correspondem ao período de tempo no qual não são atendidas as células de voz.

Por outro lado, considerando que a chegada de células de dados se modela com um processo Poisson, que o tempo usado pelo multiplexador para atender uma célula é determinístico, que existe um único processador que serve às células, que o buffer de dados é de tamanho finito igual a  $K_d$  e que durante a fase de voz não se atende tráfego de dados, o modelo utilizado nesta tese para o tráfego de dados corresponde também a uma fila  $M / D / 1 / K_d$  com férias. Onde as férias correspondem ao período de tempo em que não se atendem células de dados.

A pesar do modelo usado ser uma fila  $M / D / 1 / K$  (ou  $M / \tilde{D} / 1 / K$ ) com férias, a análise matemática que se apresenta não utiliza as equações de valor médio de uma fila  $M/G/1$  com férias [KLEI78]. Alternativamente, é feita uma análise específica para o caso em estudo.

### IV.3.4. Definições Básicas

Define-se a cadeia de Markov de parâmetro discreto  $C$ , cujas transições ocorrem nos instantes em que se inicia um ciclo. Os extremos da seta curva da Figura IV.23 mostram os instantes onde ocorrem as transições da cadeia  $C$ .

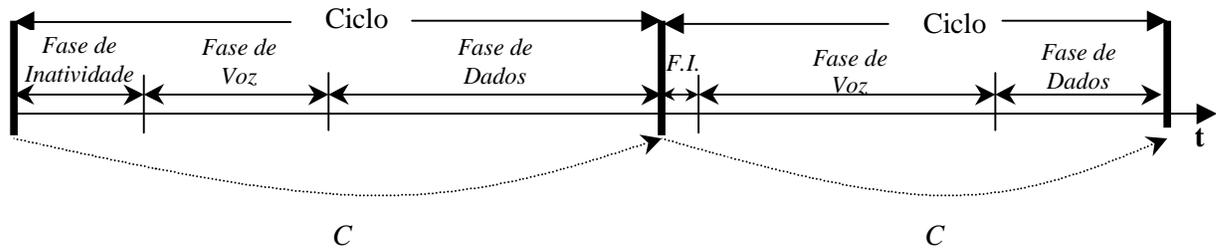


Figura IV.23 Pontos embutidos nos quais é definida a Cadeia de Markov  $C$ .

Um estado da cadeia  $C$  (nos pontos embutidos<sup>2</sup>) corresponde ao vector  $\vec{n}^C = (n_V^C, n_D^C)$ , onde a primeira componente identifica o número de pares de células de voz armazenadas no buffer de voz e a segunda componente corresponde ao número de células de dados contidas no buffer de dados. A matriz de probabilidades de transição associada à cadeia  $C$  é denominada por  $\mathbf{C}$ . O elemento  $\vec{n}_V^C, \vec{n}_D^C$  da matriz  $\mathbf{C}$ , denotado por  $\mathbf{C}[\vec{n}_V^C, \vec{n}_D^C]$ , corresponde à probabilidade que, ao início de um determinado ciclo, o multiplexador esteja no estado  $\vec{n}_D^C$ , dado que no início do ciclo anterior o multiplexador se encontrava no estado  $\vec{n}_V^C$ .

A análise matemática que permite obter o elemento  $\mathbf{C}[\vec{n}_V^C, \vec{n}_D^C]$  será dividida em três partes, que correspondem à análise do que ocorre durante a Fase de Inatividade, a Fase de Voz e a Fase de Dados.

<sup>2</sup> Sendo o ponto embutido o momento de término da transmissão de um par de células.

Em primeiro lugar, para modelar o comportamento do sistema na Fase de Inatividade (FI), define-se a cadeia de Markov  $E$  de parâmetro discreto, cujas transições ocorrem nos instantes em que se inicia e termina uma FI (ver Figura IV.24).

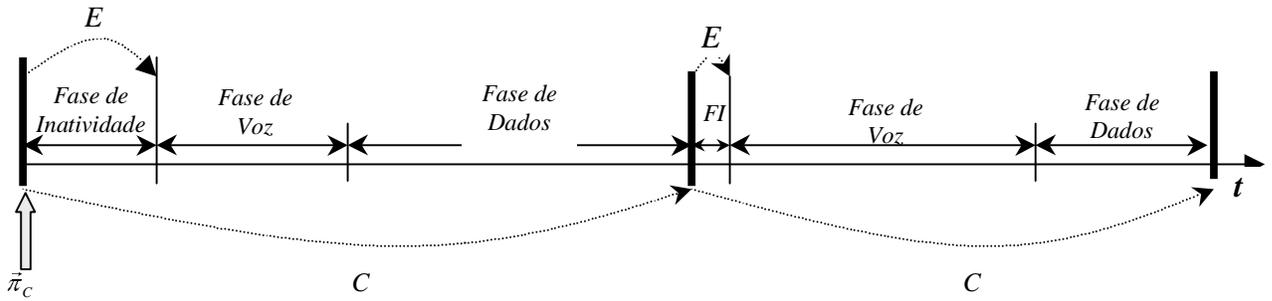


Figura IV.24. Pontos embutidos nos quais se definem as Cadeias de Markov  $C$  e  $E$ .

Um estado da cadeia  $E$  (nos instantes definidos anteriormente) corresponde ao vector  $\vec{n}^E = (n_V^E, n_D^E)$ , onde a primeira componente é o número de pares de células de voz no buffer de voz e a segunda é o número de células de dados que estão no buffer de dados. A matriz de probabilidades de transição associada à cadeia  $E$  é denominada  $\mathbf{E}$ .

Em segundo lugar, para modelar o comportamento do sistema na fase de voz, é definida a cadeia de Markov  $V$  de parâmetro discreto, nos pontos embutidos correspondentes aos instantes em que se inicia e em que termina uma fase de voz (ver Figura IV.25). Um estado da cadeia  $V$  (nos pontos embutidos) corresponde ao vector  $\vec{n}^V = (n_V^V, n_D^V)$ , onde a primeira componente denota o número de pares de células de voz presentes no buffer de voz e a segunda componente ao número de células de dados contidas no buffer de dados. A matriz de probabilidades de transição associada à cadeia  $V$  denomina-se  $\mathbf{V}$ .

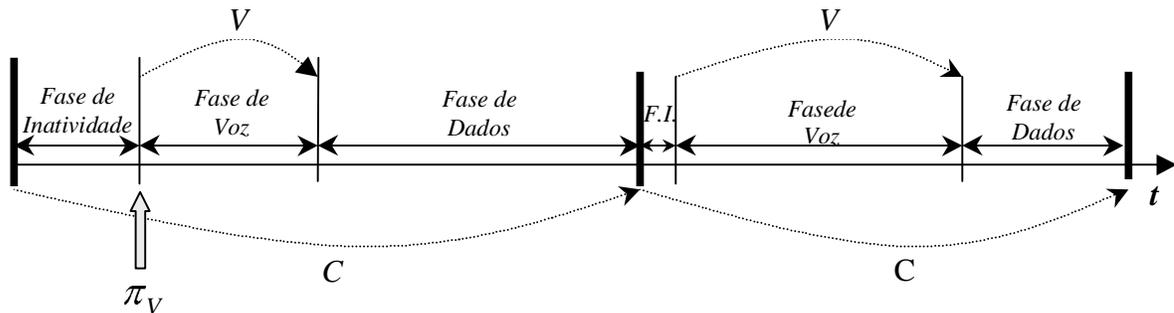


Figura IV.25. Pontos embutidos nos quais se definem as Cadeias de Markov  $C$  e  $V$ .

Finalmente, para modelar o comportamento do sistema na fase de dados, se define a cadeia de Markov de parâmetro discreto  $D$  (ver Figura IV.26), nos pontos embutidos correspondentes aos instantes em que se inicia e termina uma fase de dados. Um estado da cadeia  $D$  (nos pontos embutidos) corresponde ao vector  $\vec{n}^D = (n_V^D, n_D^D)$ , onde a primeira componente desse vector é o número de pares de células de voz presentes no buffer de voz e a segunda componente corresponde ao número de células de dados armazenadas no buffer de dados. A matriz de probabilidades de transição associada à cadeia  $D$  é denominada  $\mathbf{D}$ .

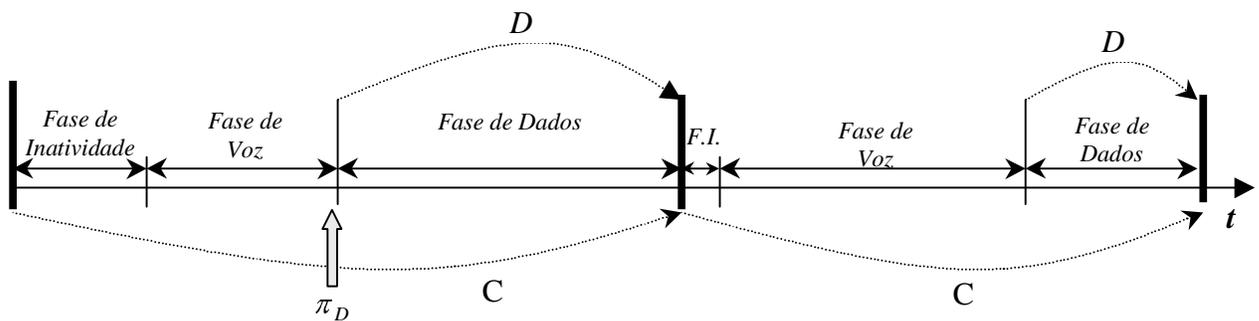


Figura IV.26. Pontos embutidos nos quais se definem as Cadeias de Markov  $C$  e  $D$ .

Das definições das cadeias  $E$ ,  $C$ ,  $V$  e  $D$  é fácil ver que os elementos da matriz  $\mathbf{C}$  são obtidos a través da seguinte equação matricial:

$$\mathbf{C} = \mathbf{E} \times \mathbf{V} \times \mathbf{D} \quad (\text{IV.62})$$

O autovector esquerdo da matriz  $\mathbf{C}$ , denominado  $\vec{\pi}_C$ , é obtido através da equação:

$$\vec{\pi}_C = \vec{\pi}_C \mathbf{C} \quad (\text{IV.63})$$

Seja  $\vec{\pi}_V$  o vector de probabilidade estacionária do estado do multiplexador no início da fase de voz. Da Figura IV.24 pode ser observado que:

$$\vec{\pi}_V = \vec{\pi}_C \mathbf{E} \quad (\text{IV.64})$$

Por outro lado, é definido  $\vec{\pi}_D$  como o vector de probabilidade estacionária do estado do multiplexador, no início da fase de dados. Da Figura IV.25 observa-se que:

$$\vec{\pi}_D = \vec{\pi}_V \mathbf{V} \quad (\text{IV.65})$$

Como resumo, na Figura IV.27 são mostradas as cadeias de Markov e os vetores de probabilidade estacionária até aqui definidos. Na figura, as setas curvas indicam os instantes nos quais acontecem as transições das cadeias  $C$ ,  $E$ ,  $V$  e  $D$ . Já as flechas verticais mostram os instantes nos quais são avaliados os vetores de probabilidade estacionária  $\vec{\pi}_C$ ,  $\vec{\pi}_V$  e  $\vec{\pi}_D$ . As setas verticais superiores correspondem ao vetor de estado do multiplexador no início de um determinado ciclo. Especificamente,  $\vec{n}^C$ ,  $\vec{n}^V$  e  $\vec{n}^D$ .

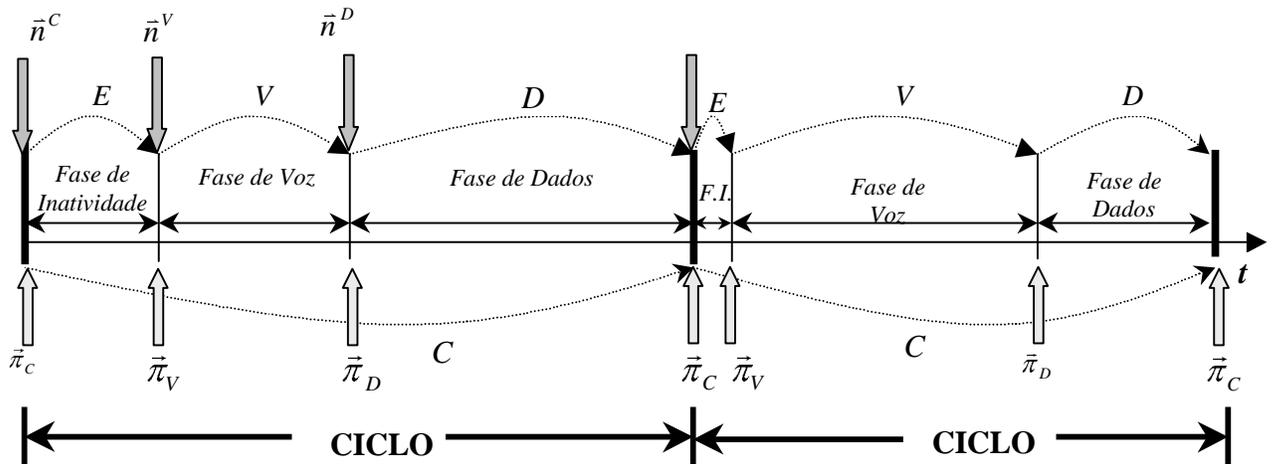


Figura IV.27 Cadeias de Markov  $C$ ,  $E$ ,  $V$  e  $D$ , e vetores de probabilidade  $\vec{\pi}_C$ ,  $\vec{\pi}_V$  e  $\vec{\pi}_D$ .

Na Seção IV.3.2.1 são obtidos os elementos da matriz  $\mathbf{E}$ . A seguir, na Seção IV.3.2.2, são obtidos os elementos da matriz  $\mathbf{V}$  e, na Seção IV.3.2.3, os elementos da matriz  $\mathbf{D}$ .

#### IV.3.4.1. Elementos da Matriz $\mathbf{E}$

Seja  $\mathbf{E}[\vec{n}^C, \vec{n}^V]$  o elemento  $[\vec{n}^C, \vec{n}^V]$  da matriz estocástica  $\mathbf{E}$ . Por definição da matriz  $\mathbf{E}$ , o elemento  $\mathbf{E}[\vec{n}^C, \vec{n}^V]$  é igual à probabilidade de, ao final da fase de inatividade, o multiplexador se encontrar no estado  $\vec{n}^V$ , dado que no início dessa fase estava no estado  $\vec{n}^C$ .

Para avaliar essa probabilidade, denotada por  $\Pr\{\bar{n}^V \mid \bar{n}^C\}$ , os seguintes casos são diferenciados:

- Um ciclo se inicia em um estado  $\bar{n}^C \neq \bar{0}$ . Nessa situação, o multiplexador inicia imediatamente a fase de voz (não existe inatividade), motivo pelo qual a fase de voz começa no mesmo estado  $\bar{n}^C$ , com probabilidade igual a 1.
- Um ciclo começa no estado  $\bar{0}$ . Nesse caso, o multiplexador permanece inativo até que chegue a primeira célula. A partir desse momento começa a fase correspondente ao tipo de célula que chegou. Como a chegada dos pares de células de voz é modelada como um processo Poisson de parâmetro  $\lambda_v$  e a chegada de células de dados é modelada como um processo de Poisson de parâmetro  $\lambda_d$ , pode-se concluir que a probabilidade da primeira célula ser uma célula de dados é dada por:  $\lambda_d/(\lambda_v+\lambda_d)$ . Analogamente, a probabilidade dela ser um par de células de voz é  $\lambda_v/(\lambda_v+\lambda_d)$ .

Usando as observações anteriores, são obtidos os elementos da matriz **E**:

$$E[\bar{n}^C, \bar{n}^V] = \begin{cases} \frac{\lambda_v}{(\lambda_v + \lambda_d)}; & [\bar{n}^C, \bar{n}^V] = [(0,0);(1,0)] \\ \frac{\lambda_d}{(\lambda_v + \lambda_d)}; & [\bar{n}^C, \bar{n}^V] = [(0,0);(0,1)] \\ 1; & \bar{n}^C = \bar{n}^V \neq 0 \\ 0; & \text{em outro caso} \end{cases} \quad (\text{IV.67})$$

#### IV.3.4.2 Elementos da Matriz V

Usar-se-á o termo “par de células de voz atendido” como uma referência ao caso no qual efetivamente sejam transmitidas as duas células do par, ou ao caso quando há congestionamento, quando somente a célula de maior prioridade é transmitida.

Seja  $\mathbf{V}[\bar{n}^V; \bar{n}^D]$  o elemento  $[\bar{n}^V; \bar{n}^D]$  da matriz estocástica **V**. Por definição da matriz **V**, o elemento  $\mathbf{V}[\bar{n}^V; \bar{n}^D]$  é igual a probabilidade do multiplexador se encontrar no estado  $\bar{n}^D$  ao

final da fase de voz,, dado que no início dessa fase estava no estado  $\vec{n}^V$ , denotada por  $\Pr\{\vec{n}^D \mid \vec{n}^V\}$ .

Observe que os elementos  $\mathbf{V}[\vec{0}, \vec{n}^D]$  da matriz  $\mathbf{V}$ , não ocorrem, uma vez que a fase de voz nunca começa no estado  $\vec{n}^V = (0,0)$ . Quando o ciclo começar no estado  $\vec{0}$ , a fase de inatividade durará até a chegada da primeira célula, fazendo com que a fase de voz se inicie no estado  $(1,0)$  ou  $(0,1)$ . Note então que a linha  $\vec{0}$  da matriz  $\mathbf{V}$  não é estritamente necessária, mas, para manter um dimensionamento matricial válido, manteve-se essa linha na equação (IV.62). A única restrição a ser feita é que a fila  $\vec{0}$  da matriz  $\mathbf{V}$  seja estocástica. Por este motivo, arbitrariamente define-se que:

$$\Pr\{\vec{n}^D \mid \vec{0}\} = \begin{cases} 1; & \vec{n}^D = (0,0) \\ 0; & \vec{n}^D \neq (0,0) \end{cases} \quad (\text{IV.69})$$

Para os casos restantes, isto é, para quando  $\vec{n}^V \neq (0,0)$ , a probabilidade  $\Pr\{\vec{n}^D \mid \vec{n}^V\}$  é avaliada através da seguinte análise matemática.

Seja  $t_v$  a variável aleatória que denota a duração da fase de voz. Para um determinado vector  $\vec{n}^V$ , o domínio de valores possíveis de  $t_v$  é  $[\text{mín}(n_v^V, L_v)D_1, 2L_vD_1]$ . Para simplificar a notação,  $L_v(\vec{n}^V)$  será escrito como  $L_v$ . Esta simplificação será mantida pelo restante deste capítulo.

Condicionando e descondicionando na duração da fase de voz, é obtida a seguinte expressão:

$$\Pr\{\vec{n}^D \mid \vec{n}^V\} = \sum_{i=\text{mín}[n_v^V, L_v]}^{2L_v} \Pr\{\vec{n}^D \mid \vec{n}^V, t_v = iD_1\} \Pr\{t_v = iD_1\} \quad (\text{IV.70})$$

Observe que, dada a duração da fase de voz, o número de células de dados ao final da fase de voz é independente do número de pares de células de voz no mesmo instante. Portanto,  $\Pr\{\vec{n}^D \mid \vec{n}^V, t_v = iD_1\}$  pode ser separada de seguinte forma:

$$\Pr\{\vec{n}^D \mid \vec{n}^V, t_v = iD_1\} = \Pr\{n_D^D \mid \vec{n}^V, t_v = iD_1\} \Pr\{n_V^D \mid \vec{n}^V, t_v = iD_1\} \quad (\text{IV.71})$$

Substituindo (IV.71) em (IV.70) tem-se a expressão:

$$\Pr\{\vec{n}^D \mid \vec{n}^V\} = \sum_{\tau=\min\{n_V^V, L_V\}}^{2L_V} \sum_{\forall \tau} \Pr\{n_V^D \mid \vec{n}^V, t_v = iD_1\} \Pr\{n_D^D \mid \vec{n}^V, t_v = iD_1\} \Pr\{t_v = iD_1\} \quad (\text{IV.72})$$

Da definição de probabilidade condicional tem-se que:

$$\Pr\{n_V^D \mid \vec{n}^V, t_v = iD_1\} \Pr\{t_v = iD_1\} = \Pr\{n_V^D, t_v = iD_1 \mid \vec{n}^V\} \quad (\text{IV.73})$$

Substituindo (IV.73) em (IV.72) chega-se a:

$$\Pr\{\vec{n}^D \mid \vec{n}^V\} = \sum_{i=\min\{n_V^V, L_V\}}^{2L_V} \Pr\{n_V^D, t_v = iD_1 \mid \vec{n}^V\} \Pr\{n_D^D \mid \vec{n}^V, t_v = iD_1\} \quad (\text{IV.74})$$

Para avaliar  $\Pr\{n_V^D, t_v = \tau \mid \vec{n}^V\}$  é preciso conhecer o comportamento do sistema durante a fase de voz. Para isso, define-se a cadeia de Markov de parâmetro discreto  $W$  (ver Figura IV.28) que evolui somente durante aquela fase. As transições de  $W$  ocorrem cada vez que o multiplexador termina de atender um par de células de voz. Um estado da cadeia  $W$  corresponde ao número de pares de células de voz presentes no multiplexador ao finalizar o atendimento de um par de células. O estado inicial da cadeia  $W$  corresponde ao número de pares de células de voz presentes no multiplexador no momento em que começa a fase de voz. A matriz de probabilidades de transição associada à cadeia  $W$  se denomina  $\mathbf{W}$ . Note que, de acordo com a definição recém dada, o estado da cadeia  $W$ , ao finalizar a fase de voz, é igual à primeira componente do estado da cadeia  $V$  no mesmo instante (ver Figura IV.28).

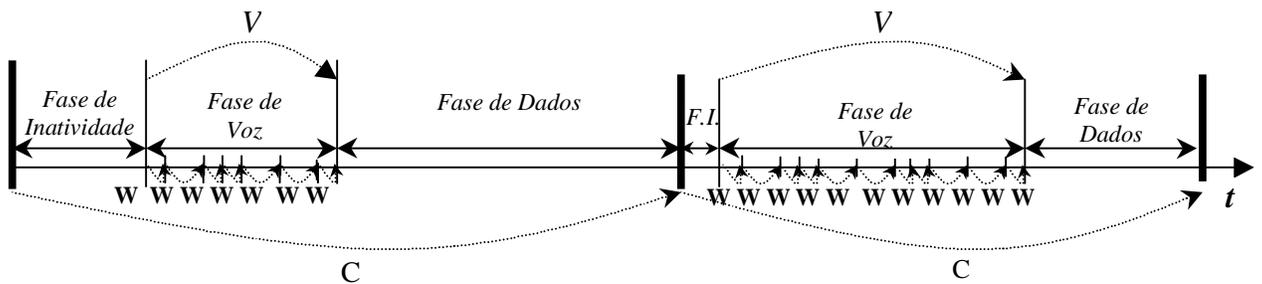


Figura IV.28. Pontos embutidos nos quais se definem as Cadeias de Markov  $C$ ,  $V$  e  $W$ .

A Figura IV.29 mostra a matriz de probabilidades de transição  $\mathbf{W}$ .

$$\begin{array}{c}
\mathbf{0} \quad \mathbf{1} \quad \mathbf{2} \quad \dots \quad \mathbf{u} \quad \mathbf{u+1} \quad \dots \quad \mathbf{K_v-2} \quad \mathbf{K_v-1} \quad \mathbf{K_v} \\
\left[ \begin{array}{cccccccccccc}
\mathbf{0} & 1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\
\mathbf{1} & a_v(0) & a_v(1) & a_v(2) & \dots & a_v(i) & a_v(i+1) & \dots & a_v(K_v-2) & 1-\Sigma & 0 \\
\mathbf{2} & 0 & a_v(0) & a_v(1) & \dots & a_v(i-1) & a_v(i) & \dots & a_v(K_v-3) & 1-\Sigma & 0 \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\mathbf{u} & 0 & 0 & 0 & \dots & a_v(1) & a_v(2) & \dots & a_v(K_v-u-1) & 1-\Sigma & 0 \\
\mathbf{u+1} & 0 & 0 & 0 & \dots & b_v(0) & b_v(1) & \dots & b_v(K_v-u-2) & 1-\Sigma & 0 \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\mathbf{K_v-1} & 0 & 0 & 0 & \dots & 0 & 0 & \dots & b_v(0) & 1-\Sigma & 0 \\
\mathbf{K_v} & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & 0 \\
\cdot & \cdot
\end{array} \right]
\end{array}$$

Figura IV.29 Matriz de probabilidades de transição  $\mathbf{W}$ .

onde o termo  $a_v(j)$  corresponde à probabilidade que cheguem  $j$  pares de células de voz durante um intervalo de tempo  $D_2$  (ver equação IV.61). Analogamente, o termo  $b_v(j)$  corresponde à probabilidade que cheguem  $j$  pares de células de voz durante um intervalo de tempo  $D_1$  (ver equação IV.61). Dado que o processo de chegada de pares de células de voz é modelado como um processo de Poisson de parâmetro  $\lambda_v$ , as expressões para  $a_v(j)$  e  $b_v(j)$  são dadas por:

$$a_v(j) = e^{-\lambda_v D_2} \frac{(\lambda_v D_2)^j}{j!} \quad (\text{IV.75})$$

$$b_v(j) = e^{-\lambda_v D_1} \frac{(\lambda_v D_1)^j}{j!} \quad (\text{IV.76})$$

Note que, nos instantes em que a cadeia  $W$  efetua transições, não é possível que o buffer de voz fique totalmente ocupado, já que nesse instante o par de células que foi servido abandona o buffer. Por este motivo, a coluna  $K_v$  da Figura IV.29 tem associada uma probabilidade igual a zero.

Por outro lado, devido a matriz  $\mathbf{W}$  ser uma matriz estocástica, o termo  $1-\Sigma$ , que aparece na

coluna  $(K_v-1)$  da fila  $i$ -ésima, corresponde a  $1 - \sum_{j=0}^{K_v-2} W[i, j]$ .

Por último, o estado 0 é definido arbitrariamente como um estado absorvente, no qual o elemento  $\mathbf{W}[0,0]$  tem associada uma probabilidade igual a 1. Com esse artifício matemático, consegue-se que o estado final do multiplexador corresponda exatamente ao estado da cadeia depois de  $L_v$  transições. Esta afirmação é evidente no caso em que a fase de voz dura  $L_v$  atendimentos. No caso em que a fase de voz dura menos que  $L_v$  atendimentos, isto é, no momento em que as células de voz se esgotam, o multiplexador aborta prematuramente a fase de voz. Nesse mesmo instante, a cadeia  $W$  entra no estado absorvente e, apesar da cadeia efetuar mais transições, o estado da cadeia continua sendo o estado 0. Desta análise pode ser concluído que agregar mais transições (para completar  $L_v$  transições) não altera o estado da cadeia, ao final da fase de voz. Esse artifício facilita a análise matemática da fase de voz, já que mantém constante o número de transições efetuada pela cadeia  $W$ .

Para avaliar  $\Pr\{n_v^D, t_v=\tau \mid n_v^V\}$ , são atribuídas recompensas às transições da cadeia  $W$ , de acordo com a seguinte função:

$$\eta(i, j) = \begin{cases} 0; & i = 0 \\ D_2; & 1 \leq i \leq u; i-1 \leq j < K_v \\ D_1; & u < i \leq K_v - 1; i-1 \leq j < K_v \end{cases} \quad (\text{IV.77})$$

onde  $\eta(i, j)$  corresponde a recompensa associada à transição que vai do estado  $i$  para o estado  $j$ .

Sejam  $RTAD(n)$  a variável aleatória que representa a recompensa total acumulada pela cadeia  $W$  em suas  $n$  primeiras transições, e  $g(n, r)$  a função de massa de probabilidade de  $RTAD(n)$ , isto é:

$$g(n, r) = \Pr\{RTAD(n) = r\} \quad (\text{IV.78})$$

Definindo  $g_f(n, r)$  como a probabilidade de  $RTAD(n)$  ser igual a  $r$  e assumindo que o estado da cadeia de Markov imediatamente depois da  $n$ -ésima transição seja  $e_f$ , é fácil comprovar que:

$$\Pr\{n_v^D, t_v = \tau \mid n_v^V\} = g_{n_v^D}(L_v, \tau) \quad (\text{IV.79})$$

onde o estado inicial de  $W$ , usado para avaliar  $g_{n_v^D}(L_v, \tau)$ , é o estado  $n_v^V$ .

Pela equação (IV.79), avaliar  $\Pr\{n_V^D, t_v=\tau \mid n_V^V\}$  é equivalente a avaliar  $g_{n_V^D}(L_v, \tau)$ . Por outro lado, para a avaliação de  $g_{n_V^D}(L_v, \tau)$  usa-se o método de [VAL93]. Dessas observações conclui-se então que para avaliar  $\Pr\{n_V^D, t_v=\tau \mid n_V^V\}$  é suficiente usar o método de cálculo existente na literatura. Com a finalidade de manter o texto desta tese autocontido, o método de avaliação de  $g_{n_V^D}(L_v, \tau)$  foi reproduzido na seção IV.1.

Prosseguindo com a avaliação das diferentes probabilidades que aparecem no lado direito da equação (IV.74), a seguir é obtida uma expressão que permite avaliar  $\Pr\{n_D^D \mid n_D^V, t_v=\tau\}$ . Dado que na avaliação desta probabilidade supõe-se que ao início da fase de voz existem  $n_D^V$  células de dados no multiplexador; que essa fase dura  $\tau$  unidades de tempo; que nela não são atendidas as células de dados; e que a chegada de células de dados é modelada como um processo de Poisson de parâmetro  $\lambda_d$ , chega-se a conclusão que:

$$\Pr\{n_D^D \mid n_D^V, t_v = \tau\} = \begin{cases} 0; & n_D^D < n_D^V \\ e^{-\lambda_d \tau} \frac{(\lambda_d \tau)^{(n_D^D - n_D^V)}}{(n_D^D - n_D^V)!}; & n_D^V \leq n_D^D < K_d \\ 1 - \sum_{j=0}^{K_d - n_D^V - 1} e^{-\lambda_d \tau} \frac{(\lambda_d \tau)^j}{j!}; & n_D^D = K_d \end{cases} \quad (\text{IV.80})$$

Finalmente, substituindo as equações (IV.79) e (IV.80) na equação (IV.74) é obtida uma expressão que permite avaliar  $\Pr\{\vec{n}^D \mid \vec{n}^V\}$  (ou seja, os elementos da matriz  $\mathbf{V}$ ):

$$V[\vec{n}^V, \vec{n}^D] = \begin{cases} 1; & \vec{n}^V = \vec{n}^D = \vec{0} \\ 0; & (n_D^D < n_D^V) \quad \vee \quad (\vec{n}^V = \vec{0} \wedge \vec{n}^D \neq \vec{0}) \\ \sum_{i=\min[n_V^V, L_v]}^{2L_v} g_{n_V^D}(L_v, iD_1) e^{-\lambda_d iD_1} \frac{(\lambda_d iD_1)^{(n_D^D - n_D^V)}}{(n_D^D - n_D^V)!}; & n_D^V \leq n_D^D < K_d \\ \sum_{i=\min[n_V^V, L_v]}^{2L_v} g_{n_V^D}(L_v, iD_1) \left[ 1 - \sum_{j=0}^{K_d - n_D^V - 1} e^{-\lambda_d iD_1} \frac{(\lambda_d iD_1)^j}{j!} \right]; & n_D^D = K_d \end{cases} \quad (\text{IV.81})$$

A primeira linha da equação (IV.81) corresponde ao estado absorvente definido arbitrariamente para o caso hipotético que a fase de voz comece no estado  $\vec{0}$ .

A segunda linha representa duas situações: 1) é impossível que a fase de voz finalize com menos células de dados do que as que tinha ao início dessa fase, 2) se a fase de voz começara no estado  $\vec{0}$ , não pode terminar em um estado diferente de  $\vec{0}$ .

A terceira linha corresponde àqueles casos nos quais o buffer de dados não fica saturado durante a fase de voz. Nesta equação o somatório considera todas as possíveis durações da fase de voz. O fator  $g_{n_v^D}(L_v, iD_1)$  captura o modo em que é modificado o número de pares de células de voz durante a fase. A probabilidade de tipo Poisson de parâmetro  $\lambda_d iD_1$  registra a variação no número de células de dados durante a fase voz.

A quarta linha é análoga a terceira, com a diferença que neste caso o buffer de dados fica saturado durante a fase de voz.

#### IV.3.4.3 Elementos da Matriz D

Em forma análoga ao caso da matriz **V**, o elemento  $\mathbf{D}[\vec{n}^D, \vec{n}^C]$  é dado por:

$$\mathbf{D}[\vec{n}^D, \vec{n}^C] = \mathbf{Pr}\{\vec{n}^C \mid \vec{n}^D\} \quad (\text{IV.82})$$

onde os vectores  $\vec{n}^D$  y  $\vec{n}^C$  correspondem ao estado do multiplexador ao início e ao final da fase de dados, respectivamente.

Depois de um raciocínio similar ao mostrado na seção IV.3.2.2, pode ser avaliada a probabilidade  $\mathbf{Pr}\{\vec{n}^C \mid \vec{n}^D\}$ , que é dada por:

$$\mathbf{Pr}\{\vec{n}^C \mid \vec{n}^D\} = \sum_{i=\min[n_D^D, L_d]}^{L_d} \mathbf{Pr}\{n_D^C, t_d = iD_1 \mid n_D^D\} \mathbf{Pr}\{n_V^C \mid n_V^D, t_d = iD_1\} \quad (\text{IV.83})$$

onde  $t_d$  corresponde à duração da fase de dados.

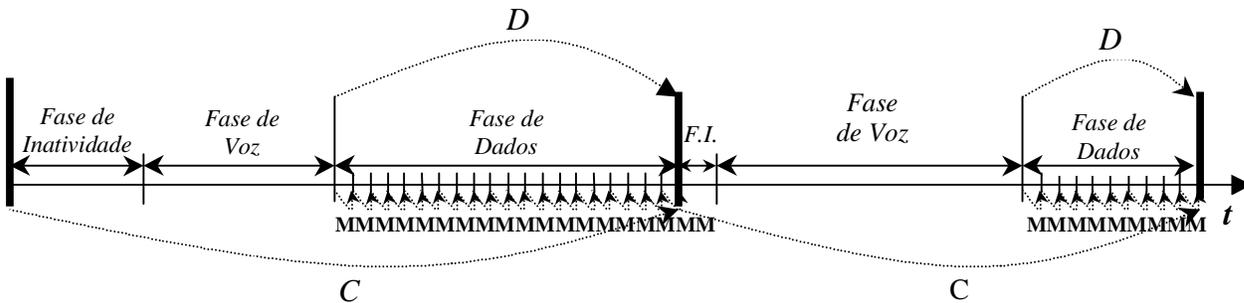


Figura IV.30 Pontos embutidos nos quais se definem as Cadeias de Markov  $C$ ,  $D$  e  $M$ .

Para avaliar  $\Pr\{n_D^C, t_d = iD_1 \mid n_D^D\}$ , é preciso modelar o comportamento do tráfego de dados durante a fase de dados. Para isto define-se a cadeia de Markov  $M$ , de parâmetro discreto (ver Figura IV. 30), que evoluciona somente durante a fase de dados e, que é análoga a cadeia  $W$  definida para a fase de voz. As transições de  $M$  acontecem nos instantes em que termina a atenção de uma célula de dados. Um estado da cadeia  $M$  corresponde ao número de células de dados presentes no multiplexador no instante em que ocorre uma transição. O estado inicial da cadeia  $M$  é o número de células de dados presentes no multiplexador no momento em que começa a fase de dados. O estado da cadeia  $M$ , ao final da fase de dados, corresponde ao número de células de dados que há no multiplexador ao final do ciclo. A matriz de probabilidades de transição associada a cadeia  $M$  é denotada  $\mathbf{M}$ .

A Figura IV.31 mostra a matriz de probabilidades de transição  $\mathbf{M}$ . Nessa matriz,  $p_d(j)$  corresponde à probabilidade de que durante o tempo  $D_1$  cheguem  $j$  células de dados. Como o processo de chegada de células de dados é modelado por um processo de Poisson de parâmetro  $\lambda_d$ , a expressão  $p_d(j)$  é a seguinte:

$$p_d(j) = e^{-\lambda_d D_1} \frac{(\lambda_d D_1)^j}{j!} \tag{IV.84}$$

O termo  $1-\Sigma$  é análogo ao mesmo termo da matriz  $\mathbf{W}$  e o estado absorvente (estado  $\vec{0}$ ) define-se tal qual foi feito para a matriz  $\mathbf{W}$ .

$$\begin{array}{c}
\mathbf{0} \\
\mathbf{1} \\
\mathbf{2} \\
\vdots \\
\vdots \\
\vdots \\
\mathbf{i} \\
\mathbf{i+1} \\
\vdots \\
\vdots \\
\mathbf{K_d-1} \\
\mathbf{K_d}
\end{array}
\left[ \begin{array}{ccccccccccc}
\mathbf{0} & \mathbf{1} & \mathbf{2} & \dots & \mathbf{i} & \mathbf{i+1} & \dots & \mathbf{K_d-2} & \mathbf{K_d-1} & \mathbf{K_d} \\
1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 \\
p_d(0) & p_d(1) & p_d(2) & \dots & p_d(i) & p_d(i+1) & \dots & p_d(K_d-2) & 1-\Sigma & 0 \\
0 & p_d(0) & p_d(1) & \dots & p_d(i-1) & p_d(i) & \dots & p_d(K_d-3) & 1-\Sigma & 0 \\
\vdots & \vdots \\
\vdots & \vdots \\
\mathbf{i} & 0 & 0 & \dots & p_d(1) & p_d(2) & \dots & p_d(K_d-u-1) & 1-\Sigma & 0 \\
\mathbf{i+1} & 0 & 0 & \dots & p_d(0) & p_d(1) & \dots & p_d(K_d-u-2) & 1-\Sigma & 0 \\
\vdots & \vdots \\
\mathbf{K_d-1} & 0 & 0 & \dots & 0 & 0 & \dots & p_d(0) & 1-\Sigma & 0 \\
\mathbf{K_d} & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 & 0
\end{array} \right]$$

Figura IV.31 Matriz de probabilidades de transição  $\mathbf{M}$ .

De modo similar ao raciocínio feito na seção IV.3.4.2, para obter uma expressão que permita avaliar a probabilidade  $\Pr\{n_D^C, t_d = iD_1 \mid n_D^D\}$ , são atribuídas recompensas às transições da matriz  $M$  da seguinte maneira:

$$\eta(i, j) = \begin{cases} 0; & i = 0 \\ D_1 & i > 0 \end{cases} \quad (\text{IV.85})$$

onde  $\eta(i, j)$  corresponde a recompensa associada a transição que vai desde o estado  $i$  para o estado  $j$ .

Utilizando o mesmo raciocínio usado para obter a equação (IV.79), é obtida a expressão:

$$\Pr\{n_D^C, t_d = \tau \mid n_D^D, L_d\} = g_{n_D^C}(L_d, \tau) \quad (\text{IV.86})$$

onde  $g_{n_D^C}(L_d, \tau)$  corresponde a probabilidade de que a cadeia  $M$  esteja no estado  $n_D^C$  e que tenha acumulado uma recompensa total igual a  $\tau$  em  $L_d$  transições.

Por outro lado, para calcular  $\Pr\{n_V^C | n_V^D, t_d = iD_1\}$ , observa-se que no começo da fase de dados existem  $n_V^D$  pares de células de voz no multiplexador, que a fase de dados dura  $iD_1$  unidades de tempo, que nesta fase não são atendidas pares de células de voz e, que as chegadas de pares de células de voz são modeladas por um processo de Poisson de parâmetro  $\lambda_v$ . Destes antecedentes conclui-se que a probabilidade  $\Pr\{n_V^C | n_V^D, t_d = iD_1\}$  é dada pela seguinte expressão:

$$\Pr\{n_V^C | n_V^D, t_d = iD_1\} = \begin{cases} 0; & n_V^C < n_V^D \\ e^{-\lambda_v iD_1} \frac{(\lambda_v iD_1)^{(n_V^C - n_V^D)}}{(n_V^C - n_V^D)!}; & n_V^D \leq n_V^C < K_v \\ 1 - \sum_{j=0}^{K_v - n_V^D - 1} e^{-\lambda_v iD_1} \frac{(\lambda_v iD_1)^j}{j!}; & n_V^C = K_v \end{cases} \quad (\text{IV.87})$$

Substituindo as equações (IV.86) e (IV.87) na equação (IV.83) é obtida a expressão para  $\Pr\{\vec{n}^C | \vec{n}^D\}$  (que corresponde aos elementos da matriz  $\mathbf{D}$ ):

$$D[\vec{n}^D, \vec{n}^C] = \begin{cases} 1; & \vec{n}^V = \vec{n}^D = \vec{0} \\ 0; & n_V^C < n_V^D \\ \sum_{i=\min[n_V^D, L_d]}^{L_d} g_{n_V^C}(L_d, iD_1) e^{-\lambda_v iD_1} \frac{(\lambda_v iD_1)^{(n_V^C - n_V^D)}}{(n_V^C - n_V^D)!}; & n_V^D \leq n_V^C < K_v \\ \sum_{i=\min[n_V^D, L_d]}^{L_d} g_{n_V^C}(L_d, iD_1) \left[ 1 - \sum_{j=0}^{K_v - n_V^D - 1} e^{-\lambda_v iD_1} \frac{(\lambda_v iD_1)^j}{j!} \right]; & n_V^C = K_v \end{cases} \quad (\text{IV.88})$$

A primeira linha da equação (IV.88) corresponde ao estado absorvente definido arbitrariamente para o caso hipotético que a fase de dados comece no estado  $\vec{0}$ .

Na segunda linha descreve a situação na qual é impossível que na fase de dados o buffer de voz termine com menos células do que as que tinha ao início dessa fase.

A terceira linha corresponde ao caso em que o buffer de voz fica saturado durante a fase de dados. Nesta linha o somatório considera todas as possíveis durações da fase de dados. O fator  $g_{n_V^C}(L_d, iD_1)$  indica como modifica-se o número de células de dados durante a fase de

dados. A probabilidade de tipo Poisson de parâmetro  $\lambda_v i D_1$  registra como varia o número de células de voz durante a fase dados.

Finalmente, a quarta linha é análoga a terceira linha, exceto que o buffer de voz fica saturado durante a fase de dados.

### IV.3.5 Avaliação de Medidas de Desempenho

Nesta seção são avaliadas diversas medidas de valor médio, que caracterizam o comportamento do multiplexador de voz e dados especificado na seção IV.3.1. As medidas avaliadas são o tamanho médio do ciclo, o número médio de pares de células de voz que são armazenadas no buffer de voz, o número médio células de dados que contém o buffer de dados, o retardo médio experimentado por uma célula (de voz o de dados) no multiplexador e, a probabilidade de perda de células de voz o de dados.

#### IV.3.5.1 Tamanho Médio do Ciclo

Tal como foi visto na Seção III, um ciclo está composto por uma fase de inatividade, uma fase de voz e uma fase de dados. Em conseqüência, a duração de um ciclo corresponde à soma da duração de cada uma dessas fases.

Seja  $T$  o tamanho médio do ciclo e,  $T_E$ ,  $T_v$  e  $T_d$  o tempo médio que dura a fase de inatividade, a fase de voz e a fase de dados, respectivamente. Então:

$$T = T_E + T_v + T_d \quad (\text{IV.93})$$

##### IV.3.5.1.1 Avaliação de $T_E$

Devido a que a fase de inatividade dura um tempo maior que zero, somente no caso em que um ciclo começa no estado  $\bar{0}$  (situação que ocorre com probabilidade  $\bar{\pi}_c[0,0]$ ) e, que a chegada de pares de células de voz e células de dados são processos de Poisson de parâmetros  $\lambda_v$  e  $\lambda_d$ , respectivamente. Aplicando o teorema de probabilidades totais, conclui-se que:

$$T_E = \bar{\pi}_c[0,0] \frac{1}{\lambda_v + \lambda_d} \quad (\text{IV.94})$$

### IV.3.5.1.2 Avaliação de $T_v$

Como a duração da fase de voz depende do estado do multiplexador no início desta fase, obtém-se que:

$$T_v = \sum_{\forall \vec{n}^V} \bar{\pi}_v[\vec{n}^V] T_v(\vec{n}^V) \quad (\text{IV.95})$$

onde  $\bar{\pi}_v[\vec{n}^V]$  é a probabilidade em estado estacionário que a fase de voz comece no estado  $\vec{n}^V$ ;  $T_v(\vec{n}^V)$  corresponde a duração média da fase de voz, em caso que ao início desta fase o multiplexador esteja no estado  $\vec{n}^V$ .

Para avaliar  $T_v(\vec{n}^V)$  é utilizado o fato de que, este intervalo corresponde a soma dos valores médios dos tempos que o multiplexador demora, em cada uma das atenções que deve realizar, durante a fase de voz. Em geral, o número de atenções efetuadas nesta fase é igual a  $L_v$ , exceto nos casos em que os pares de células de voz se esgotem antes de atender os  $L_v$  pares inicialmente programados. Neste último caso,  $T_v(\vec{n}^V)$  deveria ser avaliado considerando o número exato de atenções efetuadas durante a fase de voz. Uma forma de fazer este cálculo é considerar que foram feitas  $L_v$  atenções, mas a cada atenção extra (nas quais a cadeia de Markov  $W$  permanece no estado absorvente) é atribuída uma duração de 0 unidades de tempo. Desta forma, o cálculo de  $T_v(\vec{n}^V)$  é correto. Matematicamente este artifício corresponde a modificar a equação (IV.61) da seguinte forma:

$$s_{n_v} = \begin{cases} 0; & n_v = 0 \\ D_2; & n_v = 1, 2, \dots, u \\ D_1; & n_v = u + 1, \dots, K_v \end{cases} \quad (\text{IV.96})$$

Resumindo, ao utilizar a equação (IV.96), ao invés da equação (IV.61), o cálculo de  $T_v(\vec{n}^V)$  sempre corresponde à soma de  $L_v$  segmentos de tempo, alguns dos quais podem ter um tamanho igual a zero. Em consequência:

$$T_v(\vec{n}^V) = \sum_{i=1}^{L_v} E[a_i] \quad (\text{IV.97})$$

onde a variável aleatória  $a_i$ ,  $1 \leq i \leq L_v$  é o tempo que o multiplexador demora na  $i$ -ésima atenção dentro da fase de voz e,  $E[a_i]$ ,  $1 \leq i \leq L_v$ , corresponde ao seu valor médio.

Dado que  $a_i$  é uma variável aleatória, cujo valor depende do estado do buffer de voz no momento em que começa a  $i$ -ésima atenção, seu valor médio é dado por:

$$E[a_i] = D_2 \sum_{e=1}^u \pi_{W_e}(i-1) + D_1 \sum_{e=u+1}^{K_v-1} \pi_{W_e}(i-1) + 0 \pi_{W_0}(i-1) \quad (\text{IV.98})$$

onde  $\pi_{W_e}(i)$  corresponde a probabilidade que a cadeia  $W$  se encontre no estado  $e$  imediatamente depois da  $i$ -ésima transição e, que a condição inicial está dada por:

$$\vec{\pi}_W(0) = \vec{1}_{n_v^V}.$$

Note que no caso em que a fase de voz termine prematuramente, a cadeia  $W$  entra no estado absorvente  $\vec{0}$ . Como é mostrado, no terceiro termo do lado direito da equação (IV.98), a partir deste instante os restantes segmentos da fase demoram zero unidades de tempo. Logo, substituindo a equação (IV.98) a equação (IV.97) obtém-se:

$$\bar{T}_v(\vec{n}^V) = \sum_{i=1}^{L_v} \left( D_2 \sum_{e=1}^u \pi_{W_e}(i-1) + D_1 \sum_{e=u+1}^{K_v-1} \pi_{W_e}(i-1) \right) \quad (\text{IV.99})$$

onde a condição inicial da cadeia  $W$  está dada por  $\vec{\pi}_W(0) = \vec{1}_{n_v^V}$ .

Finalmente, substituindo (IV.99) em (IV.95) é obtida a expressão final para  $T_v$ :

$$T_v = \sum_{\forall \vec{n}^V} \vec{\pi}_v[\vec{n}^V] \sum_{i=1}^{L_v} \left( D_2 \sum_{e=1}^u \pi_{W_e}(i-1) + D_1 \sum_{e=u+1}^{K_v-1} \pi_{W_e}(i-1) \right) \quad (\text{IV.100})$$

#### IV.3.5.1.3 Avaliação de $T_d$

Em virtude que a duração da fase de dados depende do estado do multiplexador no início desta fase, então:

$$T_d = \sum_{\forall \vec{n}^D} \vec{\pi}_d[\vec{n}^D] \bar{T}_d(\vec{n}^D) \quad (\text{IV.101})$$

onde  $\vec{\pi}_d[\vec{n}^D]$  é a probabilidade em estado estacionário de que a fase de dados comece no estado  $\vec{n}^D$ ;  $\bar{T}_d(\vec{n}^D)$  corresponde à duração média da fase de dados, no caso que ao início dessa fase o multiplexador se encontre no estado  $\vec{n}^D$ .

Com um raciocínio análogo ao realizado para obter  $T_v(\vec{n}^V)$ , conclui-se que:

$$T_d(\vec{n}^D) = \sum_{i=1}^{L_d} \sum_{\forall e \neq 0} \pi_{M_e}(i-1) D_1 = D_1 \sum_{i=1}^{L_d} (1 - \pi_{M_0}(i-1)) \quad (\text{IV.102})$$

onde  $\pi_{M_e}(i)$  corresponde à probabilidade que a cadeia  $M$  se encontre no estado  $e$  imediatamente após ter efetuado a  $i$ -ésima transição e, a condição inicial da cadeia está dada por  $\bar{\pi}_M(0) = \bar{1}_{n_D}$ .

Substituindo (IV.102) em (IV.101) é obtida a expressão final para  $T_d$ :

$$T_d = \sum_{\forall \bar{n}^D} \bar{\pi}_d[\bar{n}^D] D_1 \sum_{i=1}^{L_d} (1 - \pi_{M_0}(i-1)) \quad (\text{IV.103})$$

### IV.3.5.2 Número médio de células

#### IV.3.5.2.1 Número médio de células de dados

Seja  $N_D$  o número médio de células de dados no período de um ciclo. Sejam  $N_D^V$  e  $N_D^D$  o número médio de células de dados que contém o multiplexador durante a fase de voz e de dados, respectivamente. Então, observando que durante a fase de inatividade não existem células de dados no multiplexador, e aplicando o teorema de probabilidades totais, obtém-se:

$$N_D = \frac{T_v}{T} N_D^V + \frac{T_d}{T} N_D^D \quad (\text{IV.104})$$

onde  $\frac{T_v}{T}$  e  $\frac{T_d}{T}$  correspondem às probabilidades de que o multiplexador se encontre nas fases de voz e de dados, respectivamente.

##### IV.3.5.2.1.1 Avaliação de $N_D^V$

Dado a que  $N_D^V$  depende do estado do multiplexador no início da fase de voz, é obtido que:

$$N_D^V = \sum_{\forall n} \bar{\pi}_v[\bar{n}^V] N_D^V(\bar{n}^V) \quad (\text{IV.105})$$

onde  $\bar{\pi}_v[\bar{n}^V]$  é a probabilidade em estado estacionário que a fase de voz comece no estado  $\bar{n}^V$ ;  $N_D^V(\bar{n}^V)$  corresponde ao número médio de células de dados durante a fase de voz, dado que a fase de voz inicia-se no estado  $\bar{n}^V$ .

Para obter  $N_D^V(\bar{n}^V)$  condiciona-se nas  $j$  células de dados recebidas durante a fase de voz e, aplica-se o teorema de probabilidades totais:

$$N_D^V(\bar{n}^V) = \sum_{j=0}^{\infty} p(j) N_D^V(\bar{n}^V | j) \quad (\text{IV.106})$$

onde  $p(j)$  corresponde a probabilidade de que durante a fase de voz cheguem  $j$  células de dados. Devido a que  $p(j)$  distribui-se segundo uma variável aleatória Poisson de parâmetro  $\lambda_d$ , para avaliar (IV.106) é necessário conhecer a duração da fase de voz. Por este motivo, a equação (IV.106) condiciona-se e descondiciona-se no valor  $\tau$  de duração da fase de voz, resultando em:

$$N_D^V(\bar{n}^V) = \sum_{\forall \tau} \Pr\{t_v = \tau | \bar{n}^V\} \sum_{j=0}^{\infty} p(j | \tau) N_D^V(\bar{n}^V, j, \tau) \quad (\text{IV.107})$$

onde  $\Pr\{t_v = \tau | \bar{n}^V\}$  corresponde à probabilidade de que a fase de voz dure  $\tau$  unidades de tempo, dado que o estado do multiplexador ao início da fase de voz é  $\bar{n}^V$ . O termo  $N_D^V(\bar{n}^V, j, \tau)$  é o número médio de células de dados durante a fase de voz, dado que o estado do multiplexador ao início desta fase é  $\bar{n}^V$ , que chegam  $j$  células de dados durante a fase e, que a fase dura  $\tau$  unidades de tempo. Como  $N_D^V(\bar{n}^V, j, \tau)$  depende apenas de  $\bar{n}^V$  e  $j$ , será anotado simplesmente por  $N_D^V(\bar{n}^V, j)$ . Esta observação permite rescrever a equação (IV.107) como:

$$N_D^V(\bar{n}^V) = \sum_{\forall \tau} \Pr\{t_v = \tau | \bar{n}^V\} \sum_{j=0}^{\infty} p(j | \tau) N_D^V(\bar{n}^V, j) \quad (\text{IV.108})$$

Para avaliar  $N_D^V(\bar{n}^V, j)$  podem ser distinguidas duas situações: a primeira, corresponde ao caso no qual o buffer de dados não fica saturado com a chegada das  $j$  células de dados; a segunda, acontece quando o buffer de dados se satura com a chegada destas células. Separando o lado direito da equação (IV.108) para essas situações e considerando que as chegadas de células de dados são modeladas por um processo Poisson de parâmetro  $\lambda_d$ , é obtida a seguinte expressão:

$$\sum_{j=0}^{\infty} N_D^V(\bar{n}^V, j) p(j | \tau) = \sum_{j=0}^{K_d - \bar{n}_D^V - 1} N_D^V(\bar{n}^V, j) e^{-\lambda_d \tau} \frac{(\lambda_d \tau)^j}{j!} + \sum_{j=K_d - \bar{n}_D^V}^{\infty} N_D^V(\bar{n}^V, j) e^{-\lambda_d \tau} \frac{(\lambda_d \tau)^j}{j!} \quad (\text{IV.109})$$

O primeiro termo do lado direito da equação (IV.109) representa as situações nas quais o buffer de dados não entra em uma condição de saturação, durante a fase de voz. O segundo

termo, corresponde aquelas situações nas que o buffer de voz fica saturado, durante a fase de dados.

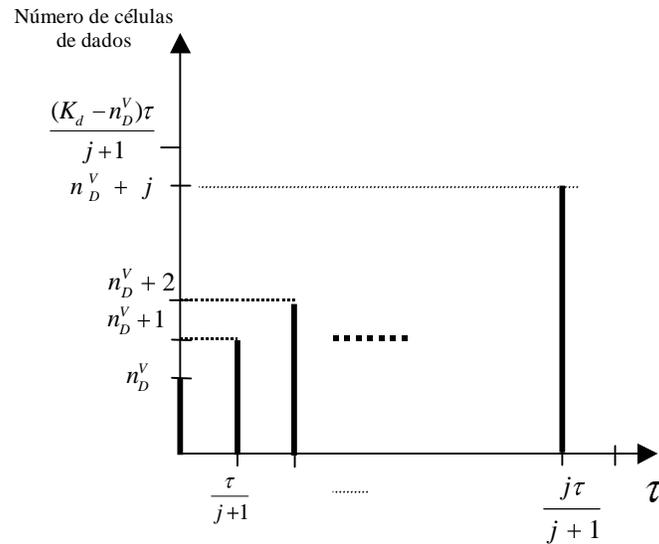


Figura IV.32. Número de células de dados durante a fase de voz, quando não existe saturação do buffer de dados.

Para calcular  $N_D^V(\bar{n}^V, j)$  em cada um dos casos acima descritos, considere os gráficos IV.32 e IV.33, que mostram o número de pares de células de dados que estão no multiplexador durante a fase de voz.

Observe que, durante a fase de voz, o número de pares de células de dados pode aumentar, uma vez que elas não são transmitidas durante esta fase.

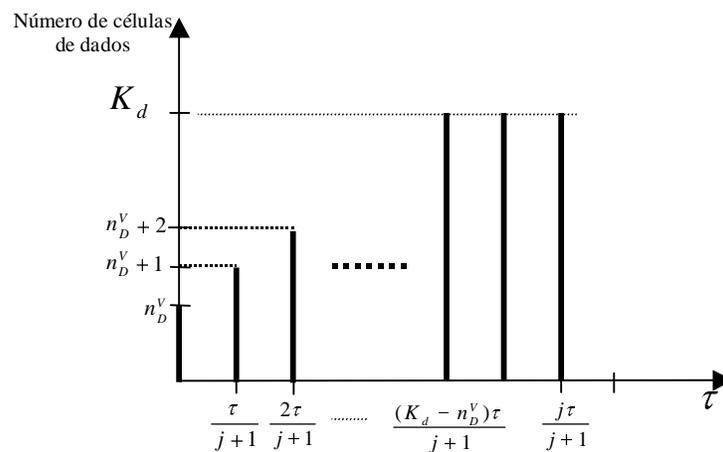


Figura IV.33 Número de células de dados durante a fase de voz, quando o buffer de dados fica saturado.

Nos gráficos das Figuras IV.32 e IV.33, o eixo das abcissas corresponde ao tempo que dura a fase de voz e o eixo das ordenadas corresponde ao número de células de dados presentes no multiplexador. Ambos gráficos representam o fato que a fase de voz dura  $\tau$  unidades de tempo e que durante ela chegam  $j$  células de dados. Ainda, em ambos gráficos, no eixo das abcissas é mostrado que em valor médio, os instantes de chegada das  $j$  células de dados dividem o intervalo  $[0, \tau]$  (isto é, a fase de voz) em  $(j+1)$  segmentos de igual tamanho [KAR81] (ver também análise da seção IV.2). Uma consequência disto é que os  $(j+1)$  segmentos, definidos pelas  $j$  chegadas, têm a mesma distribuição. Portanto, o tamanho médio de cada um de estos  $(j+1)$  segmentos é igual a  $\frac{\tau}{j+1}$ .

Para avaliar  $N_D^V(\bar{n}^V, j)$ , com  $0 \leq j \leq (K_d - n_D^V)$ , observa-se que este valor médio corresponde a altura média do gráfico da Figura IV.32. Desse modo:

$$N_D^V(\bar{n}^V, j) = \frac{\sum_{i=0}^j (n_D^V + i)}{j+1}; \quad 0 \leq j < (K_d - n_D^V) \quad (\text{IV.110})$$

Desenvolvendo e simplificando o somatório do numerador, obtém-se que:

$$N_D^V(\bar{n}^V, j) = n_D^V + \frac{j}{2}; \quad 0 \leq j < (K_d - n_D^V) \quad (\text{IV.111})$$

Usando a Figura IV.32 e, um raciocínio similar ao usado anteriormente, conclui-se que:

$$N_D^V(\bar{n}^V, j) = \frac{\sum_{i=0}^{K_d - n_D^V - 1} (n_D^V + i) + [j - (K_d - n_D^V - 1)]K_d}{j+1}; \quad j \geq (K_d - n_D^V + 1) \quad (\text{IV.112})$$

Logo, desenvolvendo o somatório e agrupando termos, conclui-se que:

$$N_D^V(\bar{n}^V, j) = K_d - \frac{(K_d - n_D^V + 1)(K_d - n_D^V)}{2(j+1)}; \quad j \geq (K_d - n_D^V + 1) \quad (\text{IV.113})$$

Substituindo as equações (IV.113), (IV.111) e (IV.109) na equação (IV.108) chega-se a:

$$N_D^V(\bar{n}^V) = \sum_{\forall \tau} \Pr \left\{ t_v = \tau \mid \bar{n}^V \right\} \left[ \sum_{j=0}^{K_d - n_D^V - 1} \binom{n_D^V + j}{j} e^{-\lambda_d \tau} \frac{(\lambda_d \tau)^j}{j!} + \sum_{j=K_d - n_D^V}^{\infty} \left( K_d - \frac{(K_d - n_D^V)(K_d - n_D^V + 1)}{2(j+1)} \right) e^{-\lambda_d \tau} \frac{(\lambda_d \tau)^j}{j!} \right] \quad (\text{IV.114})$$

Para obter  $\Pr\{t_v = \tau | \vec{n}^v\}$ , atribuem-se recompensas às transições da cadeia  $W$  (usando a mesma função de atribuições descrita na equação (IV.86)) e, utiliza-se o método explicado na seção IV.1. Com o qual é obtida a expressão:

$$\Pr\{t_v = \tau | \vec{n}^v\} = g(L_v, \tau) \quad (\text{IV.115})$$

onde  $g(L_v, \tau)$  corresponde a probabilidade de que a recompensa total acumulada pela cadeia  $W$  em  $L_v$  transições seja igual a  $\tau$ .

Lembrando que o domínio de possíveis valores de  $t_v$  é  $[\min(n_v^v, L_v)D_1, L_v D_2]$ , e substituindo (IV.115) em (IV.114), e (IV.114) em (IV.105) conclui-se que:

$$N_D^v = \sum_{\forall \vec{n}^v} \bar{\pi}_v[\vec{n}^v] \sum_{\substack{\forall \tau \\ \min[n_v^v, L_v]D_1 \leq \tau \leq L_v D_2}} g(L_v, \tau) \left[ \sum_{j=0}^{K_d - n_D^v - 1} \binom{n_D^v + j}{2} e^{-\lambda_d i D_1} \frac{(\lambda_d i D_1)^j}{j!} + \sum_{j=K_d - n_D^v}^{\infty} \left( K_d - \frac{(K_d - n_D^v)(K_d - n_D^v + 1)}{2(j+1)} \right) e^{-\lambda_d i D_1} \frac{(\lambda_d i D_1)^j}{j!} \right]$$

#### IV.3.5.2.1.2 Avaliação de $N_D^D$

Notando que  $N_D^D$  depende do estado do multiplexador ao início da fase de dados, observa--se que:

$$N_D^D = \sum_{\forall \vec{n}^D} \bar{\pi}_D[\vec{n}^D] N_D^D(\vec{n}^D) \quad (\text{IV.116})$$

onde  $\bar{\pi}_D[\vec{n}^D]$  corresponde a probabilidade em estado estacionário de que a fase de dados comece no estado  $\vec{n}^D$ ;  $N_D^D(\vec{n}^D)$  corresponde ao número médio de células de dados durante a fase de dados, sempre que esta fase inicia-se no estado  $\vec{n}^D$ .

Para calcular  $N_D^D(\vec{n}^D)$  considere o gráfico da Figura IV.34:



Figura IV.34. Número de pares de células de dados nos instantes em que termina a atenção de uma célula de dados versus tempo

Na Figura IV.34, no eixo das abcissas o tamanho do segmento  $t_i$  representa o tempo que demora a  $i$ -ésima atenção,  $1 \leq i \leq L_d$ . Nesta análise considera-se que a fase de dados está sempre composta de  $L_d$  segmentos, ainda que sejam atendidas menos de  $L_d$  células de dados durante esta fase. Por isto, na avaliação de  $N_D^D(\vec{n}^D)$  utilizar-se-á o mesmo artifício matemático usado em IV.3.4.1. Por outro lado, no eixo das ordenadas, é mostrado o número células de dados que há no multiplexador nos instantes em que é iniciada a atenção de uma célula de dados.

A partir da Figura IV.34 fica claro que:

$$N_D^D(\vec{n}^D) = \frac{\sum_{i=1}^{L_d} N_i}{L_d} \quad (\text{IV.117})$$

onde  $N_i$  corresponde ao número médio de células de dados presentes no multiplexador ao iniciar o  $i$ -ésimo segmento.

Para calcular  $N_i$  condiciona-se no número  $e$  de células de dados contidas no multiplexador no início do segmento. Aplicando o teorema de probabilidades totais obtém-se:

$$N_i = \sum_{e=1}^{K_d} \pi_{M_e}(i-1)e \quad (\text{IV.118})$$

onde  $\pi_{M_e}(n)$  é a probabilidade de que a cadeia  $M$  se encontre no estado  $e$  imediatamente após a  $n$ -ésima transição. A condição inicial da cadeia  $M$  é  $\vec{\pi}_M(0) = \vec{1}_{n_D^D}$ .

Substituindo (IV.118) em (IV.117) e, (IV.117) em (IV.116) é obtido que:

$$N_D^D = \sum_{\forall \vec{n}^D} \vec{\pi}_D[\vec{n}^D] \frac{1}{L_d} \sum_{i=1}^{L_d} \sum_{e=1}^{K_d} e \pi_{M_e}(i-1) \quad (\text{IV.119})$$

onde  $\pi_{M_e}(n)$  é a probabilidade que a cadeia  $M$  fique no estado  $e$  imediatamente após da  $n$ -ésima transição.

Note-se que, nas equações (IV.117) e (IV.118), o valor médio do número de células de dados é avaliado somente nos instantes em que começa a atenção de um par destas células. Isto significa que, o valor de  $N_D^D$  avaliado nesta seção, é igual ao número de células de dados

considerando somente os instantes de início do serviço das células de dados, o que é ligeiramente diferente ao número médio de células de dados que seria obtido ao considerar todos os instantes da fase.

Finalmente, substituindo as expressões de  $N_D^V$  e  $N_D^D$  em (IV.104) é obtida a expressão para o número médio de células de dados no ciclo, que é dada por:

$$\begin{aligned}
 N_V = & \frac{T_v}{T} \sum_{\forall \bar{n}^V} \bar{\pi}_V [\bar{n}^V] \frac{\sum_{i=1}^{L_v} \sum_{e=1}^{K_v} \pi_{W_e} (i-1) e}{L_v} + \\
 & \frac{T_d}{T} \sum_{\forall \bar{n}^D} \bar{\pi}_D [\bar{n}^D] \left[ \sum_{i=\min[\bar{n}^D, L_d]}^{L_d} g(L_d, iD_1) \left\{ \sum_{j=0}^{K_v - n_V^D} e^{-\lambda_v i D_1} \frac{(\lambda_v i D_1)^j}{j!} (n_V^D + \frac{j}{2}) + \right. \right. \\
 & \left. \left. \sum_{j=K_v - n_V^D + 1}^{\infty} e^{-\lambda_v i D_1} \frac{(\lambda_v i D_1)^j}{j!} \left( K_v - \frac{(K_v - n_V^D + 1)(K_v - n_V^D)}{2(j+1)} \right) \right\} \right] \quad (IV.120)
 \end{aligned}$$

#### IV.3.5.2.2 Número médio de pares de células de voz

Seja  $N_V$  o número médio de pares de células de voz em um ciclo. Sejam  $N_V^V$  e  $N_V^D$  os números médios de pares de células de voz que armazena o multiplexador durante a fase de voz e de dados, respectivamente. Então, com um raciocínio análogo ao efetuado para calcular o número médio de células de dados, conclui-se que:

$$N_V = \frac{T_v}{T} N_V^V + \frac{T_d}{T} N_V^D \quad (IV.121)$$

onde  $\frac{T_v}{T}$  e  $\frac{T_d}{T}$  correspondem às probabilidades que o multiplexador esteja na fase de voz ou de dados, respectivamente.

Na seção IV.3.4.1 foram obtidas as expressões matemáticas que permitem avaliar  $T$ ,  $T_v$  e  $T_d$ . Em conseqüência, para avaliar o número médio de pares de células de voz, somente resta avaliar  $N_V^V$  e  $N_V^D$ , o que é feito a seguir.

#### IV.3.5.2.2.1 Avaliação de $N_V^V$

Como  $N_V^V$  depende do estado do multiplexador ao início da fase de voz e, usando um raciocínio similar ao realizado para obter  $N_D^D$ , conclui-se que:

$$N_V^V = \sum_{\forall \bar{n}^V} \bar{\pi}_V[\bar{n}^V] \left\{ \frac{1}{L_V} \sum_{i=1}^{L_V} \sum_{e=1}^{K_V} e \pi_{W_e}(i-1) \right\} \quad (\text{IV.122})$$

onde  $\bar{\pi}_V[\bar{n}^V]$  corresponde à probabilidade, em estado estacionário, de que a fase de voz comece no estado  $\bar{n}^V$  e;  $\pi_{W_e}(i)$  é a probabilidade que a cadeia de Markov  $W$  fique no estado  $e$  imediatamente depois da  $i$ -ésima transição.

#### IV.3.5.2.2.2. Avaliação de $N_V^D$

Dado a que  $N_V^D$  depende do estado do multiplexador no início da fase de dados, observa-se que:

$$N_V^D = \sum_{\forall \bar{n}^D} \bar{\pi}_D[\bar{n}^D] N_V^D(\bar{n}^D) \quad (\text{IV.123})$$

onde  $\bar{\pi}_D[\bar{n}^D]$  corresponde à probabilidade em estado estacionário de que a fase de dados comece no estado  $\bar{n}^D$ .  $N_V^D(\bar{n}^D)$  corresponde ao número médio de pares de células de voz durante a fase de dados, uma vez que a fase de dados inicia-se no estado  $\bar{n}^D$ .

Usando um raciocínio semelhante ao feito para obter uma expressão para  $N_D^V(\bar{n}^V)$ , conclui-se que:

$$N_V^D(\bar{n}^D) = \sum_{i=\min[\bar{n}^D, L_d]}^{L_d} \Pr\{t_d = iD_1\} \left[ \sum_{j=0}^{K_V - \bar{n}_V^D - 1} \binom{\bar{n}_V^D + j}{2} e^{-\lambda_V i D_1} \frac{(\lambda_V i D_1)^j}{j!} + \sum_{j=K_V - \bar{n}_V^D}^{\infty} \left( K_V - \frac{(K_V - \bar{n}_V^D)(K_V - \bar{n}_V^D + 1)}{2(j+1)} \right) e^{-\lambda_V i D_1} \frac{(\lambda_V i D_1)^j}{j!} \right] \quad (\text{IV.124})$$

onde  $\Pr\{t_d = iD_1\}$  corresponde à probabilidade de que a fase de dados dure  $iD_1$  unidades de tempo.

Atribuindo recompensas as transições da matriz  $M$ , segundo a seguinte função:

$$\eta(i, j) = \begin{cases} 0; & i = 0 \\ D_1 & i > 0; i - 1 \leq j \leq K_v \end{cases}$$

Posteriormente, usando a metodologia da seção IV.1 obtém-se que:

$$\Pr\{t_d = iD_1 \mid \bar{n}^D\} = g(L_d, iD_1) \quad (\text{IV.125})$$

onde  $g(L_d, iD_1)$  corresponde à probabilidade que a recompensa total acumulada pela cadeia  $M$  em  $L_d$  transições seja igual a  $iD_1$ .

Finalmente, substituindo as equações (V.125), (V.124), e (V.122) em (V.121), é obtida a seguinte expressão final para  $N_V$ :

$$\begin{aligned} N_V = & \frac{T_v}{T} \sum_{\forall \bar{n}^V} \bar{\pi}_V[\bar{n}^V] \left[ \frac{\sum_{i=1}^{L_v} \sum_{e=1}^{K_v} \pi_{W_e}(i-1) e}{L_v} + \right. \\ & \frac{T_d}{T} \sum_{\forall \bar{n}^D} \bar{\pi}_D[\bar{n}^D] \left[ \sum_{i=\min\{n_D^D, L_d\}}^{L_d} g(L_d, iD_1) \left\{ \sum_{j=0}^{K_v - n_V^D} e^{-\lambda_v i D_1} \frac{(\lambda_v i D_1)^j}{j!} (n_V^D + \frac{j}{2}) + \right. \right. \\ & \left. \left. \sum_{j=K_v - n_V^D + 1}^{\infty} e^{-\lambda_v i D_1} \frac{(\lambda_v i D_1)^j}{j!} \left( K_v - \frac{(K_v - n_V^D + 1)(K_v - n_V^D)}{2(j+1)} \right) \right\} \right] \end{aligned} \quad (\text{IV.126})$$

### IV.3.5.3 Retardo Médio

#### IV.3.5.3.1 Retardo médio experimentado por um par de células de voz

Seja  $R_V$  o retardo médio experimentado por um par de células de voz no multiplexador. Pela Lei de Little obtém-se que:

$$R_V = N_V \bar{\mu}_V \quad (\text{IV.127})$$

onde  $N_V$  corresponde ao número médio de pares de células de voz no multiplexador (expressão obtida na seção IV.3.4.2.1) e  $\bar{\mu}_V$  é a taxa média de atenção de pares de células de voz.

Note-se que durante as fases de inatividade e de dados, as células de voz não são atendidas, ou seja durante essas fases a taxa de atenção de células de voz é igual a zero.

Logo, usando o teorema de probabilidades totais,  $\bar{\mu}_V$  é dado por:

$$\bar{\mu}_V = \bar{\mu}_V^V \frac{T_v}{T} \quad (\text{IV.128})$$

onde  $\bar{\mu}_V^V$  corresponde à taxa média de atenção efetiva de pares de células de voz durante a fase de voz.

Devido a que  $\bar{\mu}_V^V$  depende do estado do multiplexador no começo da fase de voz, pelo teorema de probabilidades totais, conclui-se que:

$$\bar{\mu}_V^V = \sum_{\forall \bar{n}^V} \bar{\pi}_V[\bar{n}^V] \bar{\mu}_V^V(\bar{n}^V) \quad (\text{IV.128})$$

Por outro lado, como a taxa de atenção de pares de células de voz não é constante durante a fase de voz, para calcular  $\bar{\mu}_V^V(\bar{n}^V)$  é somada a taxa média de atenção de cada um dos  $L_v$  segmentos de uma fase de voz, e esta soma é dividida por  $L_v$ . Portanto:

$$\bar{\mu}_V^V(\bar{n}^V) = \frac{1}{L_v} \left( \frac{1}{D_1} \sum_{i=1}^{L_v} \sum_{\forall e>u} \pi_e(i-1) + \frac{1}{D_2} \sum_{i=1}^{L_v} \sum_{\forall 0<e\leq u} \pi_e(i-1) \right) \quad (\text{IV.129})$$

Substituindo as equações (IV.129) e (IV.128) em (IV.127) é obtida a expressão para a taxa média de atenção de pares de células de voz:

$$\bar{\mu}_V = \frac{T_v}{T} \sum_{\forall \bar{n}^V} \bar{\pi}_V[\bar{n}^V] \left[ \frac{1}{D_1} \frac{\sum_{i=1}^{L_v} \sum_{\forall e>u} \pi_e(i-1)}{L_v} + \frac{1}{D_2} \frac{\sum_{i=1}^{L_v} \sum_{\forall 0<e\leq u} \pi_e(i-1)}{L_v} \right] \quad (\text{IV.130})$$

Multiplicando a equação (IV.130) por  $N_v$ , finalmente, conclui-se que o retardo médio experimentado por um par de células no multiplexador, é dado por:

$$R_V = N_v \frac{T_v}{T} \sum_{\forall \bar{n}^V} \bar{\pi}_V[\bar{n}^V] \left[ \frac{1}{D_1} \frac{\sum_{i=1}^{L_v} \sum_{\forall e>u} \pi_e(i-1)}{L_v} + \frac{1}{D_2} \frac{\sum_{i=1}^{L_v} \sum_{\forall e\leq u} \pi_e(i-1)}{L_v} \right] \quad (\text{IV.131})$$

#### IV.3.5.3.2 Retardo médio experimentado por uma célula de dados

Seja  $R_D$  o retardo médio experimentado por uma célula de dados no multiplexador. Então, pela Lei de Little, é obtido que:

$$R_D = N_D \bar{\mu}_D \quad (\text{IV.132})$$

onde  $N_D$  corresponde ao número médio de células de dados no multiplexador (que foi calculado na seção IV.3.4.2.2) e,  $\bar{\mu}_D$  é a taxa média de atenção efetiva de células de dados.

Utilizando um raciocínio similar ao realizado para obter  $\bar{\mu}_V$ , conclui-se que:

$$\bar{\mu}_D = \bar{\mu}_D^D \frac{T_d}{T} \quad (\text{IV.133})$$

onde  $\bar{\mu}_D^D$  corresponde a taxa média de serviço efetivo de células de dados durante a fase de dados.  $T$  e  $T_d$  representam a duração média de um ciclo e da fase de dados, respectivamente, os quais foram calculados na seção IV.3.4.1. Devido a que durante a fase de dados são atendidas células de dados a uma taxa constante igual a  $1/D_1$ , ou seja  $\bar{\mu}_D^D = 1/D_1$ , a equação (IV.133) pode ser rescrita da seguinte forma:

$$\bar{\mu}_D = \frac{T_d}{TD_1} \quad (\text{IV.134})$$

Substituindo a equação (IV.134) em (IV.132), é obtida a expressão para o retardo médio de células de dados:

$$R_D = \frac{N_D T_d}{TD_1} \quad (\text{IV.135})$$

#### IV.3.5.4. Probabilidade de Perda

##### IV.3.5.4.1 Probabilidade de Perda de Pares de Células de Voz

Seja  $P_{P_V}$  a probabilidade de perda de células de voz. Aplicando a Lei de Little ao servidor obtém-se que:

$$\lambda_v (1 - P_{P_V}) = \rho_V C \quad (\text{IV.136})$$

onde  $C$  é a capacidade do canal, que é igual a 1 par de células cada  $D_2$  unidades de tempo; e

$\rho_V$  é a ocupação média do servidor por parte do tráfego de voz, a qual é dada por  $\rho_V = \frac{T_V}{T}$

Então, substituindo os valores de  $C$  e  $\rho_v$  na equação (IV.136) e, isolando a probabilidade de perda de células de voz, chega-se a:

$$P_{P_v} = 1 - \frac{T_v}{\lambda_v D_2 T} \quad (\text{IV.137})$$

#### **IV.3.5.4.2 Probabilidade de Perda de Células de Dados**

Utilizando linha de pensamento análoga àquela usada para o caso da probabilidade de perda de células de voz, pode ser calculada a probabilidade de perda de células de dados,  $P_{P_d}$ . Esta probabilidade é dada pela seguinte expressão:

$$P_{P_d} = 1 - \frac{T_d}{\lambda_d D_1 T} \quad (\text{IV.138})$$

#### **IV.3.5.5 Vazão (throughput)**

##### **IV.3.5.5.1 Vazão do Tráfego de Voz**

Seja  $Th_v$  a vazão do tráfego de voz. Pode ser observado que, quando o multiplexador atende tráfego de voz, a vazão deste tipo de tráfego é equivalente à capacidade do canal. Ao contrário, quando o multiplexador se encontra em uma fase diferente à fase de voz, a vazão para o tráfego de voz é igual a zero. Assim, aplicando a definição de valor médio é obtido que:

$$Th_v = C \frac{T_v}{T} + 0(1 - \frac{T_v}{T}) \quad (\text{IV.139})$$

onde  $C$  é a capacidade do canal (igual a 1 par de células cada  $D_2$  unidades de tempo); e o fator  $\frac{T_v}{T}$  corresponde a probabilidade de que o multiplexador se encontre na fase de voz.

Substituindo o valor de  $C$  na equação (IV.139), é obtida a vazão do tráfego de voz, medida em pares de células de voz por unidade de tempo:

$$Th_v = \frac{T_v}{D_2 T} \quad (\text{IV.140})$$

#### IV.3.5.5.2 Vazão do Tráfego de Dados

Com um raciocínio semelhante ao realizado para o caso da vazão do tráfego de voz, é obtida a expressão para a vazão do tráfego de dados,  $Th_D$ , a qual é medida em células de dados por unidade de tempo. Esta vazão é dada pela expressão:

$$Th_D = \frac{T_d}{D_1 T} \quad (\text{IV.141})$$

#### IV.3.6 Exemplos Numéricos

Na Figura IV.35 é apresentado o gráfico da vazão de dados normalizada em função da carga de dados normalizada no canal. Ademais, o gráfico mostra diferentes curvas que representam a carga de voz que tem o canal. No gráfico a carga de dados no canal é representada por  $\rho_d$  e a carga de voz por  $\rho_v$ .

Para este exemplo foi utilizado um buffer de voz que permite armazenar dois pares de células de voz e um buffer de dados com capacidade para quatro células de dados. Esses valores foram escolhidos a propósito para mostrar o efeito que tem um buffer finito e de pouca capacidade sobre a vazão. Também, devido a pouca capacidade do buffer de voz não é necessário fazer controle de congestionamento. Por último, a política de escalonamento usada é a Política Proporcional a Carga, apresentada na seção III.4.1

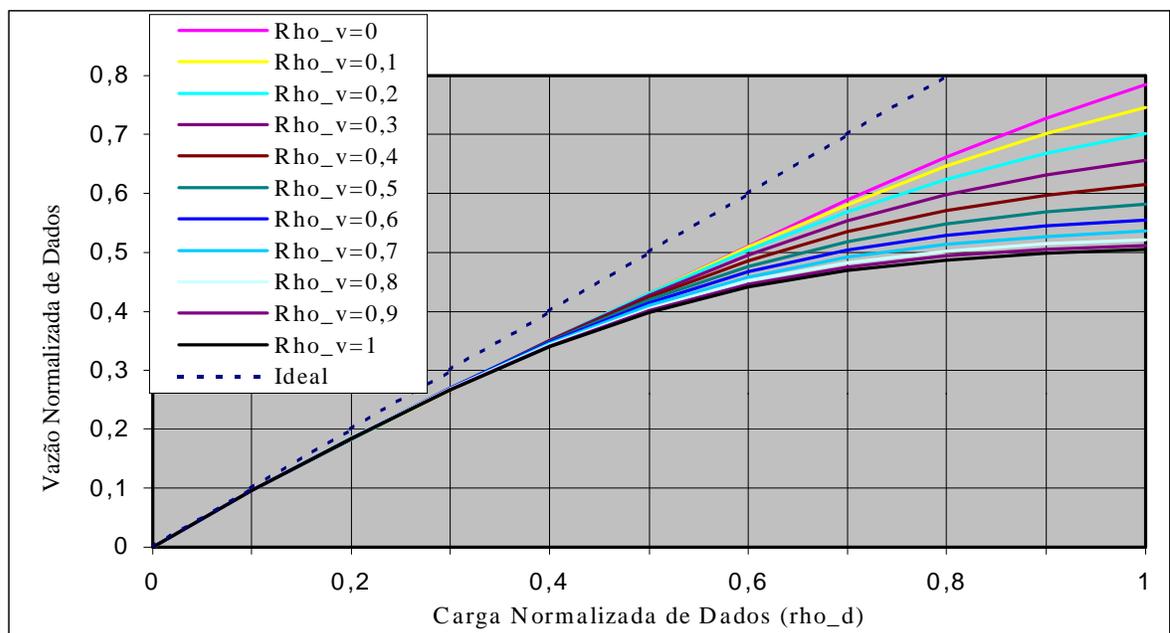


Figura IV.35 Vazão para o tráfego de dados em função da carga de dados

No gráfico da Figura IV.35, a vazão ideal (ou seja, a vazão de um buffer infinito e uma carga de voz no canal de 0) é mostrada com linha segmentada.

Também pode ser observado que para o caso  $\rho_v = 0$  a vazão normalizada de dados apresenta uma degradação em relação ao caso ideal, ainda para carga muito baixa do canal de voz. Isto é devido ao fato de que o buffer é finito e de pouca capacidade (2 pares de células). Ainda nesse gráfico, pode ser notado que para uma mesma carga de dados ( $\rho_d$ ), na medida em que a carga de voz ( $\rho_v$ ) aumenta, a vazão diminui. Esta situação é bastante notória quando a soma das cargas de voz mais a carga de dados aproxima-se ao 100% da capacidade do canal (ou equivalentemente, a carga normalizada é 1). Por exemplo, para uma carga normalizada de dados igual a 0.6 e uma carga normalizada de voz igual a 0.20, a vazão normalizada de células de dados é de 0.503; mas quando a carga normalizada de dados também é de 0.60 e a carga normalizada de voz aumenta a 0.40, a vazão normalizada de células de dados diminui a 0.48.



Figura IV.36 Probabilidade de perda de células de voz versus a carga de voz normalizada no canal.

O gráfico de Figura IV.36 mostra a probabilidade de perda de células de voz em função da carga de voz normalizada no canal ( $\rho_v$ ). No gráfico são mostradas as curvas para diferentes limiares de controle de congestionamento. Para este exemplo, supõe-se que somente são transmitidas células de voz, pelo qual a fase de dados não existe. Isto foi feito com o intuito de encontrar um limite inferior das perdas de pares de células de voz.

Para este exemplo foi utilizado um buffer de voz que permite armazenar 10 pares de células de voz e um buffer de dados com capacidade para 20 células de dados. A política de escalonamento usada é a Política Proporcional a Carga.

Em primeiro lugar note que para cargas normalizadas do canal menores ou iguais que 0.6, não existe diferença na probabilidade de perda de voz para os distintos limiares de congestionamento graficados. Para cargas maiores a 0.6, pode ser visto no gráfico que a medida que aumenta o limiar de congestionamento a probabilidade de perda diminui. Isto é devido a que, quando o limiar é menor a situação de congestionamento é detectada antes que para um limiar mais alto, e nesse instante começam a ser descartadas as células de menor prioridade, acelerando dessa maneira ao serviço das outras células que estão no buffer. Por exemplo, na Tabela IV.1 são detalhadas as probabilidades de perda de células de voz para diferentes limiares de congestionamento, para uma carga normalizada do canal de 0.9

<b>Probabilidade de perda de células de voz</b>	<b>Limiar de Congestionamento (pares de células de voz)</b>
0.1	10
0.12	7
0.14	5

Tabela IV.1. Probabilidade de Perda de Células de Voz

Na Figura IV.37 é apresentado o gráfico da probabilidade de perda de células de dados em função da carga no canal. O gráfico mostra curvas que representam a probabilidade de perda de dados para duas políticas de escalonamento: a Política WRR e a Política de Prioridade Simples (apresentadas na seção III.3.4).

Para este exemplo foi utilizado um buffer de voz que permite armazenar 10 pares de células de voz e um buffer de dados com capacidade para 20 células de dados. Não é considerado para este exemplo fazer controle de congestionamento.

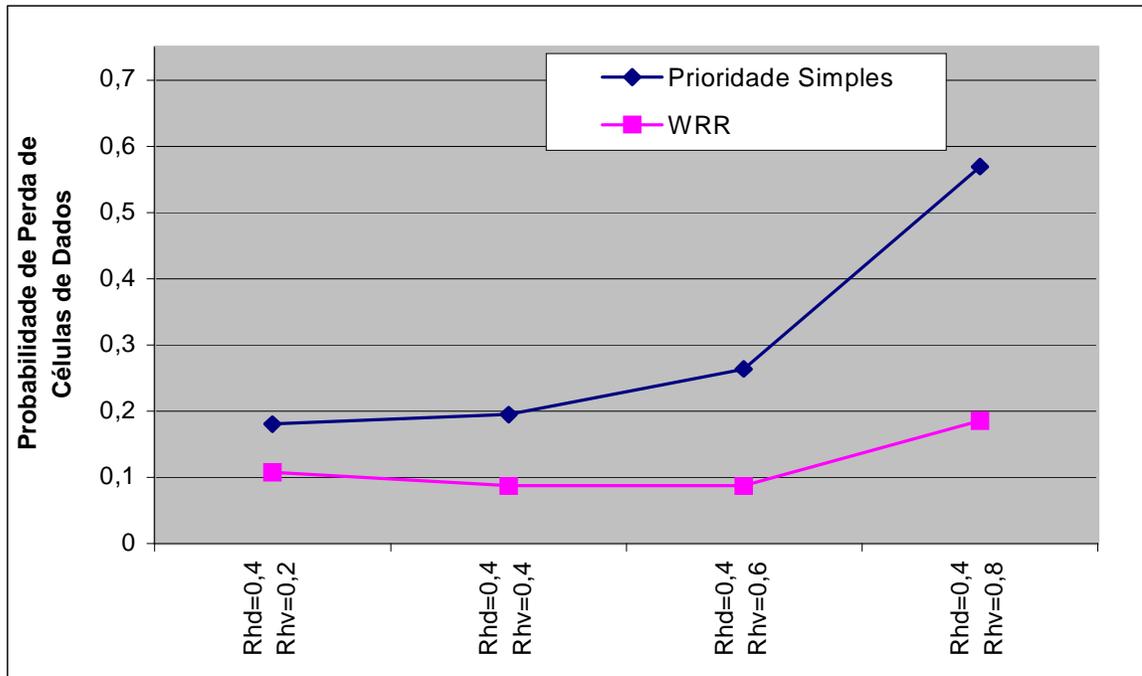


Figura IV.37 Probabilidade de perda de células de dados versus a carga normalizada do canal

No gráfico a carga de dados no canal ( $\rho_d$ ) é mantida constante e igual a 0.4. e a carga de voz ( $\rho_v$ ) é variável.

Pode ser notado que para qualquer carga normalizada do canal, a política de escalonamento WRR apresenta menos perdas de células de dados que a política de escalonamento de Prioridade Simples. Esta situação é mantida para qualquer carga do canal, mas a diferença é mais notória quando o canal está mais carregado. Isto deve-se a que o tráfego de voz tem maior prioridade que o tráfego de dados, portanto na medida que aumenta a carga de voz o tráfego de dados pode acessar o canal cada vez menos. Isto implica que o buffer chega a saturar com a conseguinte perda das células de dados que chegam ao buffer. Esta situação é similar para ambas políticas, sendo mais dramática no caso de Prioridade Simples, pois em cada ciclo pode transmitir somente uma célula. Por exemplo, para uma carga normalizada de 0.6 (onde a carga do tráfego de voz é de 0.2 e a carga do tráfego de dados é de 0.4) e uma política WRR a probabilidade de perda de dados é de 0.108 e, para Prioridade Simples a

probabilidade de perda é de 0.18; porém, para uma para uma carga normalizada de 1.0 (onde o canal esta carregado pelo tráfego de voz por 0.6 e a carga do tráfego de dados é de 0.4) e uma política WRR a probabilidade de perda de dados é de 0.09 e para Prioridade Simples a probabilidade de perda é de 0.26. Esta situação é explicada pelo fato de que o algoritmo de Prioridade Simples a cada ciclo escalona para transmissão somente uma célula de dados enquanto WRR atribui um certo tempo de transmissão.

### **IV.3.7 Resumo da seção IV.3**

Nesta seção IV.3 é apresentada a avaliação de desempenho de LDS, particularizada para o caso de um multiplexador de voz e dados. Esta avaliação foi levada a cabo utilizando a teoria de Cadeias de Markov com Recompensas, e o resultado final é um conjunto de equações que pode ser aplicada a qualquer algoritmo de escalonamento representado através de uma Tabela LDS.

Foram avaliadas diversas medidas de desempenho tais como: tamanho médio de um ciclo, número médio de células de voz e de células de dados no sistema, retardo médio, vazão média.

## Capítulo V

# Algoritmo Para a Recuperação de Erros de uma Transmissão Multiponto Confiável

### V.1 Introdução

Uma das áreas de maior interesse, na evolução atual das telecomunicações, é o desenvolvimento de aplicações multiponto, devido ao grande impacto que têm na sociedade atual. As teleconferências, os quadros compartilhados, a distribuição de filmes, os jornais interativos, são alguns dos exemplos mais comuns deste tipo de aplicações [DIO97,MIL98].

Do ponto de vista social, é indubitável que as aplicações multiponto são extremamente úteis, devido ao fato de que a relação entre as pessoas realiza-se principalmente em grupos e não pessoa a pessoa. Apesar disto e, devido a limitações tecnológicas, inicialmente os sistemas de telecomunicações só permitiam oferecer serviços interativos ponto a ponto. Felizmente a tecnologia tem evoluído ao ponto de permitir o desenvolvimento de aplicações multiponto, que, como já foi dito, são mais naturais para o ser humano. Os principais desenvolvimentos tecnológicos que tornaram possível oferecer esses serviços multiponto são: a implementação das redes de alta velocidade, o incremento da velocidade de processamento dos computadores e o grande êxito que tem tido o Internet Multicasting usando o Mbone [ERI94].

Muitas das aplicações multiponto precisam que o serviço multiponto seja confiável, isto é, uma qualidade de serviço com perdas nula. Evidentemente que a probabilidade que um pacote seja perdido (devido à ocorrência de algum erro na rede) pode ser diminuída usando diversos mecanismos. Entre eles podem ser mencionados:

- a melhora da qualidade dos elementos da rede, por exemplo, o uso de fibra ótica diminui a probabilidade de erro nos canais;
- o uso de técnicas de controle de congestionamento, já que estas diminuem a probabilidade de perda nos buffers;

- a utilização de códigos de correção, como o “*Forward Error Correction*” (FEC) [BLA83,RIZ97], que permitem que alguns dos pacotes que se perdem possam ser recuperados sem ter que retransmití-los, o que aumenta em forma considerável a vazão (throughput) na transmissão confiável a grandes grupos [HUI96, NON96, NON97].

Entretanto, mesmo com todos esses cuidados, devido ao fato que os grupos de usuários de uma aplicação multiponto podem conter milhares e até milhões de receptores, a probabilidade de que ocorra um erro em algum lugar da rede não é desprezível.

Para recuperar os pacotes que possam ser perdidos, utilizam-se mecanismos de recuperação de erros, os quais são constituídos, basicamente, por uma fase de detecção e outra de recuperação por retransmissão.

Devido à grande quantidade de pesquisa que tem sido feita nos últimos anos no tema da transmissão multiponto confiável, não é fácil ter uma visão clara do problema e as alternativas de solução possíveis. Por este motivo, é útil identificar os graus de liberdade do problema de detecção e recuperação de erros, e as distintas alternativas de solução de cada um de seus componentes.

## **V.2 Taxonomia Proposta**

Nesta seção é apresentada uma taxonomia sobre os protocolos para transmissão multiponto confiável. Esta taxonomia é realizada levando em consideração as etapas que são necessárias para solução do problema da detecção e recuperação de erros. Nas subseções seguintes são explicados em detalhe estas etapas de classificação.

### **V.2.1 Quem detecta a perda de um pacote?**

Nas comunicações ponto a ponto podem ser distinguidas basicamente as seguintes formas de detectar e corrigir a perda de um pacote transmitido:

- receptor envia à fonte reconhecimentos positivos ou ACKs.
- receptor envia à fonte reconhecimentos negativos ou NACKs.
- receptor envia à fonte uma combinação de ACKs e NACKs

No esquema baseado no uso de ACKs, o receptor envia à fonte uma mensagem de aceitação (ou ACK) por cada pacote recebido corretamente. Por sua vez, a fonte contabiliza todos os pacotes transmitidos e controla quais não receberam confirmação de sua correta recepção. Se

após um certo intervalo de tempo limite o transmissor não recebe o ACK do pacote transmitido, supõe-se que o pacote se perdeu, devendo o mesmo ser retransmitido. Assim, nesta forma de recuperação de erros, é o transmissor quem tem a responsabilidade de manter a confiabilidade da transmissão que, por este motivo, é classificada como iniciada no transmissor (sender-initiated) [PIN94, TOW97]. Como exemplo, TCP é um protocolo transmissão ponto a ponto confiável (talvez o mais popular) deste tipo.

A principal característica dos protocolos iniciados no transmissor (sender-initiated) é que a fonte (ou transmissor), necessita contabilizar todos os pacotes transmitidos e processar todos os ACKs que recebe. Por este motivo, não é eficiente estender o uso de ACKs para o caso de comunicações multiponto [PIN94,TOW97], já que num ambiente multiponto pode existir um grande número de receptores (por exemplo  $10^6$ ), o que significa que a fonte deveria receber os ACKs enviados por todos os receptores em relação a cada pacote transmitido. Isto produziria uma enorme implosão de ACKs no transmissor. Além do mais, o transmissor deveria manter o estado de avanço (isto é, qual é o último pacote recebido corretamente) de cada um dos receptores e, processar os ACKs que lhe chegam para determinar se todos os receptores receberam corretamente um determinado pacote; tudo isso é muito ineficiente. Um exemplo de protocolo de transmissão multiponto confiável que usa ACKs é o protocolo de Transporte Multiponto Confiável (Reliable Multicast Transport Protocol (RMTP)) [LIN96,PAU97].

Os mecanismos que responsabilizam a confiabilidade da comunicação aos receptores, fazem que o transmissor retransmita um pacote somente quando recebe um NACK (relacionado com esse pacote) de pelo menos um receptor. Cada receptor é responsável por detectar a perda de informação e pedir a retransmissão do pacote correspondente. As perdas geralmente são detectadas porque se produzem saltos na seqüência da numeração dos pacotes recebidos. Para dar segurança à retransmissão de um pacote, após ter enviado o NACK, o receptor ativa um relógio que é próprio a essa solicitação de retransmissão. Se o relógio indicar o máximo de um certo tempo limite sem que a retransmissão do pacote seja recebida, o receptor enviará novamente o NACK solicitando o pacote. O esquema recém descrito para obter confiabilidade é classificado como iniciado no receptor (receiver-initiated) [PIN94, TOW97], já que é o receptor quem inicia o ciclo de recuperação da informação perdida.

As principais características dos protocolos iniciados no receptor são:

A fonte não tem que processar um ACK por cada pacote e por cada receptor. Somente processa os NACKs que lhe são enviados, quando um pacote não chega a todos os receptores. A fonte não necessita conhecer o estado dos receptores. Isto implica que a implementação desta técnica requer menos memória que aquela do iniciado no transmissor (sender-initiated). Em comunicações multiponto onde o meio de transmissão tem uma baixa taxa de erro, é mais eficiente que a técnica de iniciado no transmissor, já que a perda de pacotes é um fenômeno raro (com relação à não-perda do mesmo pacote). Por este motivo, a quantidade de NACKs que o transmissor deve processar é muito menor que os ACKs do caso iniciado no transmissor. Isto implica numa diminuição da implosão no transmissor, como também uma economia no uso dos recursos da rede, tais como, uso de processadores, uso dos canais de comunicação e buffers.

Devido às boas características dos protocolos iniciados no receptor a maioria dos protocolos de transmissão multiponto modernos tem adotado este esquema para a recuperação de pacotes perdidos [PIN94, TOW97, YAM97, KAS98]. Também, para conseguir um bom desempenho em relação à vazão, estes protocolos usam o mecanismo de repetição seletiva (*selective repeat*) para a retransmissão dos pacotes perdidos [KAS98], isto é, o transmissor somente retransmite os pacotes dos quais recebe um NACK.

Entre os protocolos de transporte multiponto confiável que usam somente NACKs para garantir confiabilidade na transmissão, podem ser mencionados: *Multicast Transport Protocol* (MTP) [AMS92, BOR94] *Uniform Reliable Group Communication Protocol* (URGC) [AIE93], *Transport Protocol for Reliable Multicast* (TRM) [SAB96], *Multicast Dissemination Protocol* (MDP) [MAC96], *Adaptive File Distribution Protocol* (AFDP) [COO96], *Scalable Reliable Multicast* (SRM) [FLO97], *Structured-Oriented Resilient Multicast* (STORM) [XU97], *Pragmatic General Multicast* (PGM) protocol [SPE99].

Existem outros protocolos que obtêm confiabilidade ao usar uma combinação de ACKs e NACKs. Para isto, usam uma estrutura hierárquica, na qual existem nós intermediários (ou representantes). Um membro que detecta a perda de um pacote, envia o NACK correspondente a seu representante. Quando o representante recebe um NACK de um pacote que tem armazenado corretamente, reenvia o pacote solicitado. Quando o receptor recebe corretamente um pacote, quer seja a primeira vez ou após ter solicitado sua retransmissão, envia um ACK a seu representante. Por sua vez, quando um representante recebe todos os ACKs de seus representados, envia um único ACK (relacionado com o pacote) à fonte. No

caso em que a perda afete ao representante, este envia um NACK à fonte pedindo a retransmissão do pacote perdido. Entre estes protocolos, podem ser mencionados: *Reliable Broadcast Protocol* (RBP) [CHA84], *Xpress Transfer Protocol* (XTP) [STR92,XTP], *Reliable Multicast Protocol* (RMP) [WHE95], *Log-Based Receiver Multicast Protocol* (LBRM) [HOL95], *Tree-Based Multicast Transport Protocol* (TMTP) [YAV95], *Multicast File Transfer Protocol* (MFTP) [MIL97], etc.

### **V.2.2 Como se detecta a perda de um pacote?**

Na maioria dos casos, a perda de pacotes é detectada pelo receptor porque se produz um salto na seqüência da numeração dos pacotes que lhe chegam. Por outro lado, quando trata-se da perda de um pacote retransmitido, ou quando não se retransmite o pacote, a detecção também é realizada pelo receptor, mas desta vez usando um mecanismo de *timeout*.

### **V.2.3 Quem envia o NACK?**

Dado que os protocolos que somente usam NACKs têm o melhor desempenho nas comunicações multiponto [PIN94, TOW97], o restante desta taxonomia referir-se-á exclusivamente a este tipo de protocolos.

A perda de cada pacote particiona naturalmente a rede em duas classes: a dos membros que são afetados pela perda e a dos membros que não foram afetados por esta. Todos os membros afetados pela perda são candidatos a enviar NACKs, enquanto todos os membros não afetados pela perda são candidatos a retransmitir o pacote perdido (nos casos que mantenham cópia dele).

Sem dúvida, se cada um dos receptores afetados pela perda de um pacote enviasse um NACK, na fonte se produziria a chegada sincronizada de (possivelmente) muitos NACKs. Este fenômeno se conhece com o nome de “implosão de NACKs”. Esse é um fenômeno não desejado porque, além de fazer mau uso dos recursos da rede (tais como processadores, canais de comunicação e buffers), gera congestionamento, o que obviamente causa uma degradação nas medidas de desempenho globais (QoS) da rede, como o retardo e a vazão [PIN94,TOW97]. Por este motivo, os mecanismos de recuperação de erros tentam diminuir (ou evitar) tanto a implosão, como diminuir a latência (ou seja, o tempo para recuperar o pacote perdido). Esses objetivos, no entanto, são conflitantes. Por isso, os algoritmos que têm

sido propostos na literatura tentam obter um compromisso entre o tamanho da implosão e o tempo de latência.

O caso em que cada um dos membros afetados pela perda envia seu NACK em forma multiponto (a todos os membros do grupo), é chamado de “recuperação global”. A técnica usada para diminuir a implosão consta de duas partes. Na primeira, denominada segmentação (*slotting*), cada um dos membros candidatos a enviar o NACK realiza um experimento aleatório e, segundo este resultado, determina o instante (no futuro) em que deve transmitir seu NACK. Quando o NACK é efetivamente transmitido, o envio se faz de forma que, além de chegar aos membros que têm a informação requerida, chegue também aos membros que a perderam. A segunda parte, denominada amortecimento (*damping*), consiste na inibição da transmissão de NACKs pelo recebimento do(s) NACK(s) do(s) membro(s) que primeiro o(s) transmitiu(ram), diminuindo dessa forma a implosão.

Sobre o tipo de variável aleatória que é mais conveniente para determinar o instante em que se deve enviar um NACK, têm surgido várias propostas na literatura. Por exemplo, em SRM [FLO97] propõe-se usar uma distribuição uniforme. Por sua vez, em [NON98A] propõe-se usar uma variável aleatória com distribuição exponencial truncada. Em [GRO97] apresenta-se um algoritmo denominado DTRM (*Deterministic Timeouts for Reliable Multicast*) que usa tempos limites determinísticos para identificar o único membro que deve enviar o NACK.

Outra técnica que pode ser usada para diminuir a implosão, consiste em reduzir o tamanho do grupo dos potenciais candidatos a enviar NACKs, de forma que estes sejam somente um subconjunto próprio (com a menor quantidade possível de elementos) do conjunto de membros afetados pela perda. A este subconjunto próprio denomina-se “grupo local”. Os métodos usados para gerar grupos locais de potenciais transmissores de NACKs serão explicados mais adiante. Logicamente, o problema de implosão torna-se tanto menor quanto menor for o grupo que puder enviar o NACK. Como este grupo normalmente é composto por vários membros, persiste a possibilidade de acontecer a implosão de NACKs. Por este motivo, além de restringir-se a um grupo local a possibilidade de enviar NACKs, emprega-se também a técnica de *slotting and damping*.

## **V.2.4 Como diminuir o tamanho do grupo de potenciais candidatos a enviar NACKs?**

Para se conseguir diminuir o tamanho do grupo de potenciais candidatos a enviar NACKs, basicamente podem ser citadas três técnicas. A primeira, consiste em que o primeiro nó que detecta a perda, inibe seus descendentes de enviar o NACK [BAR00A]. A segunda forma, é a de estruturar os membros do grupo numa hierarquia de múltiplos níveis. Cada hierarquia tem um encarregado de processar e fundir os NACKs provenientes dos receptores de seu nível. Desse modo, através do encarregado, envia-se somente um NACK ao nível superior. Entre os protocolos que utilizam essa técnica podem ser mencionados: uso de “representantes” [LUC96], LBRM [HOL95] e TMTP [YAV95]. Finalmente, a terceira técnica, utiliza a passagem de fichas (*tokens*). Nesse caso, os membros do grupo são particionados em vários subgrupos (cada subgrupo é composto de nós de uma certa área da rede). Para cada subgrupo é escolhido um representante e os representantes são organizados numa estrutura de anel. Para cada pacote que é transmitido em forma multiponto, cria-se uma ficha própria deste pacote. A ficha é enviada a um dos membros do anel (ou seja, a um representante). Após receber um ACK de cada um dos membros de seu subgrupo, este representante passa a ficha ao seguinte membro do anel. Se após algum tempo limite, o representante que tem a ficha não receber todos os ACKs, que lhe devem chegar, este envia um NACK solicitando a retransmissão do pacote perdido. De acordo com o esquema, o pacote é recebido corretamente por todos os membros do grupo quando a ficha associada ao pacote retornar a um representante, quando então, a ficha é destruída. Entre os protocolos que usam este tipo de técnica pode-se mencionar o RBP [CHA84] e RMP [MON96].

## **V.2.5 Para onde enviar o NACK?**

Uma vez que um, ou vários, dos membros decidem enviar o NACK, surge a questão de para onde enviá-lo. Com relação a essa decisão existem três possibilidades:

- Enviar a todos os membros do grupo, incluindo a fonte (difusão dentro do grupo).
- Enviar a um subconjunto dos membros do grupo, em particular ao subconjunto dos membros que têm a informação requerida e que estão mais próximo do membro que envia o NACK.
- Enviar à fonte.

### **V.2.6 Quem retransmite o pacote perdido e para onde?**

Se o NACK for enviado a todos os membros do grupo, os membros que possuírem a informação perdida serão candidatos a retransmitir o pacote. Se todos transmitirem, será gerada uma situação análoga à causada pela implosão de NACKs, causando desta vez uma implosão de pacotes retransmitidos. Para evitar a implosão de retransmissões podem ser usadas as mesmas técnicas que as utilizadas na redução de implosão de NACKs. Isto significa que, para inibir retransmissões de outros possíveis candidatos, cada membro que possui o pacote determina, através de um experimento aleatório, em que instante no futuro irá transmiti-lo. No caso em que o membro chegue a este instante sem ter recebido previamente a retransmissão do pacote, ele envia o pacote por difusão ao grupo. Quando o membro recebe o pacote antes de que se cumpra o seu prazo fixado para enviá-lo, inibe sua transmissão, o que diminui a implosão de retransmissões.

No caso em que a retransmissão possa ser efetuada somente por um subconjunto de membros que recebem o NACK, (isto é, no caso em que seja usada alguma das técnicas para definir um grupo local de possíveis retransmissores do pacote), a situação é similar ao caso anterior, apenas menos dramática. Como naquele caso, também pode ocorrer implosão, que pode ser minimizada da mesma forma descrita no parágrafo anterior.

Quando a fonte é a única encarregada de efetuar a retransmissão, transmite-se o NACK em forma ponto-a-ponto à fonte. A primeira vez que a fonte recebe um NACK solicitando o pacote perdido, ela o retransmite, em forma multiponto, a todos os membros do grupo. Este método tem a vantagem de ser simples e não ter problemas de implosão de retransmissões. Sem dúvida, porém, tem a desvantagem de poder gerar uma latência muito grande, se a fonte estiver longe do lugar no qual se produziu a perda.

## **V.3 Classificação de alguns protocolos**

Na Tabela V.1 são classificados alguns protocolos para transmissão multiponto confiável de acordo com a taxonomia descrita na Seção V.2.2. Nas colunas, se indicam as etapas para resolver o problema de detecção e recuperação de erro de uma transmissão multiponto confiável.

<b>PROTOCOLO</b>	<b>Que tipo de reconhecimento se usa para obter confiabilidade?</b>	<b>Como se detecta a perda?</b>	<b>Quem envia o NACK?</b>	<b>Como diminuir o tamanho do grupo que envia NACK's?</b>	<b>Para onde enviar o NACK?</b>	<b>Quem retransmite pacote perdido?</b>
<b>AFDP</b>	NACK	Salto na seq. de numeração	Subscriber	backoff estatístico	Publicador	Publicador
<b>DTRM</b>	NACK	Salto na seq. de numeração	Receptor representante	timeouts deterministas	fonte	fonte
<b>LBRM</b>	ACK (entre fonte e servidores de logging ) NACK (entre servidores de logging e receptores)	Salto na seq. de numeração Ou pacotes não é um tempo máximo ocioso	Receptores	representativos	Servidor de Logging	Servidor de Logging
<b>PGM</b>	NACK	Salto na seq. de numeração	Receptores	Roteadores PGM Eliminam os NACK que sobem à fonte.	fonte	fonte ou Reparador Local Designado (DLR)
<b>MDP</b>	NACK	Salto na seq. de numeração	Receptores	NACK-backoff	fonte	fonte
<b>MFTP</b>	ACK –NACK	Salto na seq. de numeração	Receptores	backoff estatístico	fonte	fonte
<b>MTP</b>	NACK	Salto na seq. de numeração	Consumidores	token	produtores	produtores
<b>RAMP</b>	NACK	Salto na seq. de numeração	Receptores		transmissor	transmissor
<b>RBP</b>	ACK (entre fonte e token sites) – NACK (entre token sites e receptores)	Salto na seq. de numeração	Receptores	ficha	token site	token site ou fonte
<b>RMP</b>	ACK (entre fonte e token sites) NACK (entre token sites e receptores)	Salto na seq. de numeração	Receptores	fichas e backoff exponencial	Ao grupo	transmissor
<b>RMTP</b>	ACK	Salto na seq. de numeração	----	Representantes e grupo local	Receptor Designado (DR)	DR
<b>SRM</b>	NACK	Salto na seq. de numeração	Receptores	backoff uniformemente estatístico	multiponto	Qualquer host
<b>STORM</b>	NACK	Salto na seq. de numeração	Receptores	Representante	Nó padre	Nó pai imediato
<b>TMTTP</b>	ACK –NACK	Salto na seq. de numeração	Receptores	Representantes e slotting and damping	multiponto	pai imediato
<b>TRM</b>	NACK	Salto na seq. de numeração	Receptores	backoff estatístico	fonte	Qualquer host
<b>URGC</b>	NACK	Salto na seq. de numeração	Membros do grupo		nó mais atualizado	nó mais atualizado
<b>XTP</b>	ACK (quando solicitado pela fonte) -NACK	Salto na seq. de numeração	Receptores	slotting and damping	fonte	Fonte

Tabela V.1. Classificação de alguns protocolos de Transporte Multiponto Confiável.

### V.3.1 Métodos de Comparação dos protocolos

Os diferentes protocolos para transmissão multiponto confiável publicados na literatura têm sido analisados quase que exclusivamente por meio de simulação. Entre as medidas de desempenho utilizadas para avaliar esses protocolos, duas têm se sobressaído: 1) o tamanho da implosão e 2) a duração da latência de recuperação. Existem poucos artigos que analisem matematicamente os protocolos, estando os mesmos referenciados a seguir. Em [NON98B], emprega-se um método analítico para o cálculo do número médio de realimentações (NACKs ou ACKs) retornados à fonte e do valor médio da latência. Baseando-se nessas duas medidas de desempenho, o autor analisa o efeito de usar diferentes distribuições de probabilidade para escalonar o envio de NACKs. Entre as distribuições avaliadas estão a uniforme (usada em SRM), a exponencial truncada e a beta. Em [NON97B], o trabalho de [NON98B] foi aplicado para o caso particular de transmissão por difusão em satélites. Outras análises matemática aparecem em [RAM98], onde, matematicamente e por simulação, se analisa o comportamento do protocolo SRM [FLO97].

É importante notar que a avaliação de desempenho de protocolos de transmissão multiponto confiável tem sido realizada em relação à implosão e à latência, em um sistema de telecomunicações onde as métricas mais importantes são a vazão e o retardo, uma vez que essas medidas caracterizam a percepção que tanto o administrador como os usuários têm do sistema. Devido às dificuldades técnicas da análise, não existe um conhecimento profundo em relação à influência que têm a latência e a implosão sobre a vazão e o retardo. Alguns experimentos iniciais neste sentido foram publicados em [PIN94,TOW97,KAS98], onde obteve-se a vazão máxima ao comparar mecanismos de recuperação global de erro iniciados no transmissor, com mecanismos iniciados no receptor. Em [YAM97], se fez uma análise do retardo dos algoritmos genéricos que foram descritos em [TOW97] e em [DEC96] se fez uma análise de retardo de alguns algoritmos genéricos publicados na literatura até então. A importância da recuperação local é analisada em [NON98C] para algoritmos genéricos. Em [KAS97] se fez a análise de algoritmos genéricos iniciados no servidor versus iniciados no receptor, que utilizam recuperação local. Sem dúvida, para conseguir resolver o modelo matemático, todos esses trabalhos fazem suposições que não representam bem o que acontece na rede e, portanto, os modelos que analisam não são muito realistas. Torna-se assim ainda necessário avaliar analiticamente o impacto que têm os protocolos de transmissão multiponto confiável sobre a vazão e o retardo.

## V.4 Proposição de Algoritmo para a Transmissão Multiponto Confiável baseado em Árvore de Recuperação

### V.4.1 Motivação

O desempenho ideal seria obter um mecanismo de recuperação de pacotes perdidos em que o tamanho da implosão fosse igual a 1 e a latência fosse mínima. Essa dependência entre latência e implosão faz com que mecanismos para a recuperação de erros que têm sido propostos na literatura busquem diminuir simultaneamente a implosão e a latência, visando melhorar as medidas de desempenho globais da rede (vazão e retardo). Como estes objetivos são contraditórios, na prática os algoritmos propostos tentam obter um bom compromisso entre o tamanho da implosão e a latência (ver Figura V.1).

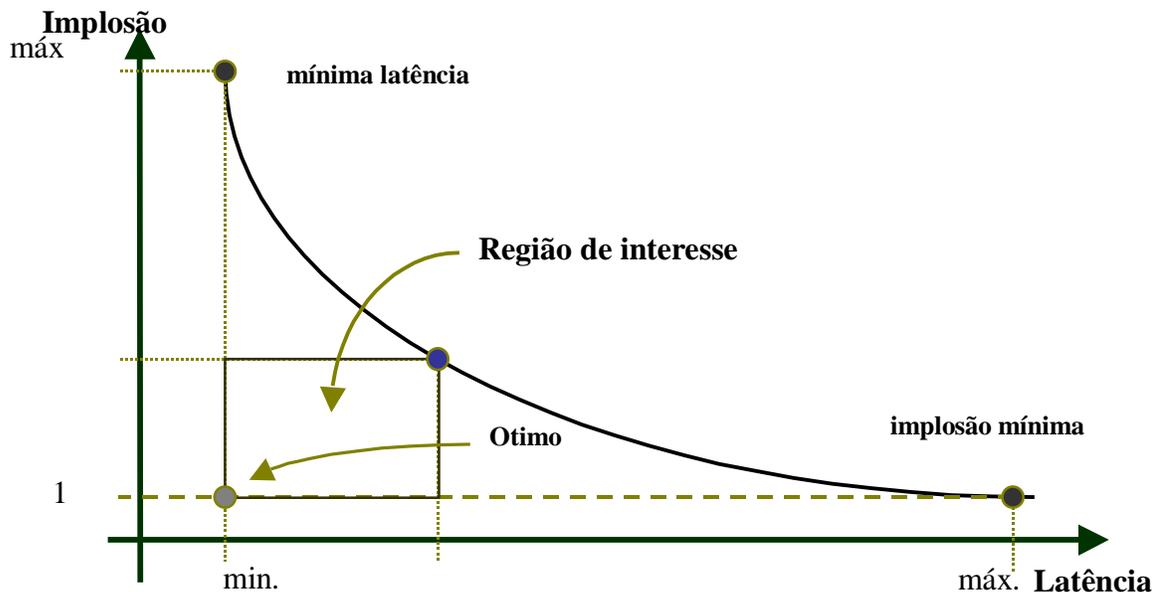


Figura V.1 Região de interesse para a proposição de algoritmos de recuperação de erros.

Esta seção apresenta um novo algoritmo de recuperação de erros, denominado ACMBAR (Algoritmo de Confiabilidade Multiponto baseado na Árvore de Recuperação), que tem alguns atributos que o fazem competitivo com relação aos algoritmos propostos existentes. Em particular, a recuperação de erros é feita em forma local. Isto é, os únicos membros do grupo que participam da recuperação do pacote perdido são os que se encontram na proximidade de onde se tem produzido o erro, obtendo desta forma um tamanho de implosão

próximo do ideal e uma baixa latência. Essa afirmação baseia-se em uma análise matemática do algoritmo proposto, onde foram avaliadas a implosão e a latência do algoritmo.

#### **V.4.2 Modelo**

A rede é representada por um grafo  $G=(V, L)$ , onde  $V=\{i, 1 \leq i \leq C\}$  corresponde ao conjunto de nós (ou comutadores) da rede,  $C$  é o número total de nós da rede e  $L=\{l_{ij}, 1 \leq i, j \leq C; i \neq j\}$  é o conjunto de arcos (que representam os canais de comunicação da rede) entre os nós de  $V$ .

Para representar a perda de pacotes, cada canal da rede tem associada uma probabilidade de perda de pacotes de dados e uma probabilidade de perda de NACKs. Essas probabilidades são diferentes, devido fundamentalmente ao fato do tamanho dos pacotes de NACKs ser menor que o dos pacotes de dados. Tanto as probabilidades de perda de pacotes de dados como de NACKs não variam no tempo. Isto significa que dois pacotes do mesmo tipo, que são transmitidos pelo mesmo canal em tempos diferentes, têm a mesma probabilidade de perder-se devido a um erro. Por outro lado, as probabilidades de perdas não são as mesmas no espaço, já que em geral canais diferentes têm associadas probabilidades de erro diferentes.

#### **V.4.3 Algoritmo Proposto**

A seguir são definidos alguns conceitos que ajudarão a explicar a operação do algoritmo.

Supõe-se que existe uma árvore de distribuição da informação [DIOT97] que é gerada pelo protocolo de roteamento multiponto subjacente na rede (normalmente chamada de árvore de multicast). A topologia da árvore de distribuição varia no tempo, já que é possível que em qualquer instante novos membros se integrem ao grupo ou que algum de seus membros o abandone.

Os pacotes que se originam em uma fonte são enviados em forma multiponto, através da árvore de distribuição, a todos seus descendentes. Quando um pacote chega a um nó da árvore, este nó envia uma cópia do pacote a cada um de seus filhos, dentro da árvore. Além disso, cada membro do grupo mantém uma cópia da informação que recebeu corretamente, a qual, em caso necessário, pode ser retransmitida a seus descendentes. Os nós que pertencem à árvore de distribuição, mas que não são membros do grupo, somente atuam como transmissores intermediários dos pacotes que recebem.

Os receptores são os responsáveis pela confiabilidade da comunicação. Isto significa que os receptores têm a missão de detectar a perda de pacotes e solicitar a retransmissão dos pacotes perdidos (NACK). A perda de pacotes normalmente é detectada pelos receptores ao notar um salto na seqüência de numeração dos pacotes que chegam, vindos de uma fonte determinada.

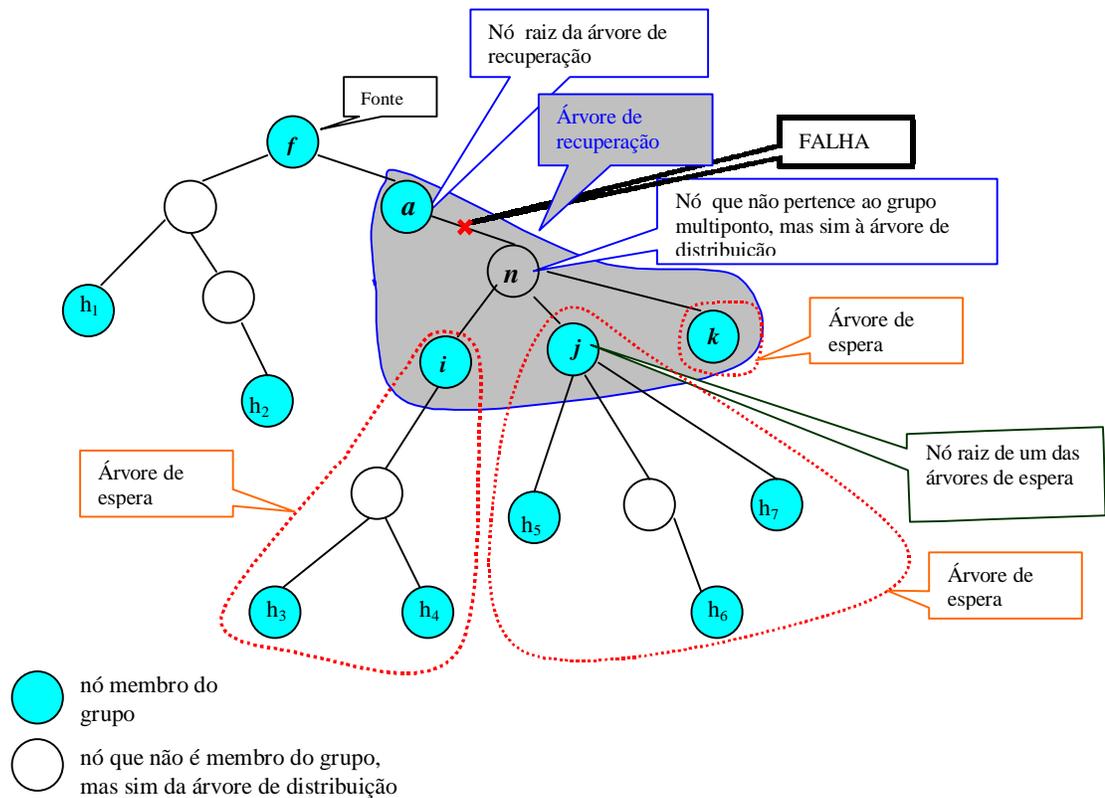


Figura V.2: Árvore de distribuição, árvore de recuperação e árvores de espera.

Se define como *árvore de recuperação* a qualquer sub-árvore da árvore de distribuição onde somente a raiz e as folhas são membros do grupo. No caso em que, a árvore de recuperação possua nós intermediários, eles não podem ser membros do grupo. Segundo essa definição, se deduz que uma árvore de distribuição pode ter muitas árvores de recuperação. Por exemplo, a árvore de distribuição da Figura V.2 possui quatro árvores de recuperação, que são: 1) a árvore com raiz em  $f$  e folhas  $h_1, h_2$  e  $a$ ; 2) a árvore com raiz  $a$  e folhas  $i, j$  e  $k$ ; 3) a árvore com raiz  $i$  e folhas  $h_3$  e  $h_4$ ; e 4) a árvore com raiz  $j$  e folhas  $h_5, h_6$  e  $h_7$ .

A definição de árvore de recuperação implica que um membro do grupo, que nem é raiz nem folha da árvore de distribuição, é simultaneamente, folha de uma determinada árvore de recuperação e raiz de outra árvore de recuperação. Por exemplo, na Figura V.2 o nó  $a$  é raiz

da árvore de recuperação circundada pela linha contínua e, simultaneamente, é folha da árvore de recuperação cuja raiz é  $f$ .

A filosofia geral do algoritmo, consiste em associar a perda de um pacote a uma árvore de recuperação, cuja raiz corresponde ao último membro do grupo não afetado pela perda. No exemplo da Figura V.2, se existe uma falha no canal  $(a,n)$ , a árvore de recuperação, associada a essa falha, é a que está circundada pela linha contínua. Portanto, os únicos nós que participam ativamente na recuperação do pacote perdido são os nós  $a, n, i, j$  e  $k$ .

Cada folha de uma árvore de recuperação tem associada uma única *árvore de espera*. A árvore de espera é uma sub-árvore da árvore de distribuição, cuja raiz é a folha que está associada. A árvore de espera contém, também, todos os nós descendentes (na árvore de distribuição) do referido membro. Por exemplo, na Figura V.2, a árvore de espera associada ao membro  $j$  é a árvore que está delimitada por uma linha segmentada que contém  $j$ . Da definição anterior deriva que, cada árvore de recuperação tem associada um número de árvores de espera igual ao número de suas folhas. Por exemplo, para a árvore de distribuição da Figura V.2, as três sub-árvores da árvore de distribuição que estão circundadas por uma linha segmentada, são as árvores de espera associadas à árvore de recuperação, que está circundada pela linha contínua.

O que permite ao algoritmo obter uma alta eficiência na recuperação do pacote perdido, é o fato que os nós pertencentes à árvore de recuperação serem os únicos a participar da recuperação da falha. Os nós que não pertencem à árvore de recuperação nem às árvores de espera, em nenhum momento ficam sabendo da ocorrência da falha. Além disso, os nós que pertencem a alguma árvore de espera que não é afetada pela falha, não participam na recuperação do pacote perdido. Por exemplo, se na Figura V.2 o erro tivesse acontecido entre os nós  $n$  e  $j$ , as árvores de espera com raízes  $i$  e  $k$  não participariam da recuperação do erro. Os únicos nós que participam na recuperação do erro são aqueles que pertencem à árvore de recuperação e que são afetados pela falha. Em particular, entre os nós desta árvore, o nó raiz (nó  $a$  no exemplo da Figura V.2) é o único membro que mantém em memória o pacote perdido e, em consequência, é o nó encarregado de retransmitir o pacote. Em relação às folhas da árvore de recuperação (nós  $i, j, k$ , na Figura V.2), somente aquelas afetadas pela

perda seriam encarregadas de solicitar a retransmissão do pacote perdido. Os nós que pertencessem a alguma árvore de espera afetada pela perda e que não são folhas da árvore de recuperação, participariam passivamente na recuperação do erro.

Considerando as definições anteriores, a operação do algoritmo de confiabilidade multiponto proposto nesta tese pode ser explicitado como se segue.

A operação normal de qualquer nó da árvore de distribuição consiste em transmitir cada pacote que recebe do seu pai (na árvore de distribuição) a todos seus descendentes. A primeira vez que um nó recebe um determinado pacote, além de transmiti-lo a seus descendentes, guarda uma cópia dele em sua memória, para o caso de ser necessária uma retransmissão.

Quando um nó membro do grupo detecta a perda de um pacote, ele passa a atuar como *folha da árvore de recuperação* do erro e como *raiz da árvore de espera* que lhe corresponde. Este nó, denotado por nó  $i$ , realiza então as seguintes operações:

1. Transmite imediatamente a todos os seus nós descendentes uma mensagem multiponto que lhes indica que foi detectado a perda do pacote. Esta mensagem é tratada como um pacote normal pelos descendentes de  $i$ , motivo pelo qual propaga-se na árvore. Ao receber a mensagem, os membros do grupo, que são descendentes de  $i$ , interpretam que o nó  $i$  tomará conta da recuperação e, portanto, ficam esperando que lhes chegue o pacote perdido. Em outras palavras, esta é uma mensagem *de inibição* que informa a detecção da perda, impedindo que os membros do grupo que o recebem enviem seu próprio NACK.
  
  2. Após transmitir a mensagem de inibição, o nó  $i$  continua transmitindo os demais pacotes, que chegam da fonte, e decide se envia ou não um NACK ao nó raiz da árvore de recuperação da falha. Para determinar se envia ou não o NACK,  $i$  efetua um experimento aleatório com distribuição Bernoulli de parâmetro  $p_1(i)$ , denotado por  $Be(p_1(i))$ . A escolha do parâmetro  $p_1(i)$  do experimento Bernoulli será explicada mais a frente. Se o resultado deste experimento for um sucesso, então o nó  $i$  envia o NACK para seu pai na árvore de distribuição.
- A execução do experimento Bernoulli tem o duplo objetivo de diminuir a latência e diminuir o tamanho da implosão que possa se produzir no nó raiz da árvore de

recuperação. A latência diminui porque o experimento Bernoulli é executado no mesmo instante que se detecta a falha. Por outro lado, a implosão diminui porque somente algumas folhas da árvore de recuperação, que efetuam o experimento Bernoulli, obtêm um resultado que lhes faz transmitir o NACK. Mais precisamente, o tamanho da implosão é igual ao número de folhas da árvore de recuperação afetadas pela falha que decidem enviar o NACK, que na maioria dos casos, é muito menor que o número de membros do grupo afetados pela falha. Segundo ver-se-á mais a frente, o algoritmo tenta conseguir que o tamanho da implosão seja aproximadamente um, o que obtém-se escolhendo apropriadamente o parâmetro  $p_i(i)$  da distribuição Bernoulli.

3. Independente de ter enviado ou não o NACK, o nó  $i$  inicia um *timeout* ( $TO$ ), para esperar o pacote perdido. O  $TO$  deve ser maior ou igual ao tempo de propagação de ida e volta da folha que fica mais longe da árvore de recuperação até a raiz desta árvore, somado ao tempo de retransmissão do pacote perdido.
4. No caso em que o  $TO$  expire sem que se tenha recebido o pacote solicitado, o nó  $i$  repete os passos 2 e 3. Dessa vez, o parâmetro da distribuição Bernoulli, que serve para decidir se envia ou não um NACK, terá uma probabilidade  $p_n(i)$ , onde  $n$  corresponde ao número de vezes que o nó  $i$  executou o passo 2 do algoritmo, com relação à recuperação de um mesmo erro..  
O nó  $i$  repete o passo 4 até receber o pacote perdido.
5. Quando o nó  $i$  receber o pacote solicitado, ele o retransmite a todos os seus descendentes. Isso faz com que os outros membros do grupo, que pertencem à árvore de espera de  $i$ , recuperem o pacote perdido sem ter participado ativamente na sua recuperação.
6. Quando um nó interno da árvore de recuperação recebe um NACK, ele o retransmite para seu nó pai na árvore de distribuição.  
Este passo do algoritmo assegura que as mensagens de NACK se propaguem para a raiz da árvore de recuperação.

Cada vez que um membro do grupo, denotado por  $a$ , recebe um NACK em relação a um determinado pacote que tem armazenado, ele realiza as seguintes operações:

7. Elimina a mensagem NACK, fazendo com que seus nós antecessores não saibam que ocorreu uma falha, evitando assim a explosão de retransmissões, já que unicamente o nó  $a$  será o encarregado de retransmitir o pacote perdido.

Dos passos 2, 6 e 7 do algoritmo, deriva-se que a trajetória das mensagens de NACK vai das folhas da árvore de recuperação afetada pela falha até a raiz desta árvore.

8. Após eliminar a mensagem de NACK, o nó  $a$  retransmite o pacote solicitado para todos seus descendentes, fazendo com que todos os nós que são raiz de alguma árvore de espera recebam o pacote perdido.
9. Finalmente, ao receber a retransmissão de um pacote perdido, a *raiz de uma árvore de espera*, se foi afetada pela falha, o transmite a todos seus descendentes; em caso contrário, descarta o pacote.

#### V.4.3.1 Determinação da probabilidade de enviar um NACK .

A filosofia do algoritmo consiste em fazer com que a probabilidade de nó folha da árvore de recuperação enviar um NACK seja proporcional à distância desse nó à raiz da árvore de recuperação. O motivo dessa definição vem do fato que as folhas longe dentro da árvore de recuperação são as que têm maior probabilidade de serem afetadas pela falha.

Seja  $p_1(i)$  a probabilidade de que a folha  $i$  da árvore de recuperação envie um NACK na primeira vez que executa o passo 2 do algoritmo. Seja  $\alpha$ , com  $\alpha > 1$ , um parâmetro que é usado para conseguir um equilíbrio entre a latência e a implosão. O valor de  $p_1(i)$  é dado por:

$$p_1(i) = \frac{\alpha f(c_i)}{S} \quad (\text{V.1})$$

onde  $c_i$  é o caminho que vai da raiz da árvore de recuperação até uma determinada folha  $i$ ;  $f(c_i)$  corresponde a probabilidade que um pacote enviado através de  $c_i$  se perca;  $A$  identifica a

árvore de recuperação; e  $S$  corresponde à soma da probabilidade de falha de todos os caminhos que vão da raiz até todas as folhas de  $A$ .

Seja  $x_l$  a probabilidade de perda de pacote associada ao canal  $l$ , então, usando o fato de que os canais falham de forma independente, se tem que  $f(c_i)$  e  $S$  são dados por:

$$f(c_i) = \left( 1 - \prod_{\forall l \in c_i} (1 - x_l) \right) \quad (\text{V.2})$$

$$S = \sum_{\forall \text{hoja } i \in A} f(c_i) \quad (\text{V.3})$$

A atribuição de probabilidades de V.1 supõe que o pacote perdido afeta a todas as folhas da árvore de recuperação (como é o caso ilustrado no exemplo da Figura V.2). O valor de  $p_1(i)$ , dado por (V.1), foi definido par que, com alta probabilidade, se gere ao menos um NACK na primeira iteração do algoritmo. Assim, uma baixa latência é obtida e consegue-se que o número de NACKs enviado seja baixo (baixa implosão).

Se após o primeiro *timeout* um receptor não tiver recebido o pacote perdido isto significa, com alta probabilidade, que a falha não ocorreu no lugar suposto previamente (no exemplo da Figura V.1, a falha poderia ter afetado somente à árvore com raiz em  $i$  e não a às árvores com raiz em  $j$  ou  $k$ ). Por esse motivo, na iteração seguinte do algoritmo, cada membro afetado pela falha supõe que ela ocorreu no canal que está um passo mais próximo, em relação à suposição realizada na iteração anterior.

Como consequência, para determinar  $p_2(i)$  mantendo a filosofia das equações (V.1), (V.2) e (V.3), cada folha da árvore de recuperação afetada pela falha deveria utilizar uma nova árvore de recuperação, que seria aquela que a inclui e cuja raiz corresponderia ao nó que está um passo antes da verdadeira raiz da árvore de recuperação (por exemplo, na Figura V.1 a folha  $j$  suporia que a raiz é o nó  $n$ ). Como isso não pode ser realizado sem alterar a definição das árvores de recuperação e sem complicar a operação do algoritmo, uma alternativa para determinar  $p_2(i)$  é usar uma heurística que simule a idéia anterior.

Uma heurística razoável consiste em aumentar exponencialmente a probabilidade de envio de NACK. Por exemplo, se a árvore de distribuição é de grau  $r$  (ou  $r$ -ário), a probabilidade de

enviar um NACK, na  $n$ -ésima execução do passo 2 do algoritmo (relacionada a uma mesma falha), deve ser incrementada por um fator  $r$  com relação à probabilidade usada na iteração prévia, isto é:

$$p_n(i) = \min \left\{ \begin{array}{l} 1 \\ r \cdot p_{n-1}(i) \end{array} \right\}, \quad \mathbf{n} > \mathbf{1} \quad (\text{V.4})$$

Onde  $i$  é uma folha da árvore de recuperação afetada pela falha.

#### V.4.4 Análise de Desempenho do Algoritmo

As duas medidas mais usadas para caracterizar o comportamento dos mecanismos de recuperação de erros são o tamanho da implosão e a latência, como já mencionado. Seja  $I$  a variável aleatória que representa o tamanho da implosão de NACKs e seja  $E[I]$  seu valor médio. Seja  $T$  a variável aleatória que representa a latência e  $E[T]$  o seu valor médio.

O desempenho de um algoritmo para transmissão multiponto confiável depende de muitos fatores, entre os quais podem ser mencionados a topologia da árvore de distribuição e quais dos nós desta árvore são membros do grupo. Por este motivo, para avaliar o desempenho do algoritmo deveriam ser analisadas diversas variantes relativas à topologia da árvore de distribuição e a colocação dos membros do grupo dentro da topologia, dando ênfase aos casos mais representativos das situações que acontecem na prática.

##### V.4.4.1 Topologia Estrela

A seguir apresenta-se a análise de desempenho para a árvore de distribuição mostrada na Figura V.3. Esta topologia foi escolhida devido ela ser uma topologia típica de Mbone para redes regionais [RIZ97]. Além disso, o mesmo caso foi analisado em [HUI96] para avaliar o protocolo SRM, o que permite comparar o desempenho do algoritmo proposto neste trabalho em relação ao SRM, que é um dos algoritmos de confiabilidade multiponto mais conhecidos.

Tal como se mostra na Figura V.3, o grupo multiponto está constituído por  $R$  membros, dos quais  $R-1$  são receptores e o outro é a fonte ( $f$ ) que transmite os pacotes (em realidade,  $f$  pode

pertencer ou não ao grupo). Também existe um nó intermediário, que não pertence ao grupo, e que somente se encarrega de retransmitir os pacotes que lhe chegam da fonte aos demais membros do grupo multiponto (e de retransmitir NACKs para a fonte).

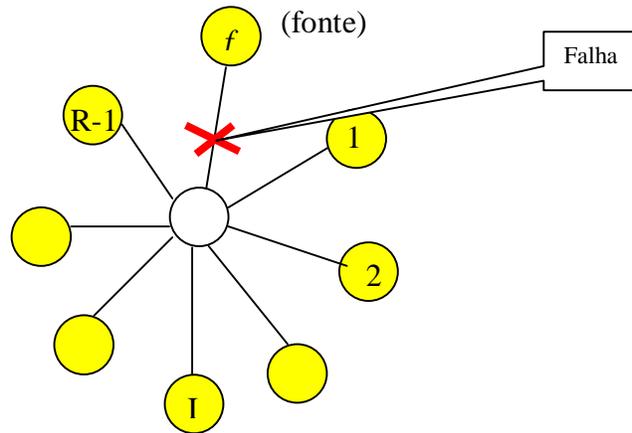


Figura V.3: Topologia estrela.

Com o objetivo de simplificar a análise, supõe-se que todos os canais da rede têm a mesma probabilidade  $x$  de perder um pacote. Assim, a probabilidade  $f(c_i)$  (ver equação V.2) é dada por:

$$f(c_i) = [1 - (1 - x)^2] = x(2 - x), \quad 1 \leq i \leq R - 1 \quad (\text{V.5})$$

Logo,  $S$  (ver equação (V.3)) é dada por:

$$S = \sum_{i=1}^{R-1} f(c_i) = (R - 1)x(2 - x) \quad (\text{V.6})$$

Substituindo (V.5) e (V.6) em (V.1), obtém-se que a probabilidade de uma folha enviar um NACK na primeira vez que se detecta a perda de um determinado pacote (na primeira iteração do algoritmo) é dada por:

$$p_1(i) = \min \left[ 1, \frac{\alpha}{R - 1} \right], \quad 1 \leq i \leq R - 1 \quad (\text{V.7})$$

Para calcular a probabilidade que uma folha envie um NACK em iterações posteriores ( $n > 1$ ), considera-se que a árvore é de grau  $R-1$  (isto é deduzido pela raiz da árvore de recuperação ao trocar mensagens com suas folhas). Em consequência, utilizando a equação (V.4) e considerando que  $\alpha$  é sempre maior que 1 (ver a análise dos exemplos que são mostrados mais adiante), tem-se que:

$$p_n(i) = 1; \quad n > 1; \quad 1 \leq i < R \quad (\text{V.8})$$

Devido a  $p_n(i)$ ,  $n \geq 1$ , não depender de  $i$ , nas equações seguintes omite-se o índice  $i$  que identifica uma folha específica, isto é:  $p_n(i) = p_n$ .

As equações (V.7) e (V.8) determinam o valor de  $p_n$ ,  $n \geq 1$ , totalmente especificando a operação do algoritmo. A seguir avalia-se a latência média e a implosão média do algoritmo.

Devido ao fato queda perda de um pacote poder acontecer em qualquer canal da árvore de recuperação e que podem ser feitas várias iterações do passo 2 do algoritmo (com relação a uma mesma falha), para avaliar  $E[T]$  condiciona-se e descondiciona-se tanto o canal que acontece a falha, quanto o número de iterações do passo 2 do algoritmo. Portanto, aplicando o teorema de probabilidades totais, tem-se que:

$$E[T] = \sum_{i=0}^{R-1} \left( \sum_{n=1}^N E[T | n, i] P(n, i) \right) q(i) \quad (\text{V.9})$$

onde o índice  $i$  identifica o canal no qual acontece a falha. Se  $i=0$ , a falha acontece no canal incidente à fonte, e se  $1 \leq i < R$ , a falha acontece no canal incidente ao membro  $i$ . O índice  $n$  identifica o número de iterações do passo 2 do algoritmo e  $N$  corresponde ao máximo valor de  $n$ . Isto é,  $N$  corresponde ao número de iterações na quais  $p_n(i) = 1$  (devido à equação (V.4),  $N=2$ ). Por outro lado,  $P(n, i)$  é a probabilidade de que seja enviado ao menos um NACK (por parte de alguns dos membros afetados pela falha) na  $n$ -ésima iteração do passo 2 do algoritmo, dado que a falha aconteceu no canal incidente ao membro  $i$ . Por exemplo,  $P(1,0)$  é a probabilidade de enviar ao menos um NACK na primeira iteração do passo 2 do algoritmo, dado que a falha acontece no canal incidente à fonte.  $E[T/n, i]$  corresponde ao valor médio da latência, dado que a falha aconteceu no canal incidente ao membro  $i$  e que o NACK foi

enviado na  $n$ -ésima iteração do passo 2 do algoritmo. A probabilidade de que a falha ocorra no canal  $i$ , é dada por  $q(i)$ .

Usando o fato de que a falha acontece com igual probabilidade em todo canal, obtém-se que:  $q(i) = \frac{1}{R}$ ,  $0 \leq i < R$ . Por outro lado, devido a suposição de que os NACKs e os pacotes retransmitidos não são perdidos, conclui-se que:  $E[T | n, i] = n \cdot TO$

Para avaliar a latência média, é necessário calcular-se os valores de  $P(n, i)$ .

Para avaliar  $P(1,0)$ , em primeiro lugar note que a probabilidade de uma folha não enviar um NACK na primeira iteração do passo 2 do algoritmo, é dada por  $(1-p)$ . Além disso, como todas as folhas da árvore de recuperação atuam de forma independente, a probabilidade de ao menos uma folha enviar um NACK à raiz da árvore de recuperação, na primeira iteração do passo 2 do algoritmo, é:  $P(1,0) = 1 - (1-p)^{R-1}$ .

Para calcular  $P(2,0)$  deve-se observar que, se de acordo com a equação (V.8) nenhuma das folhas mandou um NACK na primeira iteração, na segunda vez que se executa o passo 2 do algoritmo todas as folhas enviarão um NACK com probabilidade igual a 1, assim:  $P(2,0) = (1-p)^{R-1} \cdot 1$ .

Por outro lado,  $P(1,i) = p$ ;  $1 \leq i < R$ , já que uma folha da árvore de recuperação envia com probabilidade  $p$  um NACK, na primeira iteração do passo 2 do algoritmo. E  $P(2,i) = (1-p)$ ;  $1 \leq i < R$ , já que, se a folha não enviar o NACK na primeira iteração, o fará com probabilidade 1 na segunda iteração (concordando com a equação V.8).

Substituindo as fórmulas do parágrafo anterior em (V.9), tem-se que a latência média normalizada é dada por:

$$E[T] = \frac{1}{R} \left\{ 1 + (1-p)^{(R-1)} + (R-1)(2-p) \right\} \quad (V.10)$$

Para efetuar o cálculo do tamanho médio da implosão,  $E[I]$ , deve-se proceder de forma análoga ao cálculo de  $E[T]$ . Isto é, condicionando e descondicionando tanto com relação ao

canal no qual ocorre a falha, quanto ao número de iterações do passo 2 do algoritmo. Como consequência, aplicando o teorema de probabilidades totais, tem-se que:

$$E[I] = \sum_{i=0}^{R-1} \left( \sum_{n=1}^N E[I | n, i] P(n, i) \right) q(i) \quad (\text{V.11})$$

onde os índices  $i$ ,  $n$ ,  $N$  e as probabilidades  $P(n, i)$  e  $q(i)$  correspondem às definições e valores mencionados anteriormente. Por sua vez,  $E[I | n, i]$  corresponde ao valor médio da implosão, dado que a falha acontece no canal incidente ao membro  $i$  e que o NACK foi enviado na  $n$ -ésima iteração do passo 2 do algoritmo.

Para avaliar  $E[I | 1, 0]$ , usa-se os seguintes fatos: 1) a implosão somente acontece se ao menos um dos receptores transmite seu NACK; 2) os NACKs e os pacotes retransmitidos não se perdem (por hipótese de análise) e; 3) o número de NACKs transmitidos tem distribuição binomial de parâmetros  $(R-1)$  e  $p$ . Como consequência, usando a definição de valor esperado condicional, tem-se que:

$$E[I | 1, 0] = \frac{\sum_{i=1}^{R-1} i \Pr[\text{se transmitem } i \text{ NACKs na 1a. iteração}]}{1 - \Pr[\text{não transmitir NACKs na 1a. iteração}]} = \frac{\alpha}{1 - \left(1 - \frac{\alpha}{R-1}\right)^{R-1}}$$

Para calcular  $E[I | 2, 0]$ , usa-se o fato que, nesse caso, nenhuma folha envia seu NACK na primeira iteração do algoritmo e, devido à equação (V.8), todas as folhas enviam, com probabilidade igual a 1, seu NACK na segunda iteração do algoritmo. Portanto, a implosão neste caso é:  $E[I | 2, 0] = (R-1)$ . Por outro lado,  $E[I | 1, i] = E[I | 2, i] = 1$ ;  $1 \leq i < R$ , já que nesses casos somente a folha afetada pela falha envia um NACK à raiz da árvore de recuperação.

Substituindo os resultados da análise dos parágrafos precedentes na equação (V.11), tem-se que a implosão média é dada por:

$$E[I] = \frac{R-1}{R} \left\{ p + (1-p)^{R-1} + 1 \right\} \quad (\text{V.12})$$

A Figura V.4 mostra o gráfico da implosão média ( $E[I]$ ) versus o tamanho do grupo multiponto ( $R$ ), para diferentes valores de  $\alpha$ .

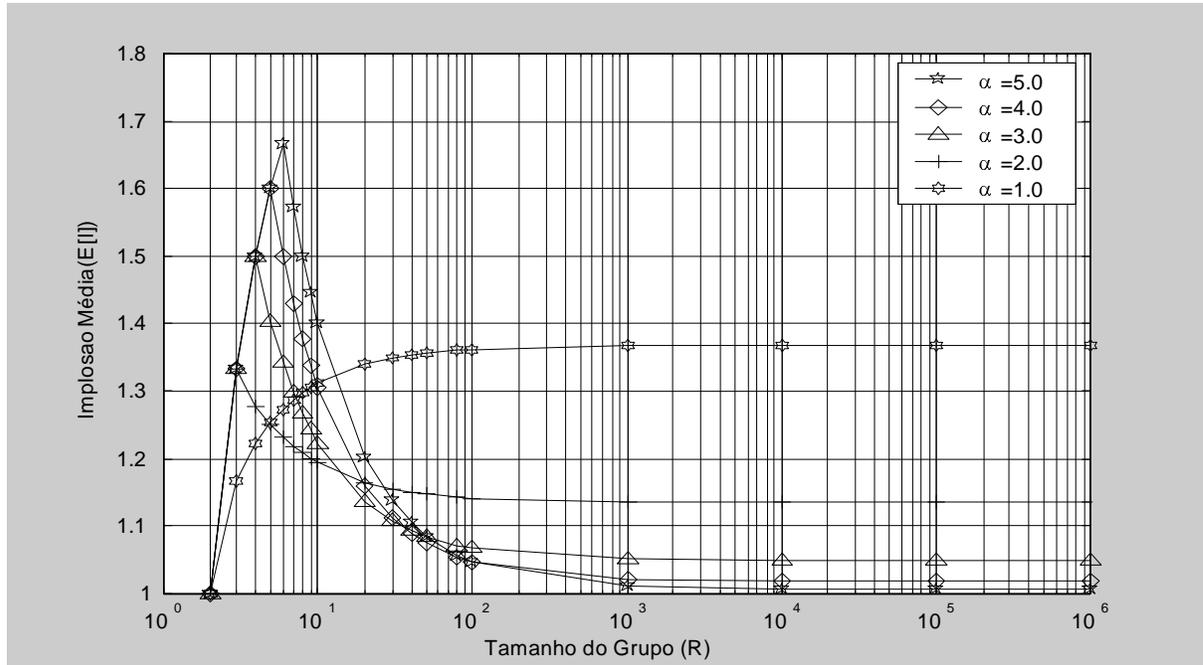


Figura V.4. Implosão Média ( $E[I]$ ) versus Tamanho do Grupo Multiponto ( $R$ ) para diferentes valores de  $\alpha$ .

Para valores de  $R-1$  que variam entre 2 e  $\alpha+1$ ,  $E[I]$  é monotonamente crescente em função de  $R$ . Isto deve-se a que, nesses casos,  $p=1$  e, portanto, a equação (V.12) se reduz a  $E[I] = \frac{2(R-1)}{R}$ . Em consequência, o valor máximo de  $E[I]$  é atingido quando  $(R-1) = \alpha$ , em

cujo caso  $E[I] = \frac{2\alpha}{\alpha+1}$ . Por exemplo, na curva correspondente a  $\alpha=5$ , observa-se que

$E[I]=1,67$  quando  $R=\alpha+1$ .

Ao aumentar o valor de  $R$  além de  $(\alpha+1)$ , a implosão média tende assintoticamente ao valor  $(1+e^{-\alpha})$ . Isto é devido em parte ao fato que, quando  $R$  é um valor muito grande, o termo  $(1-p)^{R-1}$  tende a  $e^{-p(R-1)} = e^{-\alpha}$ . Além disso, como nesse caso  $p$  tende a zero e o fator  $(R-1)/R$  tende a 1, para valores muito grandes de  $R$ , conclui-se que  $E[I]$  tende ao valor  $(1+e^{-\alpha})$ . Por exemplo, na Figura V.4, se observa que, se  $\alpha=5$ , para valores grandes de  $R$  o valor médio da implosão tende a 1.006. Além disso, pelo mesmo motivo anterior, ao aumentar o valor de  $\alpha$ , para valores grandes de  $R$ ,  $E[I]$  tende ao valor ideal da implosão, que é 1.

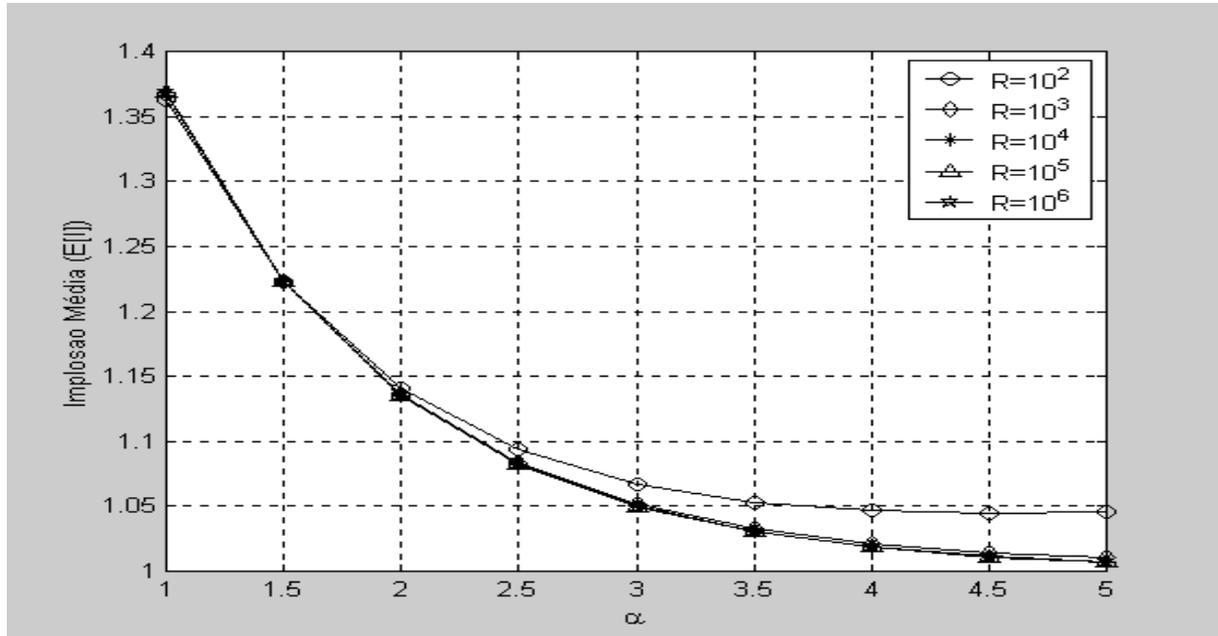


Figura V.5. Implosão Média versus  $\alpha$

A Figura V.5 ilustra a implosão média em função de  $\alpha$ , para tamanhos de grupo maiores ou iguais a 100 membros. Nesse caso, nota-se que, a medida que o tamanho do grupo aumenta, a implosão média tende exponencialmente a 1 (o valor ideal). Isso deve-se ao fato que, tal como indicado anteriormente, nesses casos,  $E[I]$  corresponde aproximadamente a  $(1+e^{-\alpha})$ . Outra forma mais prática de explicar essa tendência de  $E[I]$  é notar que, nesses casos, a falha quase com certeza afeta somente a uma folha (fenômeno que acontece com probabilidade  $(R-1)/R$ ) e, em consequência, somente ela é quem envia um NACK à raiz da árvore de recuperação.

O gráfico da Figura V.6 mostra a latência média normalizada versus o tamanho do grupo multiponto ( $R$ ), para diferentes valores de  $\alpha$ . Na figura, observa-se que, para grupos de tamanho menor que  $10^3$ , quando  $\alpha$  aumenta, a latência diminui. Isso deve-se ao fato que quando ocorre uma falha, a probabilidade que algum dos receptores envie um NACK na primeira iteração do algoritmo aumenta ao se aumentar  $\alpha$ . Além disso, para grupos maiores ou iguais a  $10^3$  membros, a latência é aproximadamente igual a 2, independente do valor de  $\alpha$ . Isso deve-se ao fato que quando o grupo aumenta, com probabilidade alta também aumenta a probabilidade que a perda de pacotes se produza nos canais que ficam mais longe da raiz da árvore de recuperação. Nesse caso, somente um membro é afetado pela falha, aquele que

envia o NACK na segunda iteração do algoritmo (ver equação V.8) com probabilidade quase igual a 1.

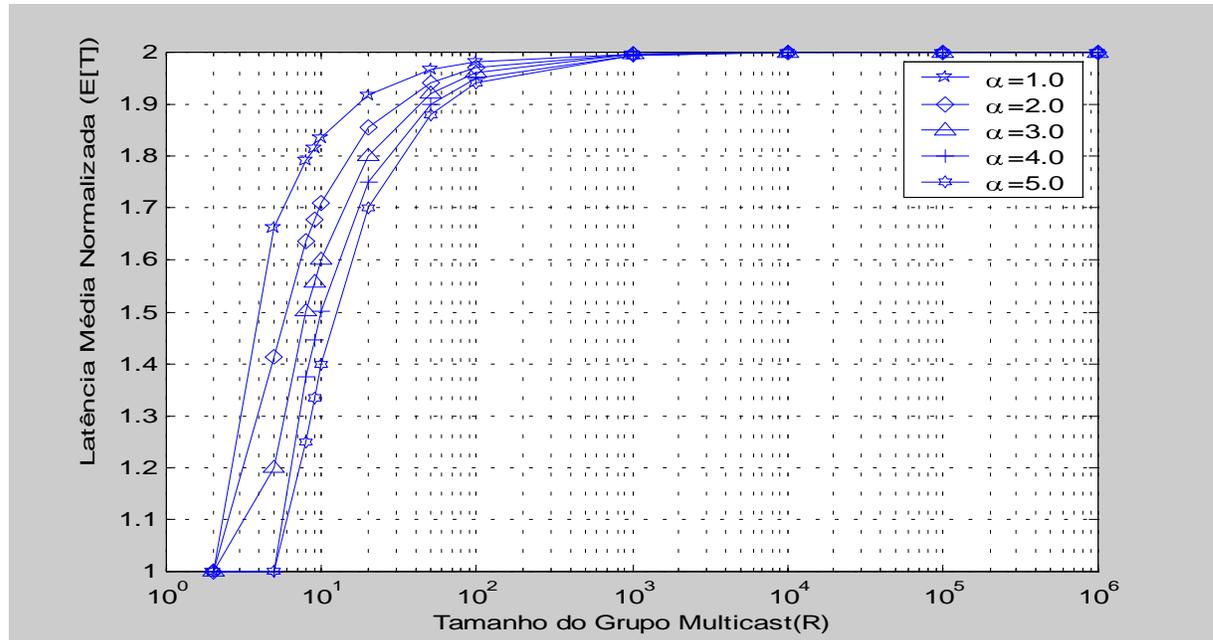


Figura V.6. Latência média normalizada versus Tamanho do Grupo Multiponto (R) para diferentes valores de  $\alpha$

A Figura V.7 mostra a latência média normalizada em função de diferentes valores de  $\alpha$ , para tamanhos de grupo que variam entre 10<sup>2</sup> e 10<sup>6</sup> membros. No gráfico, pode ser visto que, para tamanhos de grupo maiores ou iguais a 10<sup>3</sup> membros, a latência média normalizada mantém-se na ordem de 2, independentemente do valor de  $\alpha$ . Isso se deve ao mesmo fenômeno que foi explicado anteriormente, isto é: nesses casos, quase com certeza, a falha afeta somente a um membro, aquele que envia, quase com certeza, o NACK na segunda iteração do algoritmo.

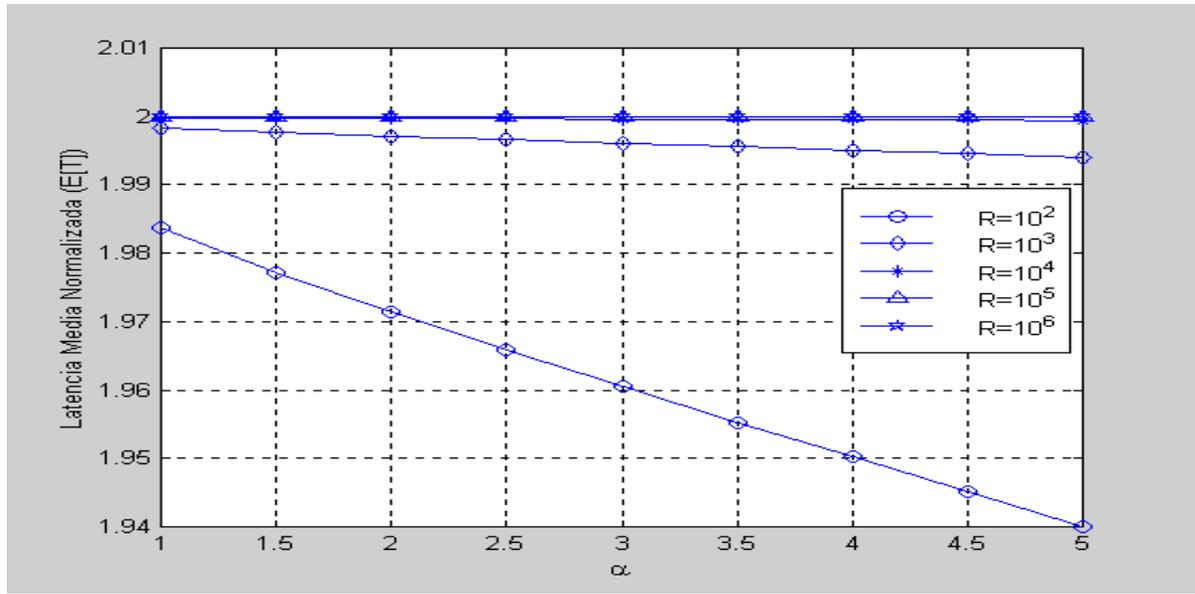


Figura V.7. Latência Média Normalizada em função de  $\alpha$

Dos gráficos das Figuras V.4, V.5, V.6 e V.7 pode-se concluir que o algoritmo é bastante escalável, e que, escolhendo adequadamente o valor de  $\alpha$ , se pode sintonizar em algum ponto de operação conveniente.

Para se escolher adequadamente um determinado valor de  $\alpha$ , deve-se ter presente que existe um compromisso entre o benefício que pode obter o usuário e o benefício que pode obter o sistema. Do ponto de vista do usuário, uma eleição de  $\alpha$  é boa quando origina uma baixa latência, já que isto implica em um menor tempo de resposta na transmissão de sua informação. Por outro lado, do ponto de vista do sistema, uma escolha adequada de  $\alpha$  é aquela que não produz um *overhead* significativo, devido a uma alta implosão, na raiz da árvore de recuperação. Portanto, dos gráficos anteriores (Figuras V.4, V.5, V.6 e V.7), pode-se concluir que uma boa eleição do valor de  $\alpha$  fica em torno de 2.5. A latência média normalizada, nesse caso, fica próxima de 2 e a implosão média é menor que 1.1 pacotes. Também se pode afirmar que a escolha de  $\alpha=2.5$  faz com que o algoritmo seja altamente escalável e, portanto, as medidas globais da rede, como o retardo e a vazão, não se degradarão significativamente devido à perda de pacotes, ainda para grupos de tamanho muito grande.

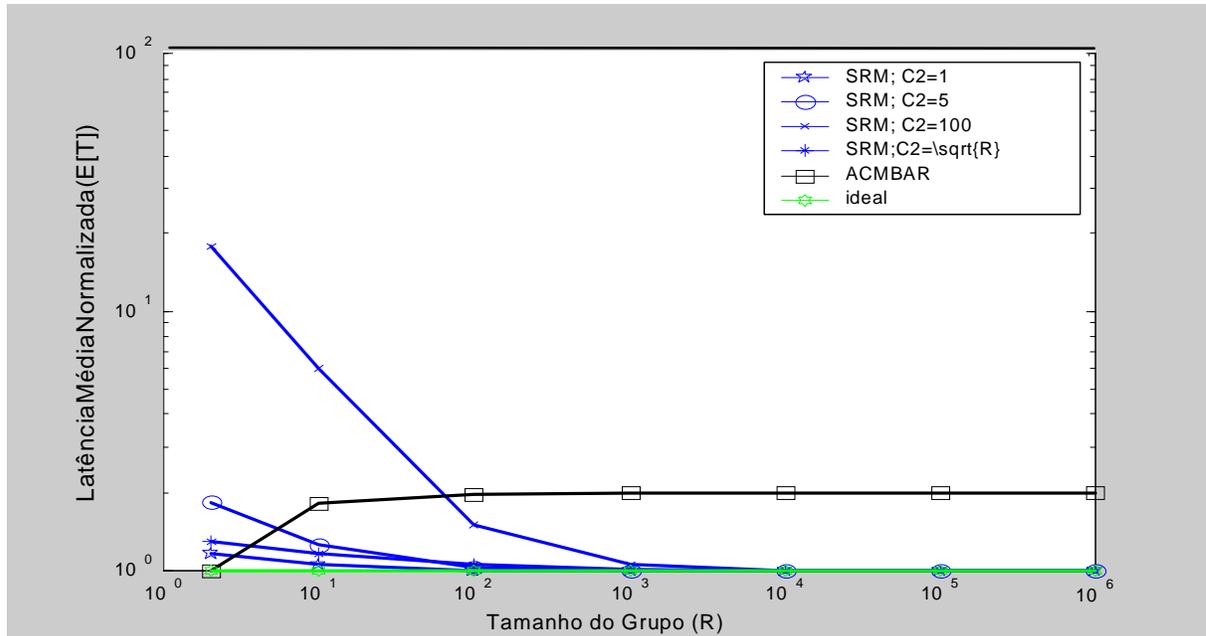


Figura V.8. Latência média versus tamanho do grupo para o algoritmo ACMBAR ( $\alpha=2.5$ ) e o algoritmo SRM (para diferentes valores de  $C_2$ ).

A Figura V.8 apresenta uma comparação entre a latência obtida pelo algoritmo proposto neste trabalho quando  $\alpha=2.5$ , o algoritmo SRM [FLO97] e o algoritmo ideal (isto é, aquele que consegue uma latência igual a 1), em função do tamanho do grupo multiponto. Como o algoritmo SRM depende basicamente de 2 parâmetros (ver [FLO97]), que são  $C_1$  e  $C_2$ , a figura ilustra a latência para o caso em que  $C_1=0$  (já que este é o melhor valor do parâmetro  $C_1$  para o exemplo sob análise [FLO97]) e para diferentes valores do parâmetro  $C_2$ .

No gráfico da Figura V.8, pode ser visto que, para grupos menores que 10 membros, o algoritmo ACMBAR se comporta melhor que o algoritmo SRM. Entre 10 e 10<sup>2</sup> membros, o desempenho dos dois algoritmos é comparável; e para grupos maiores que 10<sup>2</sup> membros, o algoritmo ACMBAR apresenta uma latência média superior ao SRM. Por exemplo, para grupos de 10<sup>6</sup> membros, ACMBAR obtém uma latência média de aproximadamente 2, enquanto a latência média entregue por SRM é de 1.005 pacotes.

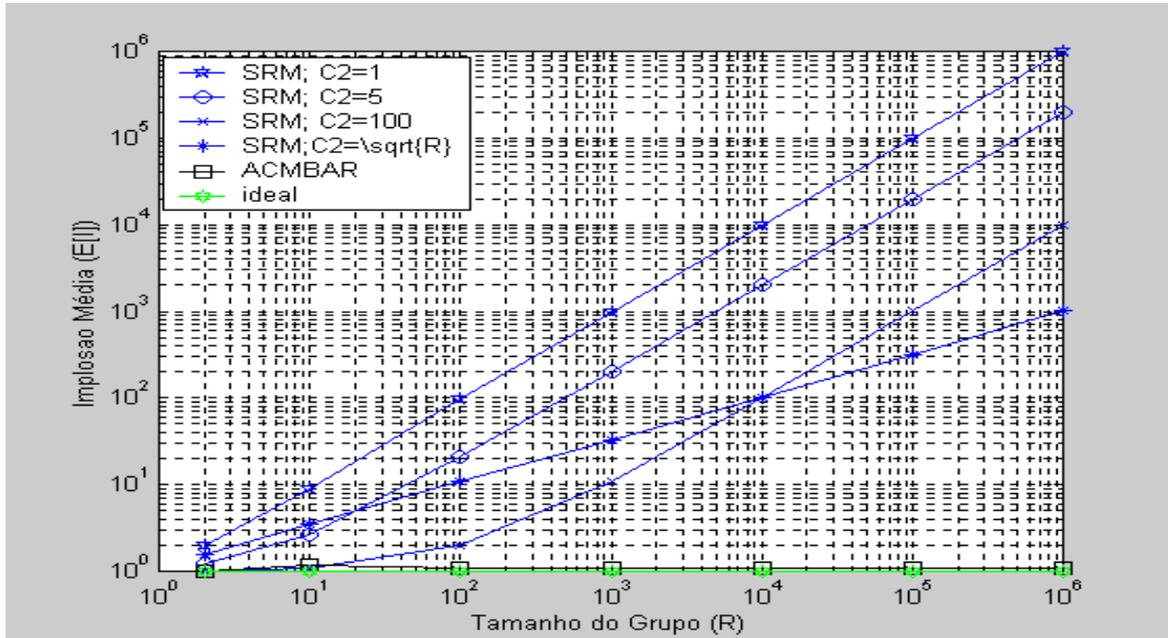


Figura V.9. Implosão versus tamanho do grupo para o algoritmo ACMBAR ( $\alpha=2.5$ ) e o algoritmo SRM (para diferentes valores de  $C_2$ ).

A Figura V.9 mostra o gráfico da implosão média versus o tamanho do grupo, para os algoritmos ACMBAR, SRM e ideal. Para o caso do algoritmo SRM, consideram-se o parâmetro  $C_1=0$  e diferentes valores do parâmetro  $C_2$ . Na figura, pode-se observar que o algoritmo ACMBAR é muito escalável, já que o valor da implosão permanece quase constante para qualquer tamanho de grupo multiponto, ao contrário da pouca escalabilidade que apresenta SRM para qualquer valor de  $C_2$ . Por exemplo, para um tamanho de grupo igual a  $10^6$  membros, o ACMBAR apresenta uma implosão média de 1.08 pacotes e o SRM apresenta uma implosão média maior que  $10^3$  pacotes (para qualquer valor de  $C_2$ ).

Ao analisar globalmente o desempenho de SRM (isto é, observando tanto a latência quanto a implosão), pode-se notar que, escolhendo adequadamente o parâmetro  $C_2$ , o algoritmo SRM possui uma baixa latência média (por exemplo, quando  $C_2=1$  em Figura V.8). No entanto, a implosão obtida por SRM para o mesmo valor do parâmetro  $C_2$  é muito grande (é praticamente linear em relação ao tamanho do grupo, ver Figura V.9). No outro extremo, se  $C_2$  aumenta, consegue-se diminuir a implosão, mas ela se mantém controlada somente para tamanhos de grupo menores que  $10^2$ , já que ao aumentar  $R$  a implosão aumenta quase que linearmente. Além disso, o aumento de  $C_2$  provoca o aumento da latência para valores de  $R$  menores que  $10^2$ .

Com relação ao desempenho global do algoritmo ACMBAR, pode-se concluir que ele apresenta uma baixa latência e uma baixa implosão média, para qualquer tamanho de grupo. Por exemplo, ACMBAR entrega uma latência média normalizada menor que 2 (para qualquer R) e uma implosão média de aproximadamente 1.1 pacotes, também para qualquer R. Mais ainda, nas Figura V.8 e V.9, pode ser observado que o algoritmo ACMBAR entrega uma latência média normalizada que fica razoavelmente próximo da latência média normalizada ideal (que é 1) e uma implosão média que também está próximo da implosão média ideal (um pacote), para qualquer tamanho do grupo multiponto.

Finalmente, cabe também notar que, para a recuperação do erro, no algoritmo de SRM participam todos os membros da árvore de distribuição (recuperação global), ao passo que em ACMBAR participam somente os membros da árvore de recuperação (recuperação local). Em geral, pode-se afirmar que um algoritmo que usa recuperação local é mais eficiente (obtem uma menor latência e uma menor implosão) que um algoritmo de recuperação global. Por esse motivo, o exemplo recém analisado é um caso que favorece a SRM em relação de ACMBAR. Tendo em vista que a árvore de distribuição da Figura V.2 é igual à árvore de recuperação (isto é, no exemplo não se exprime a diferença que existe entre recuperação global e recuperação local). Porém, apesar desta comparação favorável a SRM, o desempenho global de ACMBAR é superior, por: possuir uma alta escalabilidade, ter uma implosão bastante menor que a apresentada por SRM, e ter uma latência comparável à mostrada por SRM.

#### V.4.4.2 Outras Topologias

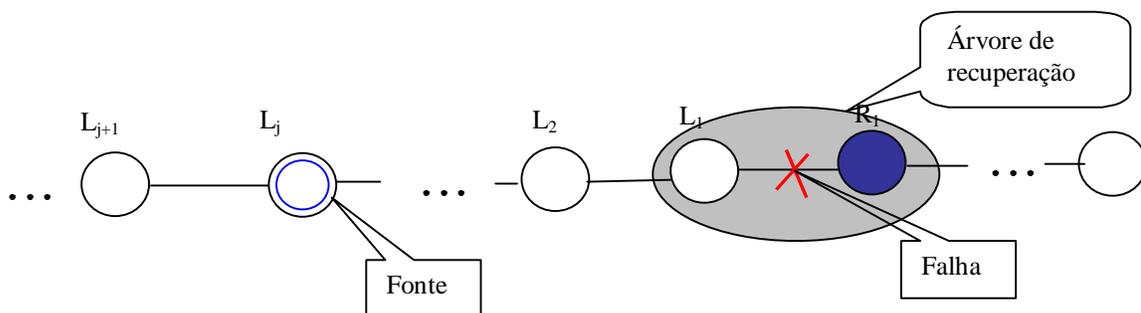


Figura V.10. Topologia cadeia

Como foi dito anteriormente, para analisar o desempenho de um algoritmo de confiabilidade multiponto, é necessário avaliar seu comportamento sob diferentes condições de operação. Por este motivo, na literatura [NON97B, RAM98] tem sido analisado desempenho de algoritmos multiponto confiável operando em outras topologias, como as que são apresentadas nas Figura V.10 e na Figura V.11.

Para esses casos é fácil notar que o algoritmo ACMBAR comporta-se de forma quase que ideal. Isto deve-se a que, em ambos casos, cada árvore de recuperação ser constituída somente pela raiz e uma folha (o membro afetado pela falha). Nesses casos, cada vez que se produz a perda de um pacote, a folha da árvore de recuperação correspondente solicita imediatamente (isto é, no instante que detecta a perda) sua retransmissão (envia o NACK). Essa situação faz com que a latência seja mínima e, por outro lado, devido a árvore de recuperação conter somente uma folha, com que a implosão seja igual a 1.

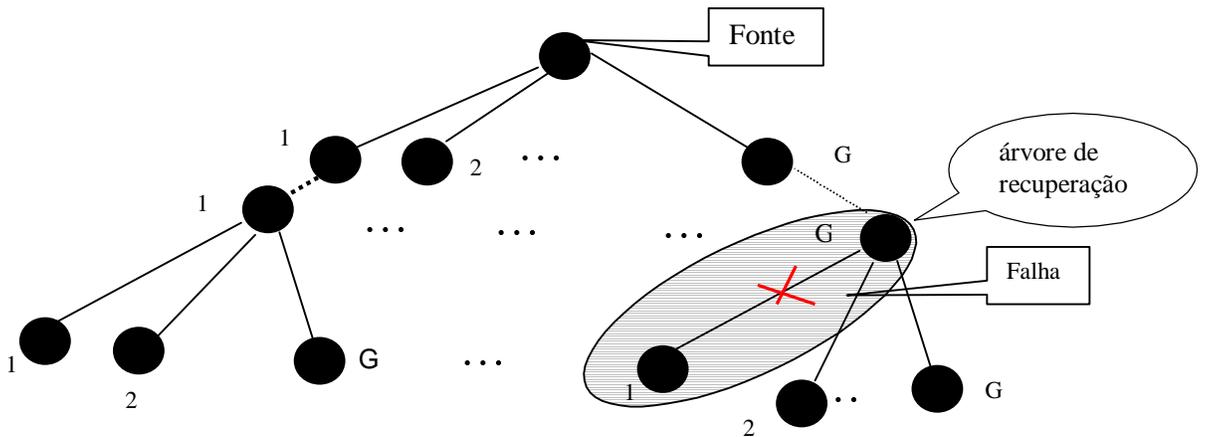


Figura V.11. Topologia em árvore balanceada de grau  $G$ , onde todos os nós pertencem ao grupo multiponto.

## V.5 Resumo

Neste capítulo foi apresentada uma conceituação sobre o problema de transmissão multiponto confiável. Também foi gerada uma taxonomia, que permite classificar alguns dos protocolos mais importantes publicados até o presente momento.

Foi proposto um novo algoritmo para a recuperação de erros de uma transmissão multiponto confiável, denominado ACMBAR, e avaliado analiticamente seu desempenho.

Sobre a base dos exemplos analisados, pode ser dito que ACMBAR apresenta uma alta escalabilidade, para qualquer tamanho de grupo multiponto. Pode ser observado ainda que a latência média e a implosão média apresentam valores próximos do ideal.

# Capítulo VI

## Conclusões e Trabalhos Futuros

### VI.1 Introdução

Neste capítulo são resumidas as conclusões mais importantes desta tese e os trabalhos futuros.

O objetivo geral desta tese foi melhorar a transmissão da informação, através de redes de computadores com QoS. Foram analisados algoritmos tanto para redes comutadas por pacotes, exemplificadas pelas redes TCP/IP, quanto redes com comutação de rótulos, exemplificados pela rede ATM. Especificamente, os parâmetros de qualidade de serviço de interesse foram a probabilidade de perda de células e o retardo no sistema. Entretanto, outras medidas de desempenho foram também calculadas, como resumido a seguir.

### VI.2 Conclusões

Uma das contribuições desta tese foi propor um novo método analítico baseado em “Performability” para avaliar diferentes medidas de desempenho.

Em primeiro lugar, na análise de desempenho de multiplexadores para tráfego de voz, a contribuição mais importante foi o desenvolvimento de um novo método para a obtenção de novas medidas de desempenho, que caracterizam o comportamento do multiplexador durante os períodos de congestionamento e períodos sem congestionamento, como apresentado na Seção IV.2, exemplificado para redes ATM. Esse método foi usado para avaliar a FDC do tempo de residência das células no multiplexador, uma tarefa muito complexa, pois essa distribuição depende não apenas das células que já estavam no multiplexador no instante de chegada do par de células sob análise, mas também daquelas que chegam durante o tempo de permanência do par de células. O método foi aplicado a avaliação de diversas medidas de desempenho durante períodos de congestionamento. Entre elas a FDC do comprimento do período de congestionamento, a FDC das células descartadas no período de

congestionamento, a FDC do número de células perdidas durante o período de congestionamento. Os valores médios dessas medidas também foram calculados.

Todas as medidas avaliadas permitem caracterizar em grande detalhe o comportamento de um multiplexador durante os períodos de congestionamento e, como consequência, ajudam a entender melhor seu comportamento e a projetá-los de forma mais eficaz.

Na tese foi desenvolvida, também, uma ferramenta para análise de desempenho, constituída pelas expressões analíticas das medidas de desempenho calculadas.

Em segundo lugar, foi proposta uma metodologia geral para emular diferentes algoritmos de escalonamento, denominada Meta-algoritmo LDS. Também foi realizada uma análise de desempenho matemática do LDS. As medidas de desempenho avaliadas foram: o número médio de pares de células de voz presentes no buffer de voz, o número médio de células de dados armazenadas no buffer de dados, o retardo médio experimentado por cada tipo de célula e a probabilidade de perda de células de voz e de dados. A avaliação dessas medidas é relevante porque permite estabelecer se o multiplexador pode ou não cumprir com a QoS que cada tipo de tráfego requisita.

Usando os resultados desta avaliação de desempenho, foi gerada uma ferramenta (geral) que permite, mediante o uso dos parâmetros adequados, obter as medidas de desempenho para qualquer algoritmo de escalonamento proposto.

Em geral, para qualquer algoritmo que se queira propor, o trabalho consiste em: 1) Propor o algoritmo; 2) Propor e resolver um modelo, que represente o algoritmo, seja na forma analítica ou por simulação; 3) Usando o ponto 2, analisar o algoritmo proposto sob diferentes condições de operação e, 4) Obter conclusões práticas. Os pontos 2 e 3 retardam o ciclo de análise e aumentam a complexidade da mesma. Portanto, utilizar esta ferramenta de LDS resulta de grande utilidade no desenvolvimento de novas propostas na área de escalonamento, já que permitirá avaliar facilmente os novos algoritmos, pois para obter as medidas de desempenho desejadas, é suficiente usar os parâmetros de entrada apropriados. Desse modo é evitado desenvolver novas análises matemáticas e ter que escrever novos programas, o que implica uma economia de esforço e uma simplificação nesta área.

Otras das vantagens de utilizar LDS é a possibilidade de comparar os diferentes algoritmos propostos na literatura e escolher aquele que seja o mais adequado para a rede que se deseje implementar.

Por último, foi proposto e avaliado analiticamente um novo algoritmo de recuperação de erros para a transmissão multiponto confiável, denominado ACMBAR (Algoritmo de Confiabilidade Multiponto baseado na Árvore de Recuperação). Esse algoritmo tem alguns atributos que o fazem competitivo em relação aos algoritmos existentes. Em particular, a recuperação de erros é feita localmente. Isto é, os únicos membros do grupo que participam da recuperação do pacote perdido são os que se encontram na proximidade onde o erro foi produzido.

Uma avaliação matemática do algoritmo foi realizada, para diferentes topologias e tamanhos de grupo de receptores, de onde concluiu-se que o tamanho da implosão é próximo do ideal e que a recuperação possui uma baixa latência. Também foi demonstrado, através da avaliação analítica, que o algoritmo tem uma alta escalabilidade.

Esta tese não teve a pretensão de analisar genericamente os diversos algoritmos de controle e gerência de QoS. Se limitou aos problemas interrelacionados de escalonamento e multiplexação de pacotes, especificamente pacotes de voz, e aos problemas de QoS, mais especificamente confiabilidade, em comunicações textuais de dados com a QoS de melhor esforço.

### **VI.3 Trabalhos Futuros**

Entre os trabalhos futuros propostos, em primeiro lugar pode ser mencionados a ampliação da análise de multiplexadores para os distintos modelos do processo de chegada de voz: como por exemplo MMPP, Fluidos.

Para o caso do Meta - Algoritmo LDS um dos trabalhos futuros é generalizar as equações para o caso de  $N$  tráfegos distintos, assim como ampliar a análise para incluir outros tipos de controle de congestionamento.

Outro dos trabalhos que podem ser feitos é adequar o modelo do Meta-Algoritmo LDS e sua análise matemática para resolver outros problemas, tais como CAC (que corresponde ao caso de uma fase) e Policing (Leaky Bucket).

No caso de ACMBAR um dos trabalhos a realizar é a análise de desempenho de topologia aleatórias.

## Referências Bibliográficas

- [AAL297] Recommendation Project ITU-T I.363.2. “*Specification about adaptation layer ATM type 2*”. Technical Report CCITT, April 1997.
- [AAL299] ATM Forum, “*ATM Trunking using AAL2 Narrowband Services*”, Feb. 1999, <ftp://ftp.atmforum.com/pub/approved-specs/af-vtoa-0113.000.pdf>
- [AIE93] R. Aiello, E. Pagani and G.P. Rossi, “*Design of a Reliable Multicast Protocol*”, *INFOCOM'93*, March 1993.
- [AMS92] S. Amstron, A. Freier and K. Marzullo, “*Multicast Transport Protocol*”, Internet RFC1301, Feb.1992.
- [BAR01] M. Barría, R.Vallejos, and L.F.G.Soares, “*A Reliable Multicast Algorithm Based on Recovery Tree (RMART)*”, 9<sup>th</sup> IFIP Conference on Performance and Evaluation of ATM & IP Networks, Budapest, Hungary, June 2001.
- [BAR00b] M. Barría, R.Vallejos, and L.F.G.Soares “*Algoritmo para la Recuperación de Errores de una Transmisión Multipunto*”, Proceedings CLEI 2000, Sept. 2000, México.
- [BAR00a] M. Barría, R.Vallejos, and L.F.G.Soares, “*LDS: A Load Dependent Scheduling Algorithm*”. Proceedings of IFIP ATM & IP 2000 Workshop , Ilkley, UK, July 2000.
- [BAR99] M. Barría, R.Vallejos, “*Mecanismos de transmisión multipunto confiable: una conceptualización*”, Proceedings CLEI'99, Asunción, Paraguay, September 1999.

- [BAR98] M. Barría, R.Vallejos, and L.F.G.Soares, "*Transient Analysis of the Behaviour of an ATM Multiplexer During Congestion Periods*", IFIP ATM'98, Ilkley, UK, July 1998.
- [BAR96a] M. Barría, R.Vallejos, and L.F.G.Soares, "*Algoritmo Dependiente de la Carga para Administración de Ancho de Banda en Redes ATM*". VIII Seminario de Telecomunicaciones, Valdivia, Chile Nov. 1996.
- [BAR96b] M. Barría., R. Vallejos and L. F. G. Soares Marta Barría M., Reinaldo Vallejos C. and Luiz Fernando G. Soares. "Performance Analysis of a Voice Multiplexer with Bit-Dropping", *Investigación Operativa*, pp 169-198 Asociación Latino-Ibero-Americana de Investigación Operativa, ALIO", Dezembro, 1996.
- [BAR95b] M. Barría., R. Vallejos and L. F. G. Soares. "*Performance Analysis of an ATM Voice Multiplexer in Congestion*", XV Congreso Internacional de la Sociedad Chilena de Computación, Arica, Nov. 1995, pp 59--70.
- [BAR95a] M. Barría., R. Vallejos and L. F. G. Soares. "*Análise de Desempenho de um Multiplexador de Células com descarte de bits*", 13 Simpósio Brasileiro de Redes de Computadores, SBRC95. Belo Horizonte, Minas Gerais, Maio 1995, pp.83--101. (prêmio ao melhor artigo).
- [BEN96] J. Bennet and H. Zhang, "*WF<sup>2</sup>Q: Worst-Case Fair weighted Queueing*", Proceedings of IEEE INFOCOM '96, San Francisco, March 1996. pp.120-128.
- [BEN97] J. Bennet and H. Zhang, "*Hierarchical Packet Fair Queueing Algorithm*", IEEE/ACM Transactions on Networking, Vol.5, No.5 , October 1997, 675-689.
- [BER87] D.Bertsekas and R. Gallager, "*Data Networks*", Prentice Hall Inc., 1987.
- [BAL94] T. Ballardie, "*Core Based Tree (CBT) Multicast: Architectural Overview and Specification*" Internet Draft RFC, July 1994.

- [BIL97] T. Bilhartz, J.B. Cain, D. Fieg and S.G. Batsell, "*Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols*", IEEE JSAC, 15, April 1997.
- [BLA83] R. Blahut, "*Theory and Practice of Error Control Codes*", Addison Wesley, 1983
- [BOW87] D.O.Bowker and C.A.Dvorak. "*Speech transmission quality of wideband packet technology*". In IEEE GLOBECOM'87, Nov. 1987, pp. 1887-1889
- [BRA68] P. T. Brady. "*A statistical analysis of on-off patterns in 16 conversations*". Bell Syst. Tech. Journal, vol 47, no.1., Jan. 1968. pp 73-91
- [BOR94] C. Borman et al. "*MTP-2: Towards Achieving the S.E.R.O. Properties for Multicast Transport*", Proceedings Int. Conf. Comp. Commun. And Networks'94, Sept.,1994
- [CCI90] CCITT. Recommendation G.726-40, 32, 24, 16 kbits/s Adaptive Differential Pulse code Modulation (ADPCM), Dec. 1990.
- [CCI90A] CCITT. Recommendation I.121. "*Broadband Aspects of B-ISDN*", 1990
- [CCI90B] CCITT. Recommendation I.211. "*B-ISDN Service Aspects*", 1990.
- [CCI92] CCITT. Recommendation G.728- *Coding of Speech at 16 kbits/s using low-delay code excited linear prediction*, Sept. 1992.
- [CES97] ATM Forum Technical Committee, "*Circuit emulation Service Interoperability Specification Version 2.0*", afvtoa-0087.000, Technical Report, ATM Forum Jan. 1997, <ftp://ftp.atmforum.com/pub/approved-specs/af-vtoa-0078.000.pdf>
- [CID95] I.Cidon, A. Khamisy, L. Georgiadis and R.Guerin. "*Optimal buffer sharing*". Proceedings of IEEE INFOCOM'95, Boston, Vol. 1, April 1995. pp. 24-31.

- [COO96] J.R. Cooperstock and S. Kotsosalto, "*Why Use a Fishing Line when you have net? An Adaptive Multicast Data Distribution Protocol*", Proceedings. USENIX'96, 1996.
- [CHA84] J.M. Chang and N.F. Maxemchuk, "*Reliable Broadcast Protocol*", Journal ACM Transactions on Computer Systems, Vol. 2, Num. 3 Aug. 1984, pp. 251-273.
- [CHO98] A.K. Chouhury and E.L. Hahne, "*Dynamic Queue Length Threshold in a Shared Memory ATM Switch*", IEEE/ACM Transactions on Networking Vol. 6, No. 2, April 1998 pp. 130-140.
- [CHE98] K. Cheon and S. Panwar, "*On the performance of ATM-UBR with Early Selective Packet Discard*", Proceedings of IEEE ICC, 1998. pp.221-227
- [CHI98] H. Chiou and Z. Tsai, "*Performance of ATM Switches with Age Priority Packet Discarding under ON-OFF sources*", Proceedings IEEE INFOCOM, 1998, pp.931-938.
- [COX97] R.V. Cox, "Three New Speech Coders from the ITU Cover a range of Applications", *IEEE Communications Magazine*, September 1997, pp.40-47.
- [DAI86] J.N. Daile and J.D. Langford. Models for analysis of packet voice communications systems, *IEEE Journal on Selected Areas in Communications*, SAC-4, Sept.1986.
- [DEM89] A. Demers, S. Keshav and S. Shenker, "Hierarchical Packet Fair Queueing Algorithms", *ACM SIGCOMM Computer Communication Review*, vol. 19, No. 4, Sept. 1986, pp. 1-12.
- [DRA93] S. Dravida and V. Saksena, "Analysis and Engineering of a Voice/Data Packet Multiplexer", *IEEE Transactions on Communications*, Vol 41, No. 11, Nov. 1993, pp. 1656-1667.

- [DIOT97] C. Diot, W.Dabbous and J.Crowcroft, "Multipoint Communication: A survey Protocols, Functions and Mechanism", *IEEE Journal on Selected Areas in Communications* Vol. 15, No. 3, April, 1997.
- [DEC96] B.DeCleene, "*Delay Characteristics of generic Reliable Multicast Protocols*", Technical Report, TR-08150-3, Sept. 1996.
- [DBC97] ATM Forum, "*Specifications of (DBCES) Dynamic Bandwidth Utilization-In 64kps Time Slot Trunking Over ATM- Using CES*", July 1997,  
<ftp://ftp.atmforum.com/pub/approved-specs/af-vtoa-0085.000.pdf>
- [ERI94] H. Ericksson, "Mbone. The Multicast Backbone". *Communications of the ACM*, Vol. 37, August 1994.
- [FER90] D. Ferrari and D. Verma, "A Scheme for Real-Time Channel Establishment in wide-Area-Networks", *IEEE Journal on Selected Areas in Communications*, vol. 8, No.3, April 1990, pp. 368-379.
- [FER99] P. Ferguson and G. Huston "*Quality of Service; Delivering QoS on the Internet and Corporate Networks*", Wiley Computer publishing, John Wiley & Sons, Inc., 1998.
- [FLO93] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance" *IEEE/ACM Transactions on Networking*, Vol.1, No.4, 1993, pp 397-413.
- [FLO95b] S. Floyd and A. Romanov, "Dynamics of TCP Traffic over ATM Networks" *IEEE Journal selected Areas in Communications*, vol.13, No. 4, 1995, pp 633-641.
- [FLO97] S. Floyd, V. Jacobson, C-G. Liu, S. McCanne and L. Zhang, "A Reliable Multicast Framework for Lightweight Sessions and Application- Level Framing",

*IEEE/ACM Transactions on Networking*, 1997 . An earlier version of this paper appeared in *Proc. ACM SIGCOMM'95*, Oct. 1995, pp 342-345

- [GOL90] S. Golestani, "A Stop-and-Go Queueing Framework for Congestion Management", *ACM SIGCOMM Computer Communication Review*, vol. 20, No. 4, , Sept. 1990, pp. 8-18
- [GOL94] S. Golestani, "A Self-Clocked fair Queueing Scheme for Broadband Applications", *Proceedings of IEEE INFOCOM '94*, , Toronto, June 1994, pp. 636-646.
- [GOY97] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, and S. Kim. "UBR+: Improving performance of TCP over ATM-UBR service", *Proceedings of IEEE ICC'97*, 1997, pp. 1042-1048
- [GRO97] M. Grossglauser, "Optimal Deterministic Timeouts for Reliable Scalable Multicast", *IEEE Journal on Selected Areas in Communications*, Vol. 15 No. 3, April 1997
- [HUI96] C.Huitema, " The case for packet level FEC ", Proc. of IFIP 5th. Int. Workshop on Protocols for High Speed Networks (PfhSN'96), INRIA. Sophia Antipolis, France, Oct. 1996, Chapman & Hall.
- [HOL95] H.W. Holbrook, S.K.Singhal and D.R. Cheriton, Log-Based Receiver Reliable Multicast for Distributed Interactive Simulation ", Proc. ACM SIGCOMM'95, Oct. 1995, pp. 328-341
- [ITU96] ITU-T. Recommendation G.729- *Coding of Speech at 8 kbits/s using Conjugate-Structure Algebraic-Code-Excited-Linear-Prediction*, March 1996.
- [ITU97] ITU-T. Recommendation I.363.2 *B-ISDN ATM Adaptation Layer Specification: Type 2 AAL*, September 1997.

- [KAL90] C. Kalamane, H. Kanakia and S. Keshav, "Rate Controlled Servers for Very High Speed Networks", *Proceedings of IEEE Globecom'90*, San Diego, 300.3.1-300.3.9, December 1990.
- [KAR81] S. Karlin and H. Taylor. *A Second Course in Stochastic Processes*. Academic Press, Inc. 1981
- [KAS97] S. Kasera, J. Kurose and D. Towsley, "*Scalable reliable Multicast using Multiple Multicast Groups*", Proc. of ACM Sigmetrics, June 1997.
- [KAS98] S. Kasera, J. Kurose and D. Towsley, "*A Comparison of Server-Based and Receiver-Based Local Recovery Approaches for Scalable Reliable Multicast*", Proc. INFOCOM'98, April 1998, San Francisco, USA.
- [LAB99] M. Labrador and S. Banerjee. "Packet Dropping Policies for ATM and IP Networks", *IEEE Communications Surveys*. Vol. 2, No. 3, 3<sup>rd</sup>. Quarter 1999.  
<http://www.comsoc.org/pubs/surveys>.
- [LAB99b] M. Labrador and S. Banerjee "Enhancing application Throughput by Selective Packet Dropping", Proc. of IEEE ICC, 1999, pp. 1217-1222
- [LEE93] D. Lee and B. Sengupta, "Queueing Analysis of a Threshold Based Priority Scheme for ATM Networks", *IEEE/ACM Transactions on Networking*, Vol 1 N<sup>o</sup> 6, Diciembre 1993.
- [LIN96] J.C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol", Proceedings INFOCOM'96, Mar. 1996
- [LUC96] D. DeLucia and K. Obraczka, "Multicast Feedback Suppression Using Representatives", Tech. Report USC TR97-638, University of Southern California, June 1996.

- [MAC96] J. Macker, W. Dang "The Multicast Dissemination Protocol (MDP) Framework", Internet draft, Nov. 1996
- [MEY80] J.F. Meyer. "On evaluating the performability of degradable computing systems". *IEEE Transactions on Computers*, C-29(8); pp. 720-731,1980.
- [MEY92] J.F. Meyer, "Performability: A Retrospective and Some Pointers to the future", performance evaluation, Vol 14, No.3-4, pp. 139-156, 1992.
- [MIL97] K. Miller et al., "Starburst Multicast File Transfer Protocol (MTFP) Specification ", Internet Draft, IETF, draft-miller-mtftp-spec-02.txt, Jan.1997
- [MIL98] C. K. Miller, "Multicast Networking and Applications", Addison Wesley Logman, Inc., 1999.
- [MON96] T. Montgomery, J. Callahan and B. Whetten, "Specification and Design of a Fault Recovery Model for Reliable Multicast Protocol" Technical Report NASA-IVV-96-009, NASA/West Virginia University Software IV& V Facility, April 1996
- [MUK92] A. Mukherjee, A. landweber and T. Faber, "Dynamic time Windows and generalized virtual Clock: combined Closed-Loop/Open-Loop congestion Control", *Proceedings of IEEE INFOCOM'92*, Florence, Italy, 1992.
- [MBO] <http://www.cs.columbia.edu/~hgs/internet /mbone-faq.html>
- [NEW99] Newbridge networks Corporation, "An introduction to Voice over Packet-based Networks", 1999. <http://www.newbridge.com/technology/voice/index.html>
- [NON96] J. Nonnenmacher and E. Biersack, "Reliable Multicast: Where to use FEC", *Proc.5th. Workshop on Protocols for High Speed Networks*, editors W.Dabbous and C.Diot, pp. 134-148, Oct,1996, Sophia-Antipolis, France.

- [NON97A] J. Nonnenmacher, E. Biersack and D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmissions", *IEEE/ACM Transactions on Networking*, 1997; Also ACM SIGCOMM'97, June 1997.
- [NON98A] J. Nonnenmacher and E. Biersack, "Scalable Feedback for Large Groups", *IEEE/ACM Transactions on Networking*, 1998.
- [NON98B] J. Nonnenmacher and E. Biersack, "Optimal Multicast Feedback", *Proceedings INFOCOM'98*, March 1998, San Francisco, USA.
- [NON97B] J. Nonnenmacher and E. Biersack, "Scalable Feedback for Satellite Broadcast", *Proc. ACM/IEEE MobiCom'97* (Workshop on Satellite-based Information Services: WOSBIS'97), 1997.
- [NON98C] J. Nonnenmacher, M. Lacher, M. Jung, E. Biersack and G. Carle, "How bad is reliable multicast without local recovery?", *Proceedings IEEE INFOCOM'98*, March 1998.
- [ONV94] R. Onvural, "*Asynchronous Transfer Mode Networks Issues*", Artech House, 1994.
- [PAR93] A. K. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case", *IEEE/ACM Transactions on Networking*, Vol.1, No.3, pp. 344-357, June 1993.
- [PAR94] A. K. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case", *IEEE/ACM Transactions on Networking*, Vol.2, No.2, pp. 137-150, April 1994.
- [PET89] D. Petr, L. Da Silva, and V. Frost. Priority discarding of speech in integrated packet networks. *IEEE J. Select Areas in Communications*, SAC-7; 644-656, June 1989.

- [PIN94] S. Pingali, D. Towsley and J. Kurose, " A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", *Proc. ACM Sigmetrics Conference*, May 1994.
- [PAU97] S. Paul, K.Sabnani, J. Lin and S. Bhattacharyya, "Reliable Multicast Protocol (RMTP)", *IEEE JSAC Special Issue on Network support for Multipoint Communications*.
- [RAM96] S. Ramesh, C. Rosemberg, and A. Kumar. "Revenue maximization in ATM Networks using the CLP Capability and Buffer Priority Management", *IEEE/ACM transactions on Networking*, Vol.4, No. 6, pp. 941—950, December 1996.
- [RAM98] S. Raman, S. McCanne, "Asymptotic Behavior of Global Recovery in SRM", *Proc.SIGMETRICS'98*, June1998, USA.
- [RIZ97] L. Rizzo and L. Viciano, "A Reliable Multicast Data Distribution Protocol Based on Software FEC Techniques", *Proc High Perf. Communications Systems Conference*, 1997.
- [ROB94] W. Roberts, "Virtual Spacing for Flexible Traffic Control", *International Journal of Communications Systems*, Vol. 7, 1994.
- [ROS93] S. M. Ross. *Introduction to Probability Models*. Academic Press. 5th. edition. 1993
- [SAB96] B. Sabata, M.J. Brown and B.A. Denny, "Transport protocol for Reliable Multicast: TRM ", *Proc. IASTED International Conference Networks*, Jan. 1996, pp.143-156.
- [SCH96] M. Schwartz, "*Broadband Integrated Networks*", Prentice Hall PTR, 1996.
- [SHR96] Shreedar M, Varghese G, "Efficient Fair Queueing Using Deficit Round Robin", *IEEE/ACM Transactions on Networking*, Vol. 4, pp.375-385, June 1996.

- [SOA95] L.F.G. Soares and G. Lemos and S. Colcher. *Redes de Computadores. Das LANs, MANs e WANs às Redes ATM*. Editora Campus, Rio de Janeiro. 2 edition., 1995.
- [SPE99] T. Speakman, N. Bhaskar, R. Edmonstone, D. Farinacci, S. Lin, A. Tweedly, L. Vicisano, and J. Gemmell, “*PGM Reliable Transport protocol Specification*”, Internet Draft <draft-speakman-pgm-spec-03.txt>, June 1999
- [SRI86] K. Sriram and W.Whitt. Characterizing superposition arrival process in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, SAC-4;833-846, Sept 1986.
- [SRI91] K.Sriram, D.Scott McKinney and M.Hashen Sheriff. Voice packetization and compression in a broadband ATM networks. *IEEE Journal on Selected Areas in Communications*, SAC-9, April 1991.
- [SRI93] Sriram K., "Methologies for bandwidth allocation, transmission scheduling, and congestion avoidance in broadband ATM networks", Computer Networks and ISDN Systems 26 (1993) North-Holland.
- [STR92] W.Strayer, B.Dempsey and A.C. Weaver, " XTP: The Xpress Transfer Protocol ", Addison-Wesley, 1992.
- [TM499]. ATM Forum, Traffic Management Specification Version 4.1. AF-TM-0121.000, March 1999.
- [TOW97] D. Towsley, J.Kurose and S. Pingali, " A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols ", *IEEE Journal on Selected Areas in Communications*, April 1997 .
- [TUR96] J. Turner “Maintaining High Throughput During Overload In ATM Switches” *Proceedings of IEEE INFOCOM*, pp 287-295, 1996.

- [VAL93] R. Vallejos C.", *Performability Models and Its Applications to Computer Systems* PhD thesis, Federal University of Rio de Janeiro, Brazil, January 1993.
- [VER91] D. Verma, H. Zhang, D. Ferrari, "Delay Jitter Control for Real-Time Communication in Packet switching Network", *Proceedings of Tricomm'91*, USA, pp.31-46, April 1991.
- [VEC95] de Veciana G, Kesidis G and Walrand J. , "Resource Management in Wide Area networks Using Effective Bandwidths. *IEEE Journal on Selected Areas in Communications*, Vol. 13 No.6, August 1995.
- [WHE95] B. Whetten, T. Montgomery and S. Kaplan, "A High Performance Totally Ordered Multicast Protocol. *Theory and Practice in Distributed Systems*", Springer Verlag, 1995.
- [WRE97] Wrege, D., Liebeherr, J. "A Near-Optimal Packet Scheduler for QoS Networks", *Proceedings of IEEE Infocom'97*, Kobe, Japan (1997), pp.576-583.
- [WRI96] D. Wright. Voice over ATM: An evaluation of implementation alternatives. *IEEE Communications Magazine*, pp. 72-80, May 1996.
- [XU97] X.R. Xu et al., "Resilient Multicast support for Continuous-Media Applications", *Proc. NOSSDAV'97*, 1997
- [XTP] XTP Web, <http://www.ca.sandia.gov/xtp/xtp.html>
- [YIN93] N. Yin and M. Hluchyj. "Implication of Dropping Packets from the Front of a Queue", *IEEE Transaction on Communications*, 37; pp.846-851, June 1993.
- [YAN95] C-O. Yang and A. Reddy, "A taxonomy for Congestion Control Algorithms in Packet Switching Networks", *IEEE Network Magazine*, Vol. 9, No. 5, July/August 1995.

- [YAM97] M. Yamamoto, J. Kurose, D. Towsley and H. Ikeda, "A Delay Analysis of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", *Proc. IEEE INFOCOM'97*, April 1997, Kobe, Japan.
- [YAV95] R. Yavatkar, J. Griffioen and M. Sudan, "A Reliable Dissemination Protocol for Interactive Collaborative Applications ", *Proc. ACM Multimedia'95*, 1995.
- [ZHA90] L. Zhang, "Virtual clock: A New Traffic Control Algorithm for Packet Switching Networks", *ACM SIGCOMM Computer Communications Reviews*, Vol. 20, No. 4, pag. 19-29, September 1990.

## PUBLICAÇÕES GERADAS

### PERIÓDICOS

- Marta Barría M., Reinaldo Vallejos C. and Luiz Fernando G. Soares. "Performance Analysis of a Voice Multiplexer with Bit-Dropping", *Investigación Operativa*, pp 169-198 Asociación Latino-Ibero-Americana de Investigación Operativa, ALIO", Dezembro, 1996.

### CONGRESSOS

- M. Barría, R.Vallejos, and L.F.G.Soares, "A *Reliable Multicast Algorithm Based on Recovery Tree (RMART)*", 9<sup>th</sup> IFIP Conference on Performance and Evaluation of ATM & IP Networks, Budapest, Hungary, June 2001.
- Marta Barría, Reinaldo Vallejos, y L.F.G.Soares " *Algoritmo para la Recuperación de Errores de una Transmisión Multipunto*", CLEI 2000, Sept. 2000. Proceedings em CD.
- Marta Barría, Reinaldo Vallejos, y L.F.G.Soares, *LDS: A Load Dependent Scheduling Algorithm*. IFIP ATM & IP 2000 Workshop , Participants Proceedings: Research Works in Progress. Pp 06/01-06/10. Ilkley, Inglaterra, July 2000.
- Marta Barría y Reinaldo Vallejos " *Mecanismos de transmisión multipunto confiable: una conceptualización*", CLEI'99, Asunción, Paraguay, Setembro 1999.
- Marta Barría, Reinaldo Vallejos , L.F.G.Soares, " *Transient Analysis of the Behaviour of an ATM Multiplexer During Congestion Periods*", IFIP ATM'98. In Participants Proceedings: Reseach Papers.pp. 9/01- 9/10, Ilkley, Inglaterra, July 1998
- Marta Barría M., Reinaldo Vallejos C. and Luiz Fernando G. Soares, " *Algoritmo Dependiente de la Carga para Administración de Ancho de Banda en Redes ATM*". VIII Seminario de Telecomunicaciones, Valdivia, Nov. 1996. pp.138-145.

- Marta Barría M., Reinaldo Vallejos C. y Luis Fernando G. Soares. *"Performance Analysis of an ATM Voice Multiplexer in Congestion"*, XV Congreso Internacional de la Sociedad Chilena de Computación, Arica 1995, pp 59--70.
- Marta Barría M., Reinaldo Vallejos C. e Luis Fernando G. Soares. *"Análise de Desempenho de um Multiplexador de Células com descarte de bits"*, 13 Simpósio Brasileiro de Redes de Computadores, SBRC95. Belo Horizonte, Minas Gerais, Maio 1995, pp.83--101. (prêmio ao melhor artigo).

## Apêndice A

Para calcular o valor médio de um período ocupado,  $B$ , denota-se por  $X_n$ ,  $n \in Z$ , a variável igual to 0 ( respectivamente 1) se o sistema está vazio ( respectivamente ocupado) no instante de saída  $D_n$  e, seja  $T_j$ ,  $j \in Z$ , a sequencia de indices  $n$  tais que  $X_n = 0$ ; então:

$$\begin{aligned} E(T_1 | T_0 = 0) \cdot P(X_0 = 0) &= E(T_1; T_0 = 0) = \\ \sum_{k \geq 1} P(T_1 \geq k; T_0 = 0) &= \sum_{k \geq 1} P(X_0 = 0, X_1 = \dots = X_{k-1} = 1) \end{aligned} \quad (\text{a.1})$$

por estacionariedade do processo  $(X_n)$ , pode ser escrito:

$$\sum_{k \geq 1} P(X_0 = 0, X_1 = \dots = X_{k-1} = 1) = \sum_{k \geq 1} P(X_{-k} = 0, X_{-k+1} = \dots = X_{-1} = 1)$$

Seja  $v$  o primeiro instante no qual o processo  $(X_n)$  atinge o valor 0 depois de um tempo  $n=0$ , então:

$$\sum_{k \geq 1} P(X_{-k} = 0, X_{-k+1} = \dots = X_{-1} = 1) = \sum_{k \geq 1} P(v = k) = 1 \quad (\text{a.2})$$

De (a.1) e (a.2), deriva-se que :

$$E(T_1 | T_0) = \frac{1}{P(X_0 = 0)} = \frac{1}{\pi_0}$$

Descondicionando em relação à distribuição do tempo de serviço, deriva-se pela identidade de Wald [9] que:

$$B = (1/\pi_0)(1/\mu)$$