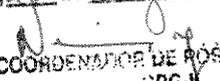


Este exemplar corresponde à redação final da
Tese/Dissertação devidamente corrigida e defendida
por: Hana Karina Salles
Rubinsztejn
e aprovada pela Banca Examinadora.
Campinas, 28 de Agosto de 2009

COORDENADOR DE PÓS-GRADUAÇÃO
PPG-II

Dimensionamento de Sistemas de
Vídeo Interativo

Hana Karina Salles Rubinsztejn

Dissertação de Mestrado

Dimensionamento de Sistemas de Vídeo Interativo

Hana Karina Salles Rubinsztein

Julho de 2001

Banca Examinadora:

- Prof. Dr. Nelson Luís Saldanha da Fonseca
Instituto de Computação, Unicamp (Orientador)
- Prof. Dr. Guido Lemos de Souza Filho
UFRN
- Prof. Dr. Célio Cardoso Guimarães
Instituto de Computação, Unicamp
- Prof. Dr. Ricardo Dahab
Instituto de Computação, Unicamp (Suplente)

200124392

UNIDADE	30
N.º CHAMADA:	T/ UNICAMP
	R825d
V.	Ex.
TOMBO DC/	46824
PROC.	76-392161
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PRECOS	R\$11,00
DATA	3/1/01
N.º CPD	

CM00161217-2

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Rubinsztein, Hana Karina Salles

R825d Dimensionamento de Sistemas de Vídeo Interativo / Hana Karina Salles Rubinsztein. -- Campinas, [S.P. :s.n.], 2001.

Orientador : Prof. Dr. Nelson Luís Saldanha da Fonseca.

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1.Sistemas multimídia. 2.Transmissão de imagem. 3. Gravações de vídeo interativas. 4.Sistemas de computação interativos eletrônicos. I. Fonseca, Nelson Luís Saldanha da. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Dimensionamento de Sistemas de Vídeo Interativo

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Hana Karina Salles Rubinsztein e aprovada
pela Banca Examinadora.

Campinas, 12 de julho de 2001.



Prof. Dr. Nelson Luís Saldanha da Fonseca

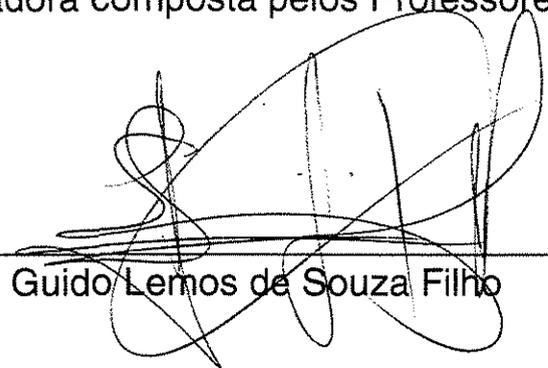
Instituto de Computação, Unicamp

(Orientador)

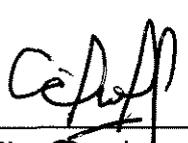
Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

TERMO DE APROVAÇÃO

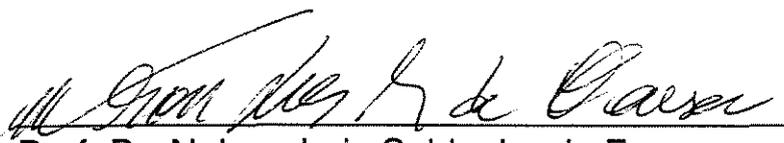
Tese defendida e aprovada em 12 de julho de 2001, pela Banca Examinadora composta pelos Professores Doutores:



Prof. Dr. Guido Lemos de Souza Filho
UFRN



Prof. Dr. Célio Cardoso Guimarães
IC - UNICAMP



Prof. Dr. Nelson Luis Saldanha da Fonseca
IC - UNICAMP



Aos meus pais e ao meu noivo

Agradecimentos

Ao meu orientador, pelas palavras de incentivo e por sua amizade. Agradeço especialmente a oportunidade de sua orientação neste trabalho.

Aos professores Edmundo Souza e Silva e Daniel Menacé, por sua atenção esclarecimentos prestados sobre os conceitos contidos em trabalhos de sua autoria, os quais foram utilizados como referências neste trabalho.

Aos amigos do IC, pelos momentos agradáveis de convivência dentro e fora da Unicamp, em especial aos conterrâneos Rogério A. Miranda e Danival T. Calegari, por suas valiosas sugestões no decorrer deste trabalho.

Ao meu noivo Fred, pela paciência, amor e também pelas palavras confortadoras nos momentos de maior dificuldade.

À minha família, por toda a confiança e apoio incondicionais, que em muito me ajudaram a concluir este trabalho.

À CAPES, pelo suporte financeiro deste trabalho, sem o qual sua realização não teria sido possível.

A Deus, que tornou possíveis todos os agradecimentos anteriores.

Resumo

Vídeo sob Demanda (VoD) é um serviço que possibilita a exibição de filmes para usuários geograficamente dispersos. Este serviço pode ser utilizado em diversas áreas, tais como entretenimento, bibliotecas digitais e ensino a distância, e através dele, um usuário pode escolher um vídeo dentre uma coleção e ter o vídeo enviado através de uma rede de telecomunicações até sua casa. Um sistema de vídeo interativo deve permitir requisições de operações de VCR a qualquer momento. Estas aplicações requerem uma grande demanda por banda passante. Assim, para o oferecimento destes serviços em larga escala é necessário utilizar técnicas de redução desta demanda. Quando um usuário efetua uma operação de VCR, sua exibição se dessincroniza da exibição do seu grupo, sendo necessário um outro canal de vídeo para dar suporte ao fluxo dessincronizado.

Nesta dissertação são avaliados aspectos de desempenho em servidores de sistemas de VoD interativo, nos quais técnicas de compartilhamento de fluxo são empregadas como mecanismo para a redução da demanda de banda passante. Apresenta-se uma nova técnica de dimensionamento do número de canais necessários para se prover a interatividade. Além disso, são introduzidas técnicas para sistemas de VoD com *batching*, e para sistemas de VoD com *batching* e *piggybacking*.

Abstract

Video-on-demand (VoD) is a service that enables the exhibition of films for geographically dispersed users. It engenders several application areas such as on-demand movies, digital libraries and distance learning. A video on-demand system which allow VCR operations (true video-on- demand) should grant users' request to perform VCR operations at any time. To reduce the huge bandwidth demand of VoD, techniques based on multicast have been considered for the deployment of such services in large scale. When a user performs a VCR operation, his/her exhibition unsynchronizes with the exhibition of his/her multicast group. Therefore, another video channel is needed to support the unsynchronized stream.

This thesis introduces a novel dimensioning technique for channel allocation in interactive VoD systems in order to provide the desired Quality-of-Service. Techniques for VoD systems with batching and for VoD systems with both batching and piggybacking are presented.

Sumário

Agradecimentos	vi
Resumo	vii
Abstract	viii
1 Introdução	1
2 Conceitos de Sistemas de Vídeo sob Demanda	5
2.1 Arquitetura do Sistema de Vídeo sob Demanda	6
2.1.1 <i>Set Top Box</i>	6
2.1.2 Rede de Distribuição	7
2.1.3 Rede de Acesso	8
2.2 Aspectos de Servidores de Vídeo	10
2.2.1 Componentes de um Servidor	11
2.2.2 Hierarquia de Armazenamento	12
2.3 Interatividade em Sistemas de Vídeo sob Demanda	14
2.4 Técnicas de Redução da Demanda de Banda Passante	16
2.4.1 Replicação e <i>Caching</i>	16
2.4.2 <i>Bridging</i>	17
2.4.3 Compartilhamento de Fluxo	17
2.5 Distribuições Estatísticas Comuns em VoD	32

3	Conceitos em Redes de Filas e Fórmula de Erlang	35
3.1	Redes de Filas	35
3.1.1	<i>Product Form Networks</i>	36
3.1.2	Redes Fechadas	37
3.2	Fórmula de Erlang	40
4	Projeto de Sistemas Interativos de Vídeo	43
4.1	Trabalhos Anteriores	44
4.2	Sistemas de VoD Interativo	45
4.3	Avaliando a Demanda de Canais	46
4.3.1	Um Sistema com <i>Batching</i>	47
4.3.2	Um Sistema com <i>Batching</i> e <i>Piggybacking</i>	47
4.4	Precisão do Modelo Aproximado	53
4.5	Descrição Simulador	55
4.6	Resultados Numéricos	57
4.6.1	Sistema de VoD com <i>Batching</i>	58
4.6.2	Sistemas de VoD com <i>Batching</i> e <i>Piggybacking</i>	68
5	Conclusões	79
A	Tabelas de Símbolos	83
	Bibliografia	89

Lista de Tabelas

2.1	Operações de VCR	14
4.1	Tempo médio do cálculo da fórmula de Erlang, em microssegundos	55
A.1	Letras Gregas.	83
A.2	Letras Maiúsculas.	86
A.3	Letras Minúsculas.	87

Lista de Figuras

2.1	Arquitetura de um Sistema de VoD.	6
2.2	Divisão de canais em uma rede HFC.	9
2.3	Esquema de gerenciamento de memória.	12
2.4	Ilustração do funcionamento da política par-ímpar.	25
2.5	Ilustração do funcionamento da política mesclagem simples.	25
2.6	Ilustração do funcionamento da política gulosa.	26
2.7	Dimensionamento da janela ótima relativo à janela máxima em função do intervalo médio entre requisições.	28
2.8	Possíveis árvores de mesclagens para quatro fluxos.	29
2.9	Possível situação da política S^2 na qual o último fluxo resultante gerado não corresponde ao fluxo resultante da última janela ótima contida na janela W'_m	31
2.10	Probabilidade de escolha dos vídeos conforme a distribuição Zipf com parâmetro $\theta = 0,271$	33
3.1	Modelo de servidor central.	38
4.1	Aproximação do modelo de redes de filas fechadas para o modelo de filas aberto.	49
4.2	Comparação entre número de canais de contingência estimado e o número médio e máximo de canais obtidos via simulação.	54
4.3	Nº de Usuários Admitidos \times Capacidade do Servidor, taxa de 10 req/min.	59
4.4	Probabilidade de Abandono \times Capacidade do Servidor, taxa de 10 req/min.	60

4.5	Porcentagem de operações de VCR Rejeitadas × Capacidade do Servidor, taxa de 10 req/min.	61
4.6	N° de Usuários Admitidos × Capacidade do Servidor, taxa de 60 req/min.	62
4.7	Probabilidade de Abandono × Capacidade do Servidor, taxa de 60 req/min.	63
4.8	Porcentagem de operações de VCR Rejeitadas × Capacidade do Servidor, taxa de 60 req/min.	63
4.9	Capacidade do Servidor × Taxa de Chegada, para probabilidade de abandono inferior a 1%.	64
4.10	Capacidade do Servidor × Taxa de Chegada, para porcentagem de requisições de VCR rejeitadas inferior a 1%.	65
4.11	Capacidade do Servidor × Número de Vídeos Armazenados, para probabilidade de abandono inferior a 1%, à taxa de 60 req/min.	66
4.12	Capacidade do Servidor × Número de Vídeos Armazenados, para porcentagem de requisições de VCR rejeitadas inferior a 1%, à taxa de 60 req/min.	66
4.13	Capacidade do Servidor × Número de Vídeos Armazenados, para probabilidade de abandono inferior a 1%, à taxa de 10 req/min.	67
4.14	Capacidade do Servidor × Número de Vídeos Armazenados, para porcentagem de requisições de VCR rejeitadas inferior a 1%, à taxa de 10 req/min.	68
4.15	N° de Usuários Admitidos × Capacidade do Servidor, taxa de 10 req/min.	69
4.16	Probabilidade de Abandono × Capacidade do Servidor, taxa de 10 req/min.	70
4.17	Porcentagem de operações de VCR Rejeitadas × Capacidade do Servidor, taxa de 10 req/min.	71
4.18	N° de Usuários Admitidos × Capacidade do Servidor, taxa de 60 req/min.	72
4.19	Probabilidade de Abandono × Capacidade do Servidor, taxa de 60 req/min.	72
4.20	Porcentagem de operações de VCR Rejeitadas × Capacidade do Servidor, taxa de 60 req/min.	73
4.21	Capacidade do Servidor × Taxa de Chegada, para probabilidade de abandono inferior a 1%.	74
4.22	Capacidade do Servidor × Taxa de Chegada, para porcentagem de requisições de VCR rejeitadas inferior a 1%.	74

3	Conceitos em Redes de Filas e Fórmula de Erlang	35
3.1	Redes de Filas	35
3.1.1	<i>Product Form Networks</i>	36
3.1.2	Redes Fechadas	37
3.2	Fórmula de Erlang	40
4	Projeto de Sistemas Interativos de Vídeo	43
4.1	Trabalhos Anteriores	44
4.2	Sistemas de VoD Interativo	45
4.3	Avaliando a Demanda de Canais	46
4.3.1	Um Sistema com <i>Batching</i>	47
4.3.2	Um Sistema com <i>Batching</i> e <i>Piggybacking</i>	47
4.4	Precisão do Modelo Aproximado	53
4.5	Descrição Simulador	55
4.6	Resultados Numéricos	57
4.6.1	Sistema de VoD com <i>Batching</i>	58
4.6.2	Sistemas de VoD com <i>Batching</i> e <i>Piggybacking</i>	68
5	Conclusões	79
A	Tabelas de Símbolos	83
	Bibliografia	89

Capítulo 1

Introdução

Multimídia refere-se à combinação de várias formas de representação de informação, tais como áudio, gráficos, texto e vídeo, e à apresentação síncrona, manipulação e interação destas mídias. Se por um lado a rápida expansão do paradigma multimídia deve-se à capacidade crescente dos computadores pessoais (*desktops*), por outro, o oferecimento de serviços multimídia remotos em larga escala depende do efetivo dimensionamento das futuras redes de comunicação.

Uma das áreas mais promissoras para o oferecimento de serviços multimídia em redes é a *Loja de Locação de Vídeo Eletrônico (Electronic Video Rental Store)* que oferece o serviço de Vídeo sob Demanda (*Video on Demand — VoD*). Este serviço pode ser utilizado em diversas áreas, tais como entretenimento, bibliotecas digitais e ensino a distância, e através dele, um usuário pode escolher um vídeo dentre uma coleção e ter o vídeo enviado através de uma rede de telecomunicações até sua casa.

Um sistema de VoD pode ser implementado de modo a oferecer uma grande diversidade de serviços. Em um extremo há o serviço mais sofisticado, no qual o vídeo é iniciado instantaneamente e o usuário pode executar operações de VCR, tais como parada (PAUSE), retrocesso (REW) e avanço rápido (FF). No outro extremo, encontra-se o serviço econômico, no qual o usuário deve esperar o início programado da apresentação. Neste caso o usuário não dispõe de operações VCR [1]. No entantanto, um sistema de vídeo sob demanda interativo (*true video-on-demand*) deve permitir requisições de operações de

VCR a qualquer momento.

Para a manutenção de entrega contínua de um fluxo de vídeo, é necessário reservar recursos na rede e no servidor. A transmissão de vídeo requer grande demanda de banda passante, o que restringe o oferecimento de serviços de vídeo a uma grande população de usuários devido aos recursos finitos alocáveis para transmissão na subrede de comunicação, bem como a capacidade finita de entrada/saída dos servidores de vídeo (canais lógicos) e de processamento. Assim sendo, diversas técnicas vêm sendo propostas a fim de reduzir a demanda de banda passante, dentre elas, pode-se citar *Batching* e *Piggybacking*.

As técnicas de *Batching* e *Piggybacking* utilizam o compartilhamento de fluxos de vídeo como mecanismo para a redução da demanda de banda passante em servidores de vídeo. Em outras palavras, um servidor de VoD pode atender a um conjunto de usuários que deseja assistir a um determinado vídeo, utilizando uma quantidade de recursos inferior à requerida em caso de alocação de um fluxo para cada usuário. Este compartilhamento ocorre de forma transparente, sem que a exibição a um usuário em particular seja prejudicada.

Piggybacking é baseado no fato de que espectadores não percebem uma variação na taxa de exibição de até 5% da taxa nominal. Em um servidor de VoD com *piggybacking*, uma requisição para assistir a um filme é imediatamente aceita, se existir um canal disponível. Quando um novo canal é alocado e outra exibição do mesmo filme está em progresso, a taxa de exibição do filme em andamento é reduzida, enquanto que a taxa de exibição da requisição recém admitida é aumentada. Quando o fluxo de vídeo com maior taxa alcança o mais lento, os dois são mesclados e, conseqüentemente, um dos canais de vídeo é liberado.

Em um servidor de VoD com *batching*, requisições para exibição de vídeo não são admitidas assim que chegam. Elas são atrasadas para que várias requisições de um mesmo filme, dentro de um certo intervalo, sejam coletadas. Um único canal de vídeo é então alocado para todo o grupo (*batch*) de requisições (usuários). Se, por um lado, *batching* aumenta a vazão, por outro, usuários podem não estar dispostos a esperar por longos períodos de tempo e podem cancelar suas requisições (abandono). No entanto, sabe-se que um sistema com *batching* admite um maior número de usuários que um sistema com

piggybacking [2]. Além disso, adotar ambos, *batching* e *piggybacking*, aumenta o número de usuários admitidos no sistema.

Quando um usuário efetua uma operação de VCR, sua exibição se dessincroniza com a exibição do seu grupo. Desta forma, um outro canal de vídeo é necessário para dar suporte ao fluxo dessincronizado. É desejável que um canal esteja imediatamente disponível para admitir a requisição de operação de VCR, de modo que seja assegurada a continuidade da exibição. Além disso, este usuário precisa de um canal dedicado até o final da exibição, ou até se resincronizar com outro fluxo. Portanto, é de suma importância determinar o número de canais necessários para atender a demanda por banda passante das requisições de operações de VCR.

O objetivo desta dissertação é avaliar aspectos de desempenho em servidores de sistemas de VoD interativo, nos quais técnicas de compartilhamento de fluxo são empregadas como mecanismo para a redução da demanda de banda passante. Apresenta-se uma nova técnica de dimensionamento do número de canais necessários para se prover a Qualidade de Serviço (*Quality-of-Service* - QoS) desejada. Além disso, são introduzidas técnicas para sistemas de VoD com *batching*, e para sistemas de VoD com *batching* e *piggybacking*.

Esta dissertação está organizada como se segue. Aspectos conceituais de sistemas de VoD e técnicas de *batching* e *piggybacking* são apresentados no capítulo 2. No capítulo 3 são introduzidos conceitos sobre redes de filas. Um modelo de sistema interativo e seu dimensionamento é descrito no capítulo 4. No capítulo 5 conclusões e considerações finais são apresentadas.

Capítulo 2

Conceitos de Sistemas de Vídeo sob Demanda

Vídeo sob Demanda é um serviço que possibilita a exibição de filmes para usuários geograficamente dispersos. O usuário ou assinante pode selecionar em tempo real a transmissão de um vídeo dentre uma coleção, que pode conter filmes, enciclopédias eletrônicas, jogos interativos e vídeos educacionais. Além disso, o usuário tem controle sobre o vídeo através de operações de VCR.

Sistemas de Vídeo sob Demanda podem se tornar disponíveis comercialmente à medida em que se obtêm avanços nas diversas tecnologias envolvidas (dispositivos de armazenamento, técnicas de compressão e estruturas de suporte à transmissão digital).

Este capítulo aborda resumidamente diversos aspectos de um sistema de VoD. Inicialmente, apresenta-se a arquitetura de um sistema de VoD e seus componentes na seção 2.1. Aspectos relacionados aos servidores de vídeo são abordados na seção 2.2. Em seguida, na seção 2.3 são citados esquemas que viabilizam a interatividade entre usuário e sistema, através de operações de VCR. Técnicas de redução da demanda de banda passante são relacionadas na seção 2.4. Finalmente, na seção 2.5, comentários são feitos sobre as principais distribuições estatísticas utilizadas em sistemas de VoD.

2.1 Arquitetura do Sistema de Vídeo sob Demanda

Nesta seção descreve-se uma arquitetura de VoD, cujos principais elementos são mostrados na Figura 2.1. O sistema inclui servidor(es) de vídeo, uma rede de distribuição com funções de gerenciamento e subredes locais de acesso ao *backbone*, além do equipamento utilizado pelo usuário (*set top box* e controle remoto).

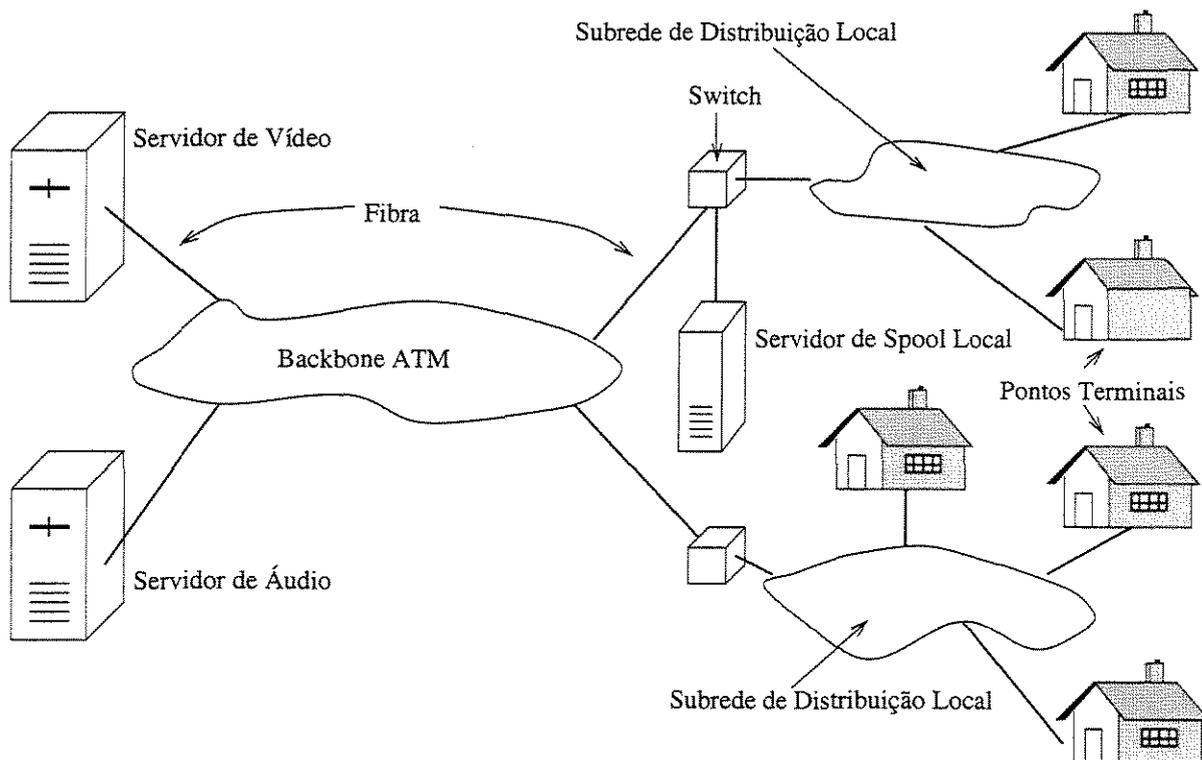


Figura 2.1: Arquitetura de um Sistema de VoD.

2.1.1 Set Top Box

Na interface da rede comunitária com o usuário encontra-se o *Set Top Box* ou STB. O STB recebe o fluxo de vídeo da rede e o apresenta para o usuário. Através dele o usuário se comunica com o sistema, selecionando o serviço ou vídeo e gerando possíveis operações interativas.

O STB conecta-se ao aparelho de TV ou vídeo cassete [3, 4] e, utilizando o controle

remoto, o usuário pode interagir com o sistema. O STB possui duas funções: *i*) enviar, receber e processar os sinais de controle e os comandos do usuário; e *ii*) receber, decodificar e apresentar os quadros de vídeo. Para tanto, é necessário uma unidade, chamada *transceiver*, que recebe os dados da rede e permite o envio de sinais de controle para o servidor de vídeo; e uma unidade de descompressão que é utilizada para decodificar a seleção realizada, geralmente codificada em MPEG-1 ou MPEG-2 [5, 6], e enviá-la ao dispositivo de exibição, geralmente a TV.

2.1.2 Rede de Distribuição

A rede de distribuição é o conjunto de *switches* e cabeamento existentes entre o servidor de vídeo e o equipamento do usuário, consistindo de um *backbone* interligado a uma rede de distribuição local. É necessário que o *backbone* possua grande largura de banda e suporte a transmissão de fluxos de dados isócronos.

Uma das opções mais cotadas é a utilização de *backbone* ATM. ATM ou *Asynchronous Transfer Mode* é o padrão para as Redes Digitais de Serviços Integrados de Faixa Larga (RDSI-FL)[7, 8, 9]. O ATM é uma tecnologia de multiplexação e comutação, de grande largura de banda e baixo retardo, que vem tornando-se disponível para redes públicas e privadas.

As unidades básicas de transferência de dados, em redes ATM, são chamadas de *células*. Elas possuem o tamanho de 53 bytes ou octetos e são transmitidas através de conexões com circuitos virtuais. O ATM busca maximizar a utilização da rede através da multiplexação estatística.

As redes ATM dispõem de mecanismos de controle de admissão de conexões (*Connection Admission Control* — CAC), e de mecanismos para o controle do tráfego das conexões estabelecidas (mecanismos de policiamento).

O CAC é responsável pela aceitação ou recusa de uma nova conexão mediante os parâmetros de “contrato” declarados pela fonte e dos recursos disponíveis. O mecanismo de policiamento é responsável por avaliar se as fontes geradoras de tráfego estão respeitando os parâmetros descritos no contrato de tráfego negociado durante a fase de estabelecimento da conexão, com o intuito de proteger os recursos da rede e manter a

QoS (*Quality of Service*) das conexões já existentes.

Através destes mecanismos, esta tecnologia é capaz de suportar e garantir diversas Qualidades de Serviço (QoS), satisfazendo os requisitos exigidos para o oferecimento de uma variedade de serviços de comunicação multimídia de alta velocidade.

2.1.3 Rede de Acesso

As redes locais de acesso compõem a parte final da estrutura de distribuição. Elas compreendem várias estruturas a partir das quais interliga-se o *backbone* ao equipamento do usuário. As soluções mais comuns existentes são ADSL, FTTC, FTTH e HFC [3, 5]. Estas tecnologias são apresentadas a seguir.

O **ADSL** (*Asymmetric Digital Subscriber Line*) propõe a utilização de par trançado para a transmissão de vídeo, de tal forma que se possa utilizar o cabeamento de telefonia já instalado. No entanto, esta estrutura não suporta a transmissão de fluxos de vídeo MPEG devido ao seu comprimento típico de 10Km. A solução ADSL é eliminar eco e outros ruídos através da tecnologia de processamento eletrônico de sinais. ADSL-1, padronizado pelo ANSI (*American National Standards Institute*), oferece um canal para transmissão de dados a 1,536 Mbps, um canal para sinais de controle com taxa de 16kbps, além de um canal analógico para telefonia que opera com frequência de 4kHz. ADSL-2 e ADSL-3 encontram-se em processo de padronização.

O **FTTC** (*Fiber To The Curb*) é uma tecnologia simétrica que utiliza uma estrutura híbrida, composta de fibra óptica e par trançado. O segmento de fibra óptica interliga o *backbone* até um dispositivo nas proximidades (*Optical Network Unit* — ONU) das instalações dos usuários. Deste ponto em diante, utilizam-se conexões ponto a ponto com o dispositivo do usuário, através de par trançado.

O **FTTH** (*Fiber To The Home*) utiliza fibra óptica desde o servidor de vídeo até os pontos terminais. Esta alternativa aparece como a solução definitiva pois oferece largura de banda suficiente para a transmissão de dados e sinais de controle, além de ser um meio com taxa de erros extremamente reduzida. Porém, é bastante cara e sua implantação não

deve ocorrer a curto prazo.

Diferentemente das alternativas anteriores, que empregam abordagem ponto a ponto, o HFC (*Hybrid Fiber Coax*) propõe a utilização de um cabo coaxial de banda larga compartilhado que interliga diversos usuários a um segmento de fibra óptica ligado ao *backbone*. Transmissões podem ocorrer na forma analógica ou digital, ou seja, possui suporte para a transmissão de TV a cabo e VoD. Entretanto, em virtude do acesso compartilhado ao meio de transmissão, esquemas de acesso múltiplo como TDMA (*Time Division Multiple Access*) ou FDMA (*Frequency Division Multiple Access*) precisam ser utilizados para os sinais de controle.

Uma rede HFC é capaz de suportar 125 canais de 6 MHz, 75 dos quais são usados para transmissões de TV e os canais restantes para tráfego multimídia. A Figura 2.2 mostra como é feita a alocação destes canais. A faixa 5 a 42 MHz é usada para transmissão de retorno. Esta faixa é dividida em canais que variam de algumas centenas de KHz a 1 MHz. Usando a técnica de modulação QPSK (*Quaternary Phase Shift Keying*) pode-se obter taxas de 2 a 3 Mb/s nestes canais. Para manter a compatibilidade, a faixa de 54 a 450 MHz continua sendo usada para transmissões analógicas de TV. O restante da faixa acima de 450 MHz é usado para transmissões digitais de tráfego multimídia. Os 50 canais multimídia são modulados usando QAM (*Quadrature Amplitude Modulation*) e oferecem cerca de 30 Mbps por canal. Supondo que cada canal atenda a 250 residências, tem-se um canal de 6 Mbps dedicado a cada usuário, permitindo interatividade completa com o sistema. A comunicação do STB com a rede HFC é feita através do uso de *cable modems*. Estes dispositivos são similares aos *modems* de voz e fazem a recepção e o envio dos sinais digitais no cabo coaxial de banda larga.

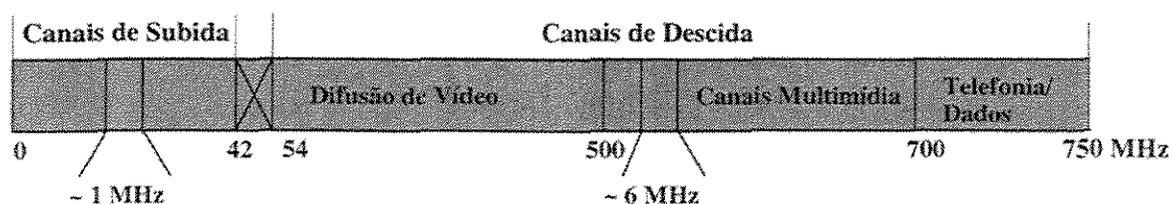


Figura 2.2: Divisão de canais em uma rede HFC.

Esta parece ser a solução mais adequada para a rede de acesso, já que o HFC fornece uma alta quantidade de banda passante, suporta centenas de conexões simultâneas e permite o uso de estratégias de compartilhamento de fluxo. Além disso, a disponibilidade de um canal de retorno, um canal analógico e outro digital permitem que esta rede sirva uma enorme quantidade de aplicações residenciais, que podem variar das tradicionais aplicações de TV a aplicações multimídia altamente sofisticadas.

2.2 Aspectos de Servidores de Vídeo

Em um sistema de VoD, os vídeos (que compõem uma biblioteca previamente digitalizada) devem ser selecionados em tempo real e controlados como em um aparelho de vídeo cassete, através de operações de VCR. Portanto, é necessário que os servidores de vídeo possuam alta capacidade de armazenamento e taxa de transferência de dados.

O subsistema de armazenamento do servidor precisa suportar um grande número de usuários simultaneamente, assim como o controle de *hardware* e *software* para recepção de requisições, localização de vídeos, transferência de dados entre dispositivos na hierarquia de armazenamento, contabilização de custos, controle de admissão, gerenciamento de operações de VCR e criptografia (necessária para evitar o acesso a informações sigilosas, como a senha no processo de identificação do usuário, e garantir que apenas usuários pagantes assistam vídeos quando se utiliza meio de transmissão com acesso compartilhado).

O servidor de vídeo precisa manipular múltiplos fluxos de saída sem variação¹ no tempo necessário para a transferência de bits, quadros ou células de informação. Para manter a vazão requerida faz-se necessária a utilização de (i) um processador poderoso, (ii) um barramento de alta velocidade e (iii) canais e dispositivos de entrada/saída de alta capacidade. Contudo, ainda podem ocorrer flutuações nos fluxos de informação. Para minimizar tais flutuações, os servidores normalmente utilizam grandes *buffers* de saída. Com isso, os dados são acessados rapidamente em grandes blocos dos discos rígidos, armazenados temporariamente no *buffer* e enviados numa taxa pré-definida.

¹Na realidade, uma pequena variação é tolerável e sua amplitude está limitada pelo *buffer* existente na *set top box*.

Nas próximas seções, diversos componentes de um servidor de vídeo são apresentados, com especial ênfase na hierarquia de armazenamento.

2.2.1 Componentes de um Servidor

Um servidor de vídeo compreende:

Hierarquia de Armazenamento: Os recursos de armazenamento são vistos como uma hierarquia (composta por armazenamento terciário, secundário e primário, além de memória RAM e *cache*), geralmente representada graficamente por um triângulo. Neste modelo, dispositivos com baixo tempo de acesso são posicionados no topo, enquanto que dispositivos de desempenho e custos inferiores e maior capacidade de armazenamento localizam-se nos níveis mais próximos à base. A Figura 2.3 ilustra um esquema de gerenciamento de memória compatível com os dispositivos atualmente existentes. O nível de *cache* é dividido em dois subníveis, o *cache* de arquivos, que ocorre no equipamento do usuário; e o *cache* de servidor, que é realizado através da distribuição de diversas cópias de objetos de vídeo em servidores de pequeno porte na rede. Os demais níveis são implementados num servidor de grande porte. Na ilustração, discos rígidos são substituídos por RAIDs (*Redundant Arrays of Inexpensive Disks*) e *jukeboxes* ópticos compõem o nível intermediário entre a memória terciária e primária.

Controladores de Fluxos: O servidor de vídeo precisa produzir um fluxo de bits contínuo isócrono para o usuário. O controlador de fluxos atua como o gerente de tráfego para o servidor. Este dispositivo precisa suportar altas taxas de transações, operações de busca e transferência de dados (entre os diversos níveis de armazenamento) e transmissão dos mesmos. Por exemplo, utiliza-se *buffers* para a determinação do *clock* da taxa de transmissão constante adequada.

Barramento: A maioria dos processadores utiliza arquiteturas baseadas em barramento. Dado que a informação relacionada a muitos vídeos migra do subsistema de discos para os módulos de comunicação e que o vídeo produz um fluxo isócrono com taxa média

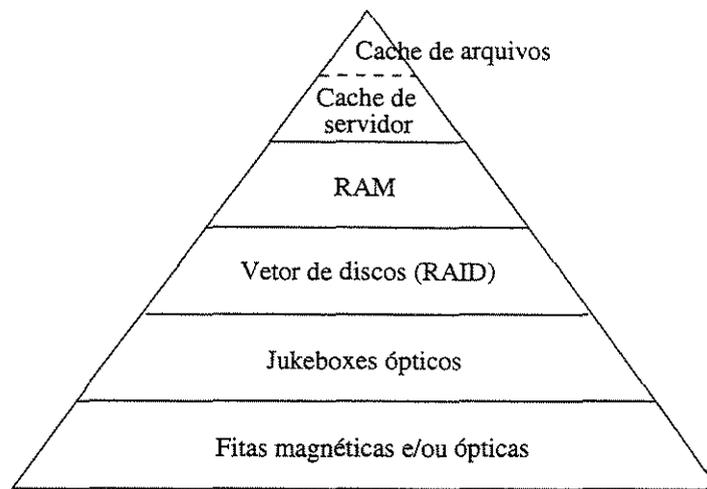


Figura 2.3: Esquema de gerenciamento de memória.

previsível, a capacidade de transferência do barramento é um fator crítico de desempenho. Portanto, uma vez que o tráfego demandado para os usuários precisa passar pelo barramento, a inexistência de um projeto detalhado pode transformá-lo no ponto crítico de desempenho do sistema.

Processador: O processador principal controla o fluxo de dados entre os vários componentes do servidor, as funções de gerenciamento e de contabilização de custos. Novas arquiteturas de computadores podem ser requeridas para suportar o número alto de usuários simultâneos em uma área metropolitana ampla. Alguns sistemas utilizam *mainframes* ou estações RISC (*Reduced Instruction Set Computer*), porém computadores massivamente paralelos são apontados como a plataforma ideal.

2.2.2 Hierarquia de Armazenamento

Os diversos níveis que compõem a hierarquia de armazenamento em um servidor de vídeo são citados a seguir.

Armazenamento Terciário: O armazenamento terciário geralmente consiste de aparelhos de vídeo cassete, fitas magnéticas e discos ópticos organizados na forma de *juke-*

boxes. Também chamado de armazenamento *off-line*, estes dispositivos possuem como características principais: baixo custo, grande capacidade de armazenamento, baixo desempenho (tempo de acesso elevado) e baixa taxa de transferência.

Os *Jukeboxes* Ópticos são grandes agrupamentos de CD-ROMs. Estes dispositivos são construídos posicionando-se os discos em estantes, acessadas por dispositivos robóticos. Quando um arquivo precisa ser acessado, o sistema de gerenciamento de arquivos identifica onde encontrá-lo e o braço mecânico substitui o disco atual pelo que contém o arquivo a ser acessado.

Armazenamento Secundário: O armazenamento secundário tipicamente consiste de tecnologia de armazenamento óptico de alta velocidade ou sistemas RAID. Sistemas mais sofisticados dispensam o nível de armazenamento terciário e armazenam todas as informações neste nível. Outros sistemas preferem utilizar os três elementos da hierarquia de armazenamento. Também chamado de armazenamento *near-line*.

Armazenamento Primário: Constitui-se de discos rígidos convencionais organizados ou não na forma de RAID. Também chamado de armazenamento *on-line*.

Memória RAM e cache: Além da utilização convencional, num servidor de vídeo a memória RAM é alocada para fluxos individuais ou difusão ampla (*broadcast*), também chamada *movieRAM*. Esta memória é utilizada pelo controlador de fluxos para gerenciar mais eficientemente a informação associada aos segmentos de vídeos mais requisitados.

Outros aspectos do projeto de servidores que ainda podem ser mencionados relacionam-se com (i) o projeto de sistemas de arquivos multimídia e organização de dados em disco [10, 11, 12], (ii) balanceamento de carga em discos através da replicação de segmentos de arquivos [13], (iii) políticas de replicação [14], (iv) gerenciamento de banda passante disponível em discos com o objetivo de aumentar a vazão (*throughput*) [15], (v) políticas de escalonamento de discos, gerenciamento de *buffer* e questões de barramento de entrada/saída [16, 17, 18], (vi) além de outras questões sobre arquitetura [19, 20].

Operação	Descrição
<i>Play/Resume</i>	Exibe o vídeo a partir do início ou de um ponto intermediário.
<i>Stop</i>	Interrompe a apresentação. Sem exibição de imagem e som.
<i>Pause</i>	Interrupção temporária. Com exibição de imagem.
<i>Jump Forward</i>	Salta para um ponto posterior. Sem exibição de imagem e som.
<i>Jump Backward</i>	Salta para um ponto anterior. Sem exibição de imagem e som.
<i>Speed Up</i>	Exibição acelerada com imagem e som (<i>fast-forward</i>).
<i>Slow Down</i>	Apresentação lenta com imagem e som.
<i>Reverse</i>	Exibição invertida com imagem e som.
<i>Fast Reverse</i>	Apresentação invertida acelerada com imagem e som.
<i>Slow Reverse</i>	Exibição invertida lenta com imagem e som.

Tabela 2.1: Operações de VCR

2.3 Interatividade em Sistemas de Vídeo sob Demanda

A interação entre usuário e sistema abordada nesta seção diz respeito às operações disponíveis a um usuário para o controle de uma sessão de VoD iniciada, ou seja, as operações de VCR. O advento do vídeo digital tornou possível a existência de novos tipos de operações. Uma relação destas [21] é apresentada na Tabela 2.1. Almeroth [22] divide as operações de VCR em duas categorias:

- Operações Contínuas: Um usuário possui controle completo sobre a duração de uma operação de VCR;
- Operações Descontínuas: A duração das operações é discretizada em múltiplos inteiros de um determinado intervalo de tempo.

A complexidade do sistema de VoD é diretamente proporcional ao grau de interatividade oferecido ao usuário. Nussbaummer *et al.* [23, 24] apresentam uma classificação contendo quatro modalidades para o oferecimento de serviços de VoD. Cada uma delas possui diferentes compromissos entre custo e o nível de interatividade oferecida ao usuário. Elas são citadas a seguir:

TVOD (*True Video on Demand*): Nesta modalidade é permitido ao usuário executar quaisquer operações de VCR. No esquema TVOD, para cada usuário que requisita acesso a um vídeo é alocado um canal lógico. Para permitir interação total, nenhum outro usuário pode acessar este canal simultaneamente, o que limita o número de usuários ao número de canais disponíveis. A ocorrência de uma requisição sem a disponibilidade de canais lógicos implica em descarte sumário do pedido. Este fato é chamado de *blocking*.

Uma variação deste esquema é a criação de uma fila com as requisições pendentes. Para isso, o usuário deve ser notificado do tempo de espera estimado, sendo que sua inclusão na fila só ocorre se o tempo máximo de espera for menor que o comunicado.

NVOD (*Near Video on Demand*)[25]: O esquema NVOD foi projetado de modo a oferecer um serviço de menor custo que o TVOD. Seu princípio básico consiste em iniciar um fluxo de cada vídeo a cada Δt minutos em canais distintos (cada fluxo está Δt minutos atrasado/adiantado em relação ao anterior/posterior), de modo que um usuário que faz um pedido para um filme sendo exibido precisa esperar no máximo Δt minutos para ser atendido, caso o filme solicitado tenha sido selecionado para ser exibido no próximo *slot*. Suas vantagens são a simplicidade e redução de custos, além da alta disponibilidade que permite a todos os usuários conectados ao cabo assistir aos filmes escalonados para exibição. Porém, o esquema possui como desvantagens não dispor de interatividade, com exceção da possibilidade de “saltar” para fluxos posteriores/anteriores e utilizar um grande número de canais, pois, para cada filme de tamanho T são necessários $T/\Delta t$ canais.

PVOD (*Partitioned Video on Demand*): Este esquema constitui-se da utilização conjunta dos anteriores, combinando suas vantagens. Parte dos canais é utilizada com o esquema NVOD para exibição dos vídeos populares, os demais canais são utilizados para exibição dos vídeos não populares, através do esquema TVOD. Os usuários que requisitam vídeos exibidos pela partição NVOD são atendidos por ela, enquanto que os demais, pela partição TVOD. O maior problema com esta abordagem é que apenas parte dos usuários terão acesso à interatividade. Os usuários para os quais foi alocado um canal NVOD não podem interagir com o sistema.

DAVOD (*Dynamically Allocated Video on Demand*): Possui comportamento similar ao PVOD. Contudo, esta modalidade permite o deslocamento de usuários da partição NVOD para TVOD por ocasião da solicitação de operações de VCR. Uma vez transferidos permanecem na partição TVOD até o final da transmissão.

2.4 Técnicas de Redução da Demanda de Banda Passante

O oferecimento de serviços de VoD em larga escala depende em grande parte do controle de admissão e gerenciamento dos recursos alocáveis; porém, em virtude da grande demanda de banda passante requerida, a utilização de outras estratégias torna-se necessária. Nos Estados Unidos (continental) estima-se que o número de aparelhos de TV ligados simultaneamente em horários de pico totaliza 77.000.000. Num sistema de VoD com alocação de fluxos sob demanda, o montante de banda passante requerido totalizaria 462 Tbps para a codificação MPEG-2 NTSC (6Mbps/fluxo), 770 Tbps utilizando-se MJPEG NTSC (10Mbps/fluxo) e 1,54 Pbps para MPEG-2 no formato de HDTV (20Mbps/fluxo) [26]. Estes altos requerimentos evidenciam a necessidade da utilização de técnicas de redução da demanda de banda passante isoladamente ou de forma conjunta. As próximas seções discorrem sobre estas técnicas, com ênfase nas técnicas que propõem o compartilhamento de fluxo.

2.4.1 Replicação e *Caching*

A técnica de replicação [26, 27, 28] propõe a distribuição de diversas cópias de um mesmo objeto de vídeo em vários servidores, objetivando-se reduzir o custo de transmissão (volume de dados) em segmentos da rede. Por um lado, o custo de transmissão está diretamente relacionado à topologia de rede empregada; por outro, o grau de replicação é influenciado pelo custo de armazenamento das diversas cópias. Deste modo, a técnica de replicação objetiva a minimização do custo de transmissão na rede sem, contudo, aumentar demasiadamente o custo de armazenamento.

A técnica de *caching* [29, 30, 31, 32] consiste na implementação da replicação, porém de forma dinâmica. Para isso, faz-se necessário o monitoramento constante dos padrões de requisições dos usuários para determinar quais vídeos serão replicados e em quais servidores da rede. Neste contexto, pode-se explorar o escalonamento prévio de filmes para distribuir suas cópias em períodos de tráfego reduzido.

2.4.2 *Bridging*

A técnica de *Bridging* [33, 34, 35, 36] propõe o armazenado dos últimos k minutos de um vídeo em um *buffer* de memória, de tal forma que se possa atender requisições (por novos fluxos ou de retomada de exibição após operações de VCR) que cheguem ao sistema num intervalo com duração máxima de k minutos após o quadro de filme atualmente exibido. O atendimento de tais requisições pelos dados armazenados no *buffer* reduz, desta forma, a demanda de banda passante nos dispositivos de armazenamento dos servidores de vídeo.

2.4.3 Compartilhamento de Fluxo

Uma abordagem diferente das anteriores é proposta pelas técnicas que propõem o compartilhamento de fluxo como mecanismo para a redução da demanda de banda passante. Com vistas a este objetivo (compartilhamento de fluxo), tais técnicas levam em consideração a probabilidade de um conjunto de requisições pelos vídeos populares chegar ao sistema dentro de um intervalo de tempo relativamente curto, de forma a atender às requisições com um único fluxo. Neste grupo encontram-se as técnicas de *Batching* e *Piggybacking*, detalhadas a seguir.

2.4.3.1 *Batching*

A técnica de *Batching* [37, 34, 38, 39, 40, 41, 42] consiste em reter requisições por um determinado intervalo de tempo com o objetivo de agrupar o máximo número de requisições e servir ao grupo com um único fluxo de vídeo. A este intervalo dá-se o nome de **Janela** ou **Intervalo de *Batching***² e seu dimensionamento representa um compromisso entre a

²Não há uma nomenclatura padronizada para se referir ao tempo de espera a que um usuário é submetido enquanto o sistema aguarda a chegada de novas requisições do mesmo vídeo. Nesta dissertação,

minimização do tempo esperado pelo usuário e a maximização do **Efeito de *Batching***, ou seja, o número médio de usuários servidos por fluxo.

A seguir estão relacionadas as políticas de *Batching* propostas na literatura.

Espera Forçada: Exige que pelo menos uma das requisições seja retida durante a janela de *Batching*. Ao final da mesma, aloca-se um fluxo para o vídeo solicitado. Este funcionamento pode causar, nos horários de pico, o fenômeno da formação de ciclos, que consistem em momentos de intensa alocação de canais seguidos por longos intervalos de espera, isto é, índices altos de rejeição de pedidos. Os ciclos decorrem da alocação independente de canais para vídeos distintos, uma vez que o intervalo de espera é aplicado em cada fila independentemente das demais.

Controle de Taxa Puro: Política que (i) impõe um limite superior na taxa a qual canais podem ser alocados, e (ii) limita o número total de canais que podem ser alocados durante um intervalo de tempo fixo (*intervalo de medida*) com o objetivo de evitar a formação de ciclos (como descrito anteriormente).

Controle de Taxa Desviado: Difere da política anterior, pois permite que a taxa de alocação, durante os horários de pico, esteja momentaneamente acima da taxa de alocação máxima com o objetivo de evitar o abandono de usuários (sem serviço) caso um canal não seja alocado em um curto intervalo de tempo.

Maior Tamanho de Fila (*Maximum Queue Length*) — MQL: Busca-se maximizar o efeito de *Batching* alocando-se o fluxo para a fila cujo número de requisições pendentes seja maior. Sua maior vantagem é a grande redução de demanda de banda passante. Esta política é injusta com os filmes não populares, e o tempo de espera é imprevisível.

Fila de Maior Índice (*Maximum Factored Queue Length*) — MFQL: Define-se um índice que determina qual das filas de requisições receberá o fluxo a ser alocado, este ambos os termos são utilizados indistintamente.

índice representa um compromisso entre a maximização do efeito de *Batching* e o tempo de espera na fila. Busca-se, desta forma, maximizar a utilização dos recursos do sistema sem desprezar os vídeos menos populares.

Primeiro a Chegar, Primeiro a ser Atendido (*First Come First Served*) — **FCFS:** Nesta política, considera-se principalmente o aspecto de justiça entre os vídeos (populares e não populares) quando da alocação de fluxos. Assim sendo, todas as requisições são inseridas em uma única fila e a alocação é realizada conforme a ordem de chegada. Quando um canal torna-se disponível, a requisição da cabeça da fila é atendida e todas as outras requisições para o mesmo filme também são servidas por este canal. Apesar de ser justa, como não leva em consideração o tamanho do grupo (*batch*), esta política possui um menor desempenho em relação a redução de banda passante.

Primeiro a Chegar, Primeiro a ser Atendido- n (*First Come First Served- n)* — **FCFS- n :** Esta é uma extensão da política FCFS. Canais dedicados são atribuídos aos n filmes mais populares, de tal forma que um novo fluxo para estes n filmes pode ser iniciado a cada w segundos. Assim o tempo de espera é no máximo w segundos para estes filmes. Além disso, como os filmes populares possuem taxa de requisições altas, cada fluxo serve mais de um usuário. Se o tempo w for ajustado apropriadamente à taxa de chegada, os recursos do sistema podem ser substancialmente reduzidos.

Paradigma de Tolerância de Espera: Busca-se maximizar o efeito de *Batching*. Com este objetivo, define-se um *intervalo de espera* (*wait threshold*), ou seja, um intervalo de tempo no qual requisições são retidas, explorando-se o tempo estimado de espera dos usuários. Definem-se duas classes de políticas:

Max_Batch: Um vídeo é escalonado para alocação se pelo menos uma de suas requisições foi retida durante o intervalo de espera. Caso mais de um vídeo satisfaça esta condição, aloca-se o fluxo utilizando-se o critério de maior tamanho de fila (*Max Batch MQL* — MBQ) ou de maior número estimado de abandonos (*Max Batch with Minimal*

Loss — BML). Caso contrário, um fluxo permanece disponível até que uma requisição satisfaça a este critério.

Min_Idle: Dividem-se os vídeos em dois conjuntos: populares, \mathcal{H} (*hot videos*), e não populares, \mathcal{C} (*cold videos*). Aplica-se *Batching* apenas sobre o conjunto \mathcal{H} , sem a restrição de tempo mínimo de espera no conjunto \mathcal{C} . Um vídeo é inserido em \mathcal{H} se *i*) é popular, *ii*) sua fila possui mais de uma requisição, e *iii*) a única requisição de sua fila excedeu o intervalo de espera. Se o conjunto \mathcal{H} não estiver vazio, aloca-se um vídeo utilizando-se o critério de maior tamanho de fila (*Min Idle MQL* — IMQ) ou maior número esperado de perdas (*Min Idle with Minimal Loss* — IML). Caso contrário, aloca-se um fluxo para um vídeo do conjunto \mathcal{C} utilizando-se a política FCFS.

Look-Ahead-Maximize-Batch — LAMB: Esta é a política que admite o maior número de usuários dentre as políticas de *Batching* apresentadas. Ela se utiliza do conhecimento sobre o comportamento de abandono dos usuários, com o objetivo de maximizar o efeito de *Batching*.

A política LAMB assume que a janela de *Batching* possui duração determinada pelo tempo de abandono do usuário. A alocação de fluxos ocorre quando o tempo de abandono esgota-se, isto é, os *instantes de alocação* de fluxos ocorrem quando um usuário está prestes a abandonar o sistema. Deste modo, os usuários devem ser inseridos nas filas obedecendo-se a ordenação temporal dos eventos de abandono, ou seja, a primeira requisição de uma fila é aquela cujo abandono ocorre primeiro; não obrigatoriamente é a primeira a chegar ao sistema. Esta característica impõe intervalos de espera distintos, nos quais espera-se a chegada de novas requisições, explorando-se ao máximo o tempo de abandono dos usuários, enquanto busca-se minimizar as desistências.

A janela de *Batching* consiste num intervalo de tempo cujos extremos são definidos pelos primeiro e último instantes de alocação das primeiras requisições nas diversas filas, ou seja, seu ponto inicial é determinado pelo instante de alocação atual (intervalo de *Batching* atualmente esgotado) e o ponto final constitui-se no último instante de alocação dentre as primeiras requisições de cada fila. Adicionalmente, ela inclui todos os eventos

que representam instantes de alocação que ocorrem neste intervalo e também os eventos de liberação de fluxos, que ocorrem em virtude do final da exibição de um vídeo.

LAMB maximiza o número de usuários admitidos no sistema considerando todas as potenciais perdas na janela de *batching*. O critério de alocação de fluxos é o mesmo para todos os vídeos (independentemente de sua popularidade), o qual pode assumir dois modos de atuação. Se o número de vídeos com requisições pendentes é menor que o número de canais atualmente disponíveis no sistema, aloca-se um canal para o vídeo cuja janela de *Batching* esgotou-se. Caso contrário, é necessário avaliar se alocar um canal neste instante não levará a falta de canais para futuros instantes de alocação associados a filas maiores. É válido lembrar que sempre que se aloca um canal a uma fila, todos os usuários desta são atendidos por um único canal. Além disso, se a alocação do canal for recusada, apenas o usuário associado a este instante de alocação é perdido.

Para maximizar o número de usuários admitidos durante a janela de *batching*, ou equivalentemente, minimizar o número de usuários perdidos, é necessário determinar quando um canal deve ser alocado a cada fila, levando em consideração as informações disponíveis no atual instante de alocação.

A maximização do número de usuários admitidos pode ser formulada como um problema de programação linear inteiro 0-1, descrito a seguir:

$$\max \sum_{k=1}^F \sum_{i=1}^{q_k} r_k \times (q_k - i + 1) \times x_{k,i} \quad (2.1)$$

Sujeito às seguintes restrições:

Restrição 1: Número de fluxos alocados por fila;

$$\sum_{i=1}^{q_k} x_{k,i} \leq 1, \quad k = 1, \dots, F \quad (2.2)$$

Restrição 2: Número total de fluxos alocáveis;

$$\begin{aligned} x_{k,i} + \sum_{k'=1}^V \sum_{j=1}^{q_{k',z}} x_{k',j} &\leq S + A_{k,i} && k' \neq k; t_{k',j} < t_{k,i}; \\ & && k = 1, \dots, F; \\ & && i = 1, \dots, q_k \text{ e } t_{k,i} \in \mathfrak{F} \end{aligned} \quad (2.3)$$

Restrição 3: Integridade da solução;

$$x_{k,i} = 0, \quad \forall t_{k,i} \notin \mathfrak{S}, \quad k = 1, \dots, F \quad \text{e} \quad i = 1, \dots, q_k \quad (2.4)$$

Restrição 4: Definição de limites;

$$\begin{aligned} 0 &\leq S \leq C \\ 0 &\leq A_{k,i} \leq C \quad k = 1, \dots, F \quad \text{e} \quad i = 1, \dots, q_k \\ 0 &\leq q_k \leq \infty \quad k = 1, \dots, F \\ 0 &\leq x_{k,i} \leq 1 \quad k = 1, \dots, F \quad \text{e} \quad i = 1, \dots, q_k \end{aligned} \quad (2.5)$$

onde:

\mathfrak{S}	Janela de <i>batching</i> . $\mathfrak{S} = [\min\{e_{k,1}\}, \max\{e_{k,1}\}]$, $k = 1, \dots, F$
$e_{k,1}$	Tempo de abandono do primeiro usuário da k -ésima fila.
$A_{k,i}$	Número de canais liberados desde o início da janela de <i>batching</i> até o instante $t_{k,i}$.
F	Número total de vídeos armazenados no servidor. Considera-se uma constante, pois assume-se que o número de vídeos no servidor é fixo.
C	Capacidade do Servidor, ou seja, número total de canais.
S	Número de canais disponíveis no instante $\min\{e_{k,1}\}$.
q_k	Número de requisições atualmente contidas na fila k .
r_k	Custo para assistir ao vídeo k (<i>revenue</i>), portanto possui valor positivo. r_k pode ser considerado uma constante para o problema pois seu valor é definido previamente.
$t_{k,i}$	Instante de abandono da i -ésima requisição do vídeo k .
$x_{k,i}$	Variável que assume valor $x_{k,i} = 1$ caso ocorra a alocação de um canal no instante $t_{k,i}$, caso contrário, $x_{k,i} = 0$.

A Equação 2.1 (função objetivo) traduz o ganho obtido com a configuração de alocação de canais determinada pela solução do problema. Pode-se perceber que para a maximização da função objetivo, a alocação dos canais deve ocorrer nas primeiras requisições de

cada fila. Porém, em determinadas situações, tal fato pode não ocorrer em virtude da limitação no número de canais disponíveis.

A restrição representada pela Equação 2.2 determina que, para o conjunto de requisições atualmente inseridas numa fila, o número máximo de canais alocados é 1. Esta é uma restrição óbvia, pois, uma vez que um canal é alocado a uma fila, todos os usuários desta são servidos por este único canal.

A Equação 2.3 expressa que o número total de canais alocados até o instante $t_{k,i}$ é limitado superiormente pelo número de canais disponíveis no início da janela de *batching*, somado ao número de canais liberados desde então.

Note que, se na solução de um determinado problema, a variável $x_{k,i}$ for igual a 1, a solução obtida indica que as requisições $1, \dots, i - 1$ não são atendidas (abandonando o sistema) e que no instante $t_{k,i}$ ocorrerá a alocação de um canal para o vídeo k , suportando as requisições i, \dots, q_k . Esta é a solução ótima considerando as informações disponíveis no instante de alocação atual.

O problema de otimização formulado indica se é atrativo (ou não) para o sistema alocar um canal para o vídeo cuja requisição representa o primeiro evento da janela de *batching*, isto é, a requisição cujo tempo de abandono esgotou-se. Através da análise realizada, pode-se decidir retardar a alocação para o instante de alocação de uma fila com um número maior de requisições. Em outras palavras, se o instante de alocação atual (intervalo de *Batching* esgotado) pertence à fila do vídeo k , aloca-se um canal se $x_{k,1}$ for igual 1. Caso contrário, a alocação é retida e a primeira requisição da k -ésima fila abandona o sistema, de modo que para cada requisição cujo intervalo de *Batching* esgota-se, define-se uma nova janela de *batching* e modela-se um novo problema.

A estratégia descrita acima privilegia os vídeos populares. No entanto, pode-se perceber que este é um mecanismo adaptativo, ou seja, as soluções dos problemas modelados tornam-se mais favoráveis em relação aos vídeos populares à medida em que o número de canais disponíveis decresce. No caso extremo, a alocação do único canal disponível é retardada até que ocorra o instante de alocação para a primeira requisição da maior fila.

2.4.3.2 Piggybacking

A técnica de *Piggybacking* [35, 36, 43, 44, 45] baseia-se no fato de que alterações da ordem de 5% na taxa de exibição não são perceptíveis aos usuários. Desta forma, as requisições são atendidas imediatamente e o fluxo compartilhado é obtido através da superposição de fluxos dessincronizados, ou seja, altera-se a taxa de exibição dos fluxos de vídeo de tal forma que estes venham a exibir um mesmo quadro de filme em um determinado instante, descartando-se um dos fluxos após a sincronização.

Se por um lado o *Piggybacking* não proporciona retardo inicial como no *Batching*, por outro, não é tão eficiente em termos de redução da demanda de banda passante. Isto porque, na chegada de uma requisição, um canal lógico é alocado imediatamente (se houver algum disponível) e este somente será liberado por ocasião de uma mesclagem ou finalização da exibição do vídeo.

Na técnica de *Piggybacking* denomina-se **Janela de Mesclagem** (*Catch up Window*), $W_{Pol}(f_i)$, a distância, dada em quadros e calculada relativamente à posição f_i , que indica a possibilidade de sincronização de dois fluxos para a política *Pol*. Quanto às taxas de exibição, definem-se as taxas mínima, normal e máxima, em quadros por segundo, denotadas respectivamente por, V_{min} , V_n e V_{max} .

Um aspecto importante a ser considerado constitui-se em como obter as variações nas taxas de exibição dos quadros dos vídeos. Uma solução simples seria alterar o parâmetro de frequência de amostragem dos quadros durante a codificação e armazenar versões alteradas juntamente com as originais, aumentando-se o custo de armazenamento. Uma segunda alternativa para se obter a alteração das taxas de apresentação dos quadros consiste em utilizar dispositivos que possam inserir ou retirar quadros, obtendo assim expansão ou compactação através de interpolação. As desvantagens desta última abordagem são que *i*) o *layout* dos dados no disco geralmente está adequado a uma determinada taxa de transmissão. Assim sendo, o suporte a diferentes taxas pode prejudicar o escalonamento e/ou requisição de espaço adicional de *buffer* de armazenamento; *ii*) o suporte à modificação em tempo real implica em custo adicional com *hardware* especializado.

A seguir estão relacionadas as políticas propostas na literatura para implementação

de *Piggybacking*.

Par-Ímpar: Define-se $W_{pi}(0)$ como a janela de mesclagem para a política par-ímpar relativa ao início de um vídeo. Para cada fluxo iniciado pelo sistema em W_{pi} , atribuem-se as “velocidades” V_{min} e V_{max} alternadamente, de modo que ao primeiro fluxo atribui-se V_{min} (Figura 2.4). Pela formulação desta política, o ganho máximo obtido é de 50%.

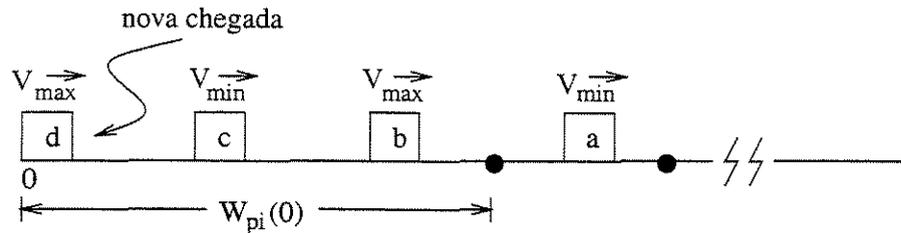


Figura 2.4: Ilustração do funcionamento da política par-ímpar.

Mesclagem Simples: Definem-se duas janelas de mesclagem, $W_{ms}(0)$ e $W_{ms}^m(0)$ (janela máxima de mesclagem da política mesclagem simples), calculadas com relação ao início do vídeo. $W_{ms}^m(0)$ indica a última posição em que dois fluxos podem sincronizar-se, ou seja, se o fluxo i chega ao sistema e encontra o fluxo j $W_{ms}(0)$ quadros à sua frente, i e j são sincronizados no limite final de $W_{ms}^m(0)$.

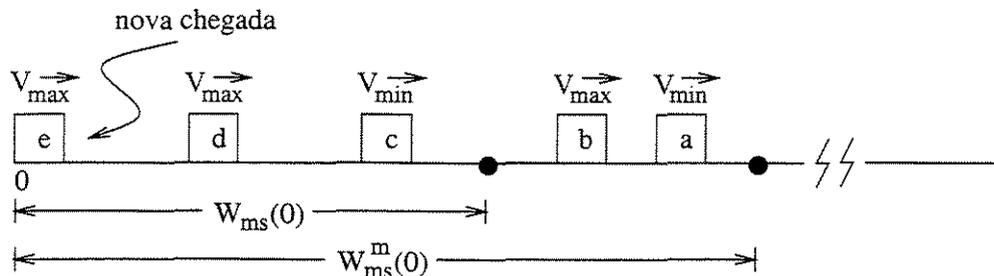


Figura 2.5: Ilustração do funcionamento da política mesclagem simples.

Esta abordagem associa fluxos a “grupos de mesclagem”, em que os fluxos são superpostos formando um único fluxo resultante antes de deixarem $W_{ms}^m(0)$. Se o grupo é

composto de n fluxos, o primeiro terá sua taxa ajustada em V_{min} e os demais em V_{max} (Figura 2.5), até o instante da última mesclagem, quando o fluxo resultante passa a ter velocidade V_n .

Gulosa: Esta política realiza mesclagens sempre que possível durante a exibição do vídeo. Com este objetivo, definem-se janelas de mesclagem adicionais além da inicial, $W_g(0)$, calculadas relativas à uma posição f_i , $W_g(f_i)$. Esta janela de mesclagem é utilizada como um indicativo para as mesclagens adicionais.

A política gulosa funciona como segue: se ao cruzar a janela os fluxos ainda não estiverem sincronizados, calcula-se $W_g(W_g(0))$ para verificar a possibilidade de mesclagens com fluxos anteriores. Quando uma mesclagem ocorre na posição f_i , uma nova janela de mesclagem é calculada, $W_g(f_i)$. Se não houver fluxos na janela, a taxa de exibição é ajustada para V_n , caso contrário, a velocidade do fluxo anterior é ajustada em V_{min} e a do fluxo atual em V_{max} .

A Figura 2.6 ilustra a política gulosa. Neste caso, os fluxos b e d já foram mesclados com a e c , respectivamente.

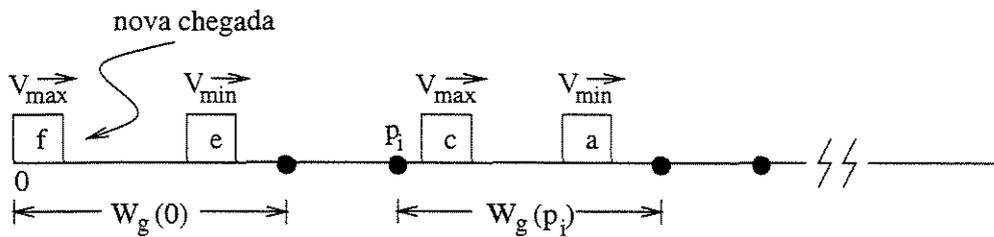


Figura 2.6: Ilustração do funcionamento da política gulosa.

Mesclagem Simples Generalizada: Define-se a janela de mesclagem máxima, W_m , como a última posição em quadros que permite a ocorrência de uma mesclagem antes do fim do vídeo. Dada a diferença das taxas, calcula-se W_m como:

$$W_m = \frac{V_{max} - V_{min}}{V_{max}} \cdot L,$$

em que L é o número de quadros do vídeo.

O dimensionamento da janela de mesclagem é um fator importante porque:

- Quando o tamanho da janela é grande, um número maior de fluxos pode ser mesclado em um único fluxo. Porém, as mesclagens ocorrem próximas do final do vídeo;
- Quando o tamanho da janela é muito pequeno, as mesclagens tendem a ocorrer próximas do início do vídeo, porém, em quantidade reduzida.

Pode-se perceber que ambas as situações não favorecem à redução do número de quadros exibidos por um conjunto de fluxos. Aggarwal *et al.* [43, 44] apresentam um método analítico para otimizar o tamanho da janela de mesclagem, o qual leva em consideração o tamanho do vídeo e a taxa de chegada prevista, assumindo que as chegadas são modeladas por um processo de Poisson com parâmetro λ . Com isso, a janela de mesclagem derivada otimiza o número e a posição, relativamente ao tamanho do vídeo, em que mesclagens ocorrem; e, deste modo, propicia uma maior redução na demanda de banda passante.

Assim sendo, define-se a política mesclagem simples generalizada em função do parâmetro W ($0 < W \leq W_m$) chamado **Janela Ótima de Mesclagem**, sendo esta a única diferença entre esta política e sua versão original, definida em função de W_m . A Figura 2.7 ilustra o dimensionamento da janela ótima de mesclagens em função da janela máxima. Observa-se que à medida em que o intervalo médio entre chegadas assume valores maiores, o tamanho de W aproxima-se do tamanho de W_m .

Quanto às taxas de exibição, V_{max} é associada a um fluxo se já existir um outro em W com taxa V_{min} , caso contrário, associa-se V_{min} . Quando um fluxo “rápido” sincroniza-se com um “lento”, descarta-se o rápido.

Algoritmo *Snapshot*: Dentre as políticas de *Piggybacking* mencionadas, a política *Snapshot* [43, 44] é a que busca minimizar o número de quadros exibidos por um conjunto de fluxos de vídeo de forma mais eficiente.

A computação realizada pelas políticas de *Piggybacking* pode ser vista como uma árvore binária em que as folhas correspondem aos fluxos, os nós internos são as mesclagens intermediárias e a raiz é a mesclagem final, formando o fluxo resultante do conjunto (Figura 2.8). Estas representações gráficas das estratégias de mesclagem de um conjunto de fluxos são chamadas de **Árvores de Mesclagens**. Portanto, o número de possíveis

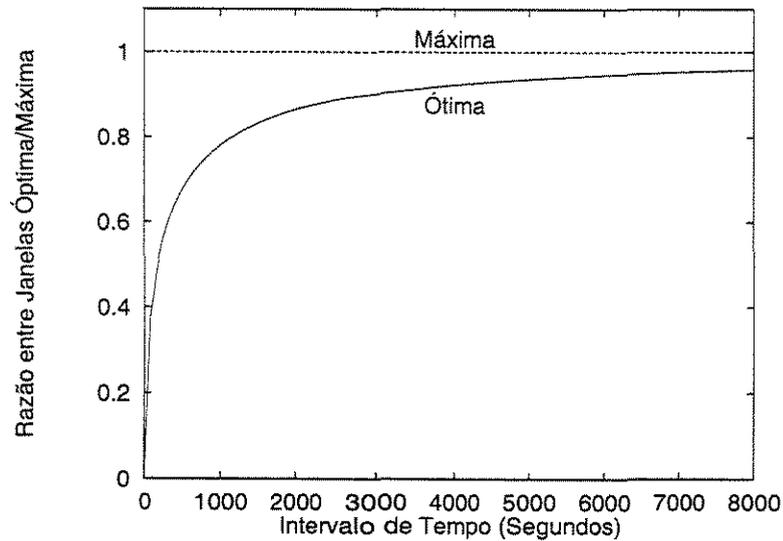


Figura 2.7: Dimensionamento da janela ótima relativo à janela máxima em função do intervalo médio entre requisições.

árvores formadas a partir de um conjunto de fluxos constitui-se no número de políticas de *Piggybacking* potencialmente ótimas e é dado pelo $(n - 1)$ -ésimo número de **Catalan** [43, 44, 46]:

$$Catalan(n - 1) = \frac{(2n - 2)!}{(n - 1)!n!} \quad (2.6)$$

o que implica que o número de árvores cresce rapidamente inviabilizando uma busca exaustiva da estratégia ótima e sua árvore binária correspondente.

Deste modo, a política *Snapshot* constrói a árvore ótima de mesclagem de fluxos de vídeo da seguinte forma: considere um conjunto de n fluxos de um mesmo vídeo (o qual é composto de L quadros) e suas posições dadas em quadros e denotadas por f_1, f_2, \dots, f_n , em que $f_1 \geq f_2 \geq \dots \geq f_n$ em um determinado instante de tempo. Sejam i e j dois fluxos entre 1 e n , com $i \leq j$. Denota-se $Pos(i, j)$ como a posição de mesclagem (em quadros), na qual os fluxos i e j apresentam o mesmo quadro de filme. Uma vez que o fluxo i possui velocidade V_{min} e o fluxo j , V_{max} , a posição de mesclagem é dada por:

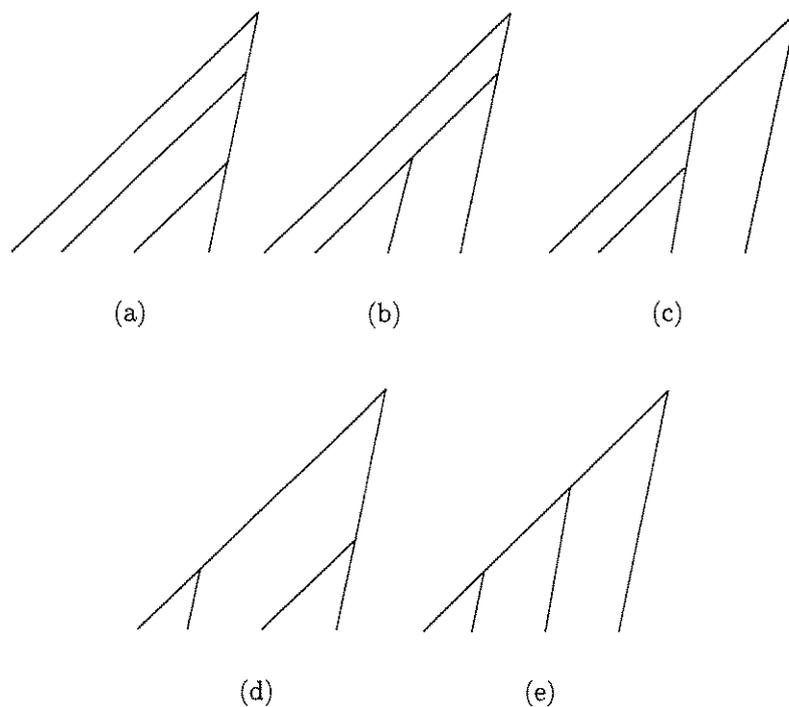


Figura 2.8: Possíveis árvores de mesclagens para quatro fluxos.

$$Pos(i, j) = f_i + \frac{V_{min} \cdot (f_i - f_j)}{V_{max} - V_{min}} \quad (2.7)$$

Seja $C(i, j)$ o custo de uma política e $\mathcal{A}(i, j)$ sua árvore binária correspondente. O custo para cada fluxo é dado pelo número de quadros exibidos desnecessariamente, ou seja:

$$C(i, i) = L - f_i \quad (2.8)$$

Para se obter o custo $C(i, j)$ mínimo é necessário que o princípio de otimalidade seja satisfeito, ou seja: *Existe um fluxo k , com $i \leq k < j$, tal que as subárvores esquerda (i, \dots, k) e direita $(k + 1, \dots, j)$ também são ótimas.* Estas subárvores são denotadas por $\mathcal{A}(i, k)$ e $\mathcal{A}(k + 1, j)$. O custo da árvore que corresponde ao conjunto de fluxos i, \dots, j é dado por:

$$C(i, k) + C(k + 1, j) - \max(L - Pos(i, j), 0) \quad (2.9)$$

Portanto, a política ótima para os fluxos i, \dots, j possui subárvores i, \dots, k^* e $k^* + 1, \dots, j$, em que:

$$k^* = \operatorname{argmin}_{i \leq k < j} \{C(i, k) + C(k + 1, j) - \max(L - Pos(i, j), 0)\} \quad (2.10)$$

Deste modo, o custo para o conjunto dos n fluxos, $C(1, n)$, pode ser calculado de forma *bottom-up* através de um algoritmo de programação dinâmica.

Assim como as demais políticas de *Piggybacking*, a política *Snapshot* atribui velocidades aos fluxos que são disparados pelo sistema dentro de um intervalo, neste caso o intervalo *Snapshot*, denotado por I . O intervalo *Snapshot* é dado por $I = W/V_{max}$, em que W é o valor ótimo da janela de mesclagem derivado para a política mesclagem simples generalizada [44, 43]. Dentro do intervalo I , a política *Snapshot* comporta-se como a política mesclagem simples. Ao final do intervalo os procedimentos de otimização descritos anteriormente são aplicados.

Política S²: Esta política é uma extensão à política *Snapshot*, utilizando dois níveis de mesclagem (otimização).

A política *Snapshot* visa minimizar o número de quadros exibidos pelos fluxos iniciados durante o último intervalo I , de tal modo que é necessário aguardar o final deste intervalo. É interessante mencionar que quando o tamanho de W é pequeno com relação ao valor da janela máxima de mesclagem, W_m , a redução no número de quadros exibidos por todos os fluxos no sistema não é ótima, mesmo que as reduções locais (em cada intervalo) sejam.

É fácil verificar que, quanto menor o intervalo médio entre requisições (taxas de chegadas maiores) as janelas ótimas de mesclagem serão cada vez menores, como pode ser visto na Figura 2.7, podendo haver uma ou várias janelas ótimas contidas numa mesma janela máxima de mesclagem. A possibilidade da superposição de dois fluxos separados por no máximo W_m quadros permite que ganhos superiores aos da política *Snapshot* possam ser obtidos. Assim, a política S² busca otimizar o número de quadros exibidos por um conjunto de fluxos disparados pelo sistema numa **Janela Máxima Alterada** de mesclagem, W'_m , em que $W \leq W'_m \leq W_m$, e não somente no intervalo I . A janela máxima alterada é constituída por um número inteiro de janelas ótimas, isto é, a janela máxima alterada

possui $\lfloor W_m/W \rfloor$ janelas ótimas. Em outras palavras, a política S^2 introduz um segundo nível de mesclagens, ou seja, a mesclagem das resultantes dos intervalos I da política *Snapshot*.

O funcionamento da política S^2 é como se segue: aplica-se primeiramente o algoritmo *Snapshot* sobre os fluxos iniciados pelo sistema nos intervalos *Snapshot* (como proposto originalmente). Em seguida, aplica-se também o algoritmo de programação dinâmica sobre os fluxos resultantes destes intervalos. É interessante enfatizar que, de acordo com a definição da própria política *Snapshot*, atribui-se a velocidade V_{max} a estes fluxos resultantes, com exceção daquele gerado pela primeira janela ótima contida em W'_m . Outro aspecto importante é que, ao contrário do que ocorre com o primeiro nível de otimização, o ponto de aplicação do procedimento de otimização não ocorre ao final de intervalos de duração fixa. Estes intervalos são denotados por I_{S^2} e sua duração é determinada pelo padrão das requisições de vídeos em cada janela W'_m (Figura 2.9).

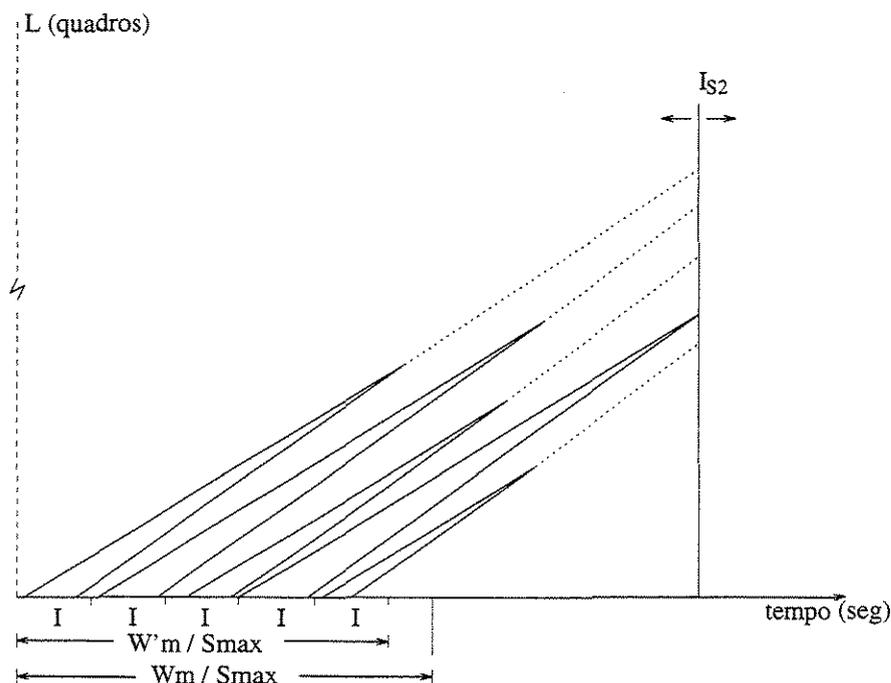


Figura 2.9: Possível situação da política S^2 na qual o último fluxo resultante gerado não corresponde ao fluxo resultante da última janela ótima contida na janela W'_m .

Uma generalização natural da política S^2 seria considerar n níveis de otimização. No entanto, os ganhos obtidos com a implementação destes níveis seriam praticamente nulos, dado que os fluxos nestes níveis estariam separados por valores bem próximos a W_m quadros (ou ainda maiores), o que implica em mesclagens próximas do (ou após o) final do vídeo. Portanto, a introdução de níveis extras de mesclagens não proporciona reduções efetivas na demanda de banda passante.

2.5 Distribuições Estatísticas Comuns em VoD

Para a avaliação de diversos aspectos dos sistemas de Vídeo sob Demanda, são utilizados experimentos de simulação. Para tanto, é necessário levantar as características e o comportamento do sistema. As distribuições estatísticas utilizadas na modelagem dos diversos parâmetros do sistema são apresentadas nesta seção [47].

As chegadas das requisições de vídeo são modeladas por um processo de Poisson, cujo intervalo médio entre requisições é exponencialmente distribuído com média $1/\lambda$.

A probabilidade de escolha de um vídeo é caracterizada pela da distribuição **Zipf** com parâmetro $\theta = 0,271$ [40, 48]. Nesta distribuição, a probabilidade de escolha de um vídeo k é dada por $p_k = \frac{z}{k^{(1-\theta)}}$, em que θ é o parâmetro para a distribuição e z é a constante de normalização, dada por:

$$z = \frac{1}{\sum_{k=1}^F \frac{1}{k^{(1-\theta)}}}. \quad (2.11)$$

No gráfico da Figura 2.10 mostra-se a caracterização das diversas probabilidades de escolha dos vídeos conforme a distribuição Zipf.

O tempo de abandono dos usuários pode ser modelado conforme o esquema proposto por Philip Yu *et al.* [38]. Neste esquema, assume-se que cada usuário espera durante um intervalo de tempo cuja duração é definida pela distribuição Normal (*Gaussiana*) com média μ e variância σ^2 , ou seja, $Abd \sim N(\mu, \sigma^2)$. Deste modo, considera-se que após R minutos o usuário abandona o sistema sumariamente.

O intervalo entre as chegadas de requisições de operações de VCR pode ser modelado de maneira adequada por uma distribuição exponencial. A duração das operações de

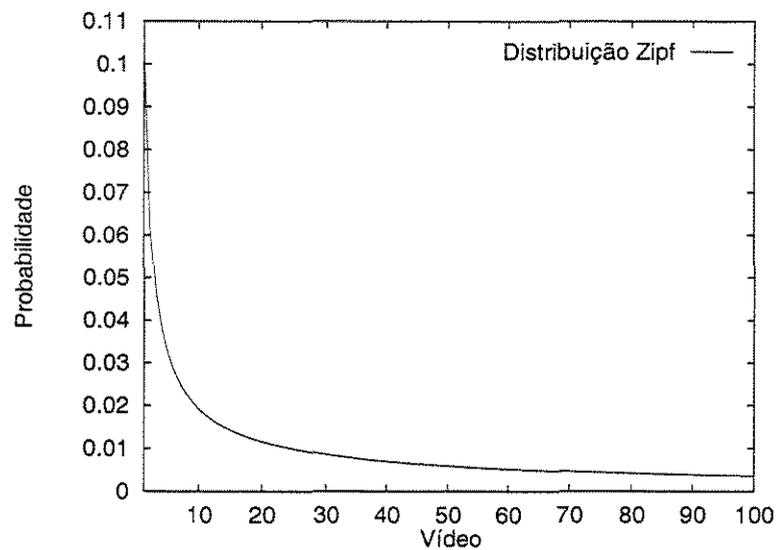


Figura 2.10: Probabilidade de escolha dos vídeos conforme a distribuição Zipf com parâmetro $\theta = 0,271$.

VCR é modelada por uma distribuição exponencial com média λ_{op} . Porém, quando são necessários altos níveis de precisão, a utilização de distribuições lognormais é mais indicada [49].

Finalmente, o tipo da operação de VCR requisitada pode ser determinado através de uma distribuição uniforme, onde a probabilidade de cada uma das possíveis operações (PAUSE, FF, REW) é previamente estabelecida.

Capítulo 3

Conceitos em Redes de Filas e Fórmula de Erlang

Um conjunto de filas interconectados é normalmente utilizado na modelagem de sistemas de computadores e redes de comunicação. A característica comum destes sistemas é um recurso limitado que deve ser compartilhado entre um número de usuários que requisitam serviços. Modelagem por redes de filas fornece um meio para a análise de sistemas que são estocásticos por natureza e que envolvem um fluxo de tráfego em uma rede de recursos.

Além do modelo de redes de filas, este capítulo apresenta a fórmula Erlang B, muito utilizada em engenharia de tráfego e em projetos de redes de telecomunicações.

Os conceitos aqui apresentados serão utilizados no próximo capítulo (4) para a modelagem do sistema de vídeo interativo.

A seção 3.1 apresenta uma introdução a teoria de redes de filas, focando sua atenção em redes fechadas e em seus algoritmos exatos. A seção 3.2 apresenta a fórmula Erlang B.

3.1 Redes de Filas

Uma Rede de filas é um conjunto de filas interconectadas. Modelos de redes de filas são os métodos analíticos mais utilizados para estimar as medidas de desempenho padrões em sistemas de computadores.

Um modelo analítico tipicamente fornece um algoritmo para calcular a proporção de tempo que n_i usuários estão presentes no dispositivo i , os tamanhos das filas em cada dispositivo, etc. Se o número total de usuários no sistema for constante, o modelo é *fechado*. De outro modo, se a taxa de chegada for especificada, o modelo é *aberto*, e neste caso é permitido aos usuários entrar e sair da rede. Usuários podem ser diferenciados por sua *classe* r ; uma rede com múltiplas classes é chamada *rede multiclasse*.

3.1.1 Product Form Networks

A solução direta de uma rede de filas genérica poderia envolver uma solução complexa baseada em equações balanceadas. Entretanto, sob algumas hipóteses pode-se encontrar um balanceamento local para as equações. Neste caso, diz-se que o sistema tem uma solução na forma de produtos (*product form solution*) que pode ser calculada de forma eficiente.

O teorema BCMP, desenvolvido por Baskett, Chandy, Muntz e Palacios, especifica a combinação de distribuições de tempo de serviço e disciplinas de escalonamento que produz redes de filas multiclasse na forma de produtos [50]. São permitidas redes abertas, fechadas e mistas. Basicamente o conjunto de restrições requeridas pelo teorema BCMP para soluções *product-form* são as seguintes:

- *Centrais de serviço com disciplina FCFS* — Neste caso, usuários são atendidos pela ordem em que chegam. As distribuições de tempo de serviço devem ser exponenciais com a mesma média para todas as classes. As classes podem ter diferentes taxas de visitas (*visit ratio*). Além disso, a taxa de serviço pode ser dependente do tamanho total da fila do servidor, mas nunca do tamanho da fila de nenhuma classe em particular.
- *Centrais de Serviço com disciplina PS* — Quando há n usuários em um servidor com uma disciplina *Processor Sharing*, cada usuário recebe serviço a uma taxa de $1/n$ de sua taxa de serviço normal. Cada classe pode ter uma distribuição de tempo de serviço distinta.

- *Centrais de serviço com servidores infinitos (IS)* — Há um número infinito de servidores na central de serviço, ou seja, nunca há espera por um servidor. Esta situação é conhecida como IS (*Infinite Server*), servidor de retardo (*delay server*) ou sem fila.
- *Centrais de serviço com disciplina LCFS-PR* — Sob *Last Come First Served Preemptive Resume*, sempre que um novo usuário chega, o servidor cessa (ou seja, preempta) o serviço do usuário anterior (se houvesse algum) e começa a servir a nova chegada. Quando o usuário termina, o servidor recomeça executar o serviço previamente preemptado. Cada classe pode ter uma distribuição de serviço distinta.

3.1.2 Redes Fechadas

Em sistemas de computadores, dois tipos de processamento, *batch* e interativo, são normalmente modelados como redes de filas fechadas do tipo BCMP. A característica principal é que o total de carga oferecido ao sistema é constante; nenhum usuário deixa o sistema, como mostrado na Figura 3.1. Este modelo é conhecido como modelo de servidor central (*central-server*).

No caso do processamento *batch*, assume-se que o sistema opera sob *backlog*, o que significa que sempre que um usuário sai do sistema ele é substituído por um novo. Desta forma, o número de usuários sendo servidos no sistema permanece constante.

No sistema interativo, cada terminal é associado a um usuário, que permanece nesta fila enquanto está “pensando”. O estado “pensando” corresponde ao período de tempo decorrido entre a resposta do sistema a uma requisição prévia e uma nova requisição de serviço. Uma fila de IS é usada para representar os terminais, visto que se assume não haver disputa por este dispositivo. Quando o usuário requisita um serviço ao sistema, ele deixa a fila de IS e prossegue para as filas dos dispositivos. Uma vez que o serviço é concluído, o usuário retorna a fila de IS.

Este modelo também pode ser utilizado para diferentes classes de usuários, ou seja, um modelo de redes multiclasse.

Redes multiclasse em forma de produtos possuem algoritmos computacionais eficientes para sua solução. O objetivo destes algoritmos é calcular as medidas de desempenho de

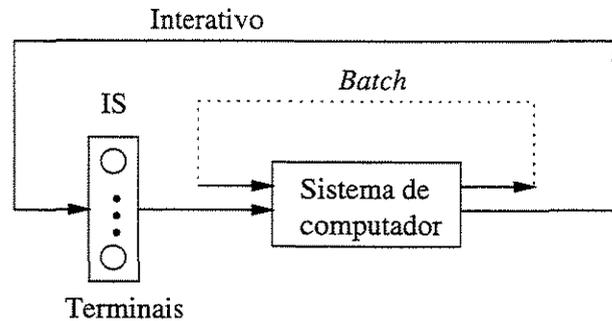


Figura 3.1: Modelo de servidor central.

uma rede em função do número de usuários no sistema. Os dois principais algoritmos são convolução e MVA (*Mean Value Analysis*) [51, 52]. Este último será apresentado a seguir.

3.1.2.1 MVA - *Mean Value Analysis*

O algoritmo *Mean Value Analysis* (MVA) envolve uma recursão que é computada diretamente em função das médias das medidas de desempenho e, em certas situações, das distribuições marginais de tamanho de fila. Este algoritmo foi originalmente desenvolvido com base no teorema de chegada. Este teorema determina simplesmente que um usuário que está chegando observa o sistema em equilíbrio com ele próprio removido.

A rede de filas fechada multiclasse com solução em forma de produtos, com N_d dispositivos e com \vec{k} usuários, é denotada $B(N_d, \vec{k})$. \vec{k} é o vetor de população, definido por $\vec{k} = (k_1, \dots, k_R)$, onde R é o número de classes. A seguinte notação será utilizada pelo algoritmo.

N_d	Número de dispositivos
R	Número de classes
K	População total no sistema
\vec{k}	Vetor de população
S_{ir}	Tempo de serviço médio dos usuários da classe r no dispositivo i
V_{ir}	Taxa média de visita (<i>visit ratio</i>) dos usuários da classe r no dispositivo i

$\mu_i(j)$	Taxa de serviço do servidor no dispositivo i , em função do número j de usuários em sua fila, em um dado instante de tempo
T_{ir}	Vazão da classe r no dispositivo i
W_{ir}	Tempo médio de espera, incluindo o tempo de serviço, dos usuários da classe r no dispositivo i
Q_{ir}	Número médio de usuários da classe r no dispositivo i
U_{ir}	Utilização do dispositivo i pelos usuários da classe r

Os parâmetros de entrada para o algoritmo são S_{ir} , V_{ir} e μ_i . As medidas de desempenho calculadas são T_{ir} , W_{ir} e Q_{ir} . Suponha que $P_i(j, \vec{k})$ denote a probabilidade de haver j usuários no dispositivo i na rede de filas $B(N_d, \vec{k})$. O seguinte conjunto de equações recursivas constituem o algoritmo MVA:

$$W_{ir}(\vec{k}) = \begin{cases} S_{ir} & \text{se o dispositivo } i \text{ for um IS} \\ S_{ir}[1 + \sum_{s=1}^R Q_{is}(\vec{k} - \vec{1}_r)] & \text{se o dispositivo } i \text{ possuir uma taxa} \\ & \text{de serviço constante} \\ \frac{S_{ir}}{\mu_i(j)} \sum_{j=1}^k P_i(j-1, \vec{k} - \vec{1}_r) & \text{se o dispositivo } i \text{ for dependente de carga} \end{cases} \quad (3.1)$$

$$T_{ir}(\vec{k}) = \frac{V_{ir} k_r}{\sum_{j=1}^{N_d} V_{jr} W_{jr}(\vec{k})}, \quad (3.2)$$

$$Q_{ir}(\vec{k}) = T_{ir}(\vec{k}) W_{ir}(\vec{k}), \quad (3.3)$$

$$P_i(j, \vec{k}) = \sum_{s=1}^R \frac{S_i}{\mu_i(j)} T_i(\vec{k}) P_i(j-1, \vec{k} - \vec{1}_s), \quad (3.4)$$

$$P_i(0, \vec{k}) = 1 - \sum_{j=1}^k P_i(j, \vec{k}) \quad (3.5)$$

onde o vetor $\vec{1}_s$ consiste de um 1 na s -ésima posição e zeros no resto do vetor,

$$\begin{aligned} i &= 1, \dots, N_d; & r &= 1, \dots, R; \\ j &= 1, \dots, k; & k &= \sum_{s=1}^R k_s. \end{aligned}$$

O algoritmo MVA permite calcular $W_{ir} = W_{ir}(K)$, $T_{ir} = T_{ir}(\vec{K})$ e $Q_{ir} = Q_{ir}(\vec{K})$, usando as equações acima descritas. As condições iniciais são $Q_{is}(\vec{0}) = 0$, $P_i(j, \vec{0}) = 0$ para $j > 0$ e $P_i(0, \vec{0}) = 1$. A recursão é aplicada sobre o vetor populacional \vec{k} , onde $\vec{0} \leq \vec{k} \leq \vec{K}$. Quando o dispositivo i possui uma função de taxa de serviço constante, também calcula-se:

$$U_{ir} = S_{ir} T_{ir}(\vec{K}). \quad (3.6)$$

Se não houver funções de taxa de serviço dependentes na rede, o algoritmo MVA não precisa calcular as distribuições marginais de tamanho de fila $P_i(j, \vec{k})$. Além disso, neste caso, o tempo requerido, em número de operações, para obter as medidas de desempenho, é aproximadamente

$$2R(N_d - 1) \prod_{r=1}^R (K_r + 1)$$

O espaço requerido, em número de posições de vetores é

$$N_d \prod_{r=1}^R (K_r + 1).$$

Quando há estados dependentes, o algoritmo MVA é complicado pela necessidade de calcular e armazenar as distribuições de tamanho de fila $P_i(j, \vec{k})$. O algoritmo pode numericamente falhar conforme $P_i(0, \vec{k})$ tende a zero. Reiser [53] desenvolveu meios de superar este problema em potencial, que implicam em um aumento nos custos computacionais.

3.2 Fórmula de Erlang

Erlang foi um matemático dinamarquês que no fim do século XIX iniciou o estudo de como uma grande população poderia ser atendida por um número limitado de servidores. Em sua homenagem, a medida de quantidade de tráfego é expressada em unidades de Erlangs.

Um Erlang representa a quantidade de tráfego transmitida por um canal completamente ocupado.

A fórmula Erlang B [54] é usada em sistemas onde não há fila para as requisições de serviço. Nestes sistemas, quando um usuário requisita o serviço, um canal é imediatamente atribuído a ele, se houver um canal disponível. Se não houver canais disponíveis, o usuário que fez a requisição é bloqueado sem acessar o sistema. O usuário não recebe o serviço, mas é livre para requisitá-lo outra vez. Assume-se que as chegadas de requisições obedecem à distribuição de Poisson. Além disso assume-se que há um número quase infinito de usuários, bem como o seguinte:

- Não há memória das requisições, o que implica que todos os usuários, incluindo os bloqueados, podem requisitar um canal a qualquer momento.
- Todos os canais livres estão totalmente disponíveis para servir requisições até que todos estejam ocupados.
- A duração da operação (chamado de tempo de serviço) é exponencialmente distribuída.
- Há um número finito de canais disponíveis.
- Requisições são descritas por uma distribuição de Poisson, o que implica em intervalos entre requisições exponencialmente distribuídos.
- Os intervalos entre chegadas de requisições são independentes uns dos outros.
- O número de canais ocupados é igual ao número de usuários ocupados.

Este modelo é conhecido como uma fila M/M/c, e leva a derivação da fórmula Erlang B, que fornece a probabilidade das requisições serem bloqueadas devido a congestionamento, chamada probabilidade de bloqueio. A fórmula Erlang B é dada por:

$$E_c = \frac{\frac{A^c}{c!}}{\sum_{k=0}^c \frac{A^k}{k!}} \quad (3.7)$$

onde c é o número de canais e A é a razão entre a taxa de chegada de requisições e a taxa de serviço do sistema.

A fórmula Erlang B é de fundamental importância em engenharia de tráfego e tem muitas aplicações no projeto de redes de telecomunicações. Devido a sua importância, é necessário calcular a equação 3.7 de forma rápida e precisa, para grandes valores de c . Observando a fórmula, é fácil notar que os termos fatoriais podem atrasar este cálculo. Através de uma pequena análise pode-se obter uma relação recursiva entre E_k e E_{k-1} . Por uma simples operação algébrica, usando a Equação 3.7, obtém-se:

$$\frac{E_k}{E_{k-1}} = \frac{A}{k}(1 - E_k) \quad (3.8)$$

Isolando E_k , chega-se a:

$$E_k = \frac{A E_{k-1}}{k + A E_{k-1}} \quad \text{para } k = 0, 1, 2, \dots, c \quad (3.9)$$

Quando $k = 0$ não há servidores disponíveis e portanto todos os usuários são bloqueados. Em outras palavras,

$$E_0 = 1. \quad (3.10)$$

As duas equações acima fornecem um algoritmo recursivo simples pelo qual a probabilidade de bloqueio, para grandes valores de c , pode ser calculada. Por exemplo, para resolver um problema de dimensionamento, no qual se deseja saber quantos servidores são necessários para se obter uma determinada probabilidade de bloqueio, basta incrementar o número de servidores e calcular cada probabilidade de bloqueio. Assim, o resultado é obtido quando a probabilidade de bloqueio desejada for atingida.

Capítulo 4

Projeto de Sistemas Interativos de Vídeo

Para o oferecimento de serviços de vídeo sob demanda em larga escala em redes de computadores são utilizadas técnicas baseadas em *multicast* (seção 2.4.3), para reduzir a grande demanda por banda passante destes serviços. Uma característica desejável em serviços de vídeo é a interatividade, que abrange a capacidade de efetuar operações de VCR. Sempre que um espectador solicita uma operação de VCR, seu fluxo de vídeo dessincroniza-se do fluxo de seu grupo de *multicast*. Faz-se preciso, portanto, o dimensionamento do número de canais necessários em sistemas VoD interativos para atender uma dada demanda.

Este capítulo tem como objetivo apresentar um modelo para determinar o número de canais de vídeo necessários em sistemas interativos. Um resumo de trabalhos anteriores é apresentado na seção 4.1. Na seção 4.2, o funcionamento de um sistema de VoD interativo é apresentado, sendo o cálculo do número de canais necessários para este sistema descrito na seção 4.3. Na seção 4.4 verifica-se a precisão do modelo aproximado. Na seção 4.5, é introduzido um modelo para o sistema de VoD interativo. Finalmente, os resultados numéricos são discutidos na seção 4.6.

4.1 Trabalhos Anteriores

Quando operações de VCR são efetuadas, é necessária uma quantidade maior de banda passante que aquela requerida no modo *playback*, a fim de manter a mesma qualidade da imagem. Entretanto, reservar banda passante extra para cada usuário admitido no sistema leva a um desperdício de banda passante. São diversos os esquemas para o suporte completo ou parcial a operações de VCR propostos na literatura.

Philip Yu *et al.* [33, 34] apresentam um mecanismo integrado com *Batching* em que usuários que realizam operações de *pause/resume* possuem garantias determinísticas, através do *caching* de fluxos, da retomada da apresentação; uma flexibilização para oferecer garantias probabilísticas também é considerada.

Li *et al.* [55] introduzem um esquema em que usuários que compartilham um fluxo de vídeo podem “destacar-se” deste fluxo em decorrência de uma operação de VCR. Em seguida, busca-se a resincronização de tais usuários através da utilização de *buffers* circulares, chamados de *buffers de sincronização*.

Towsley *et al.* [56] apresentam uma estratégia de *Batching* com oferecimento de operações VCR com reserva de canais para três conjuntos: um reservado para filmes populares, um de canais alocados sob demanda, para filmes não-populares, e um terceiro usado para tratar operações de VCR, chamados canais de contingência. O número de canais em cada um destes conjuntos é determinado levando-se em consideração a carga do sistema a longo prazo, isto é, flutuações instantâneas da carga do sistema não são consideradas. Para dimensionar o número de canais em cada conjunto, foi proposto um problema de otimização para minimizar a probabilidade de abandono dos usuários. O problema de minimização supõe que a distribuição do tempo de espera pode ser determinada por uma distribuição de tempo de espera de uma fila M/M/c. Mostrou-se que o uso de canais de contingência pode reduzir significativamente o número de canais necessários para suportar um determinado número de usuários, especialmente sob altas cargas e para longas durações da operação de PAUSE.

Em [57], Dey-Sircar *et al.* apresentam um esquema em que parte da banda passante do sistema é reservada para operações de VCR. Eles compararam um sistema no qual

banda passante extra é reservada para todo usuário admitido no sistema (garantia determinística) com um sistema no qual um conjunto de canais é compartilhado entre todos os usuários para executar operações de VCR (garantia estatística). Duas abordagens são consideradas para sistemas de VoD com garantias estatísticas. Na primeira, chamada de *Delay Scheme*, há limitação no número de usuários que compartilham a porção de banda passante utilizada para operações de VCR. Assim, quando não há banda passante disponível para efetuar operações de VCR, um usuário pode esperar ou liberar o canal, de forma que este possa ser reutilizado por outra requisição. Na segunda abordagem, denominada *Loss Scheme*, o usuário não tem que esperar e nem liberar um canal. Entretanto, as operações de VCR são manipuladas com uma qualidade de imagem inferior, uma vez que a banda passante de cada fluxo deve ser reduzida a fim de acomodar o novo usuário. Dey-Sircar *et al.* mostraram que um sistema com garantias estatísticas aceita um número maior de usuários que um sistema com garantias determinísticas.

4.2 Sistemas de VoD Interativo

Neste trabalho são analisados um sistema de VoD com *batching* e outro com *batching* e *piggybacking*. Em ambos os sistemas as requisições de exibição de vídeos não são imediatamente atendidas. Elas são agrupadas, e um único canal é alocado para um grupo de requisições, de acordo com uma política de *batching* específica. Além disso, em um sistema com *batching* e *piggybacking*, n fluxos de vídeo podem ser mesclados, liberando $n - 1$ canais, de acordo com o critério de *piggybacking* adotado.

Sempre que um usuário requisita uma operação de VCR, sua exibição se dessincroniza com a exibição de seu grupo *multicast* e, conseqüentemente, um canal dedicado deve ser alocado para ele. Dois esquemas são considerados. No primeiro esquema, canais são reservados para dar suporte a realização de operações de VCR. Estes canais formam um conjunto de contingência. Sempre que um grupo de usuários é admitido no sistema, o número de canais do conjunto de contingência é calculado, de forma que possa acomodar futuras requisições de operações de VCR. Se não houver canais disponíveis no conjunto de contingência, um canal usado para *playback* pode ser alocado. O canal alocado é retido

até o fim da exibição, ou até ocorrer uma mesclagem com outro fluxo.

No segundo esquema, não há reserva de canais para o suporte de fluxos de vídeo dessincronizados. Uma requisição de canal para realizar operações de VCR compete com requisições de admissão de novos grupos de usuários. Em ambos os esquemas, se o usuário que requisitou uma operação de VCR é o único usando o canal de vídeo, não há, obviamente, necessidade de se alocar um canal extra.

Nos dois esquemas, se não há canais disponíveis, a requisição é recusada, e o usuário permanece em seu grupo *multicast*. Uma abordagem ortogonal poderia ser retardar a admissão da requisição. A banda passante alocada por um fluxo de vídeo para operações de VCR é a mesma alocada no modo de *playback*, isto é, quando REW e FF são executadas, a qualidade da imagem é reduzida.

4.3 Avaliando a Demanda de Canais

Para assegurar o adequado funcionamento de um sistema de VoD interativo, o número de canais do conjunto de contingência deve ser dimensionado, de tal forma que apenas um pequeno número de requisições de operações de VCR sejam rejeitadas. Por isso, sempre que um usuário, ou grupo de usuários, é aceito no sistema, ou deixa o sistema, o número de canais do conjunto de contingência muda. Um usuário ou grupo de usuários deve ser aceito no sistema, se e somente se existir uma provisão de canais capaz de suportar o potencial número de operações de VCR a serem requisitadas.

Assume-se que a chegada de requisição de exibição de vídeo segue um processo de Poisson, e que os filmes são escolhidos de acordo com uma distribuição de Zip. Operações de VCR não são efetuadas por todos os usuários. Apenas aqueles que optem por estes serviços são permitidos emitir requisições de operações de VCR. O intervalo de tempo entre requisições de um mesmo usuário é exponencialmente distribuído, bem como a duração das operações de VCR. Estas suposições são fundamentadas em dados reais coletados em um sistema operacional [49].

A análise de um sistema com *batching* difere substancialmente da análise de um sistema com *batching* e *piggybacking*.

4.3.1 Um Sistema com *Batching*

Em um sistema com *batching* sempre que um canal é alocado a um fluxo dessincronizado, ele é retido, até o fim da exibição, uma vez que não há como ressincronizar este fluxo de vídeo com outros fluxos. Portanto, dimensionar o número de canais do conjunto de contingência é uma tarefa simples. Toda vez que um grupo de n usuários é aceito no sistema, $n \times p_{u_{vcr}}$ canais devem ser reservados para tratar os fluxos dessincronizados, sendo que $p_{u_{vcr}}$ é a probabilidade de um usuário optar por serviços interativos.

4.3.2 Um Sistema com *Batching* e *Piggybacking*

Em um sistema com *batching* e *piggybacking*, após a execução de uma operação de VCR, o fluxo de vídeo pode se sincronizar com outro, permitindo a liberação do canal de vídeo. O tempo de retenção (*holding time*) de um canal de contingência inclui não só o tempo de execução da operação de VCR, mas também o tempo para se ressincronizar com outro fluxo.

O número de usuários no sistema altera-se nas admissões de grupos de usuários, e nos termos de exibições de vídeo. Portanto, entre a ocorrência de quaisquer dois destes eventos, o número de usuários permanece fixo. Assim, o sistema pode ser modelado por uma rede de filas fechada, ou seja, por um modelo de servidor central (seção 3.1). No modelo de servidor central, o tempo em que usuários ficam no modo *playback* corresponde ao tempo de serviço de um servidor infinito (*Infinite Server*). Após visitar o servidor infinito, os usuários vão para uma fila com múltiplos servidores, isto é, uma fila M/M/c (um servidor dependente da carga - *load dependent server*). O tempo de serviço desta fila corresponde ao tempo de retenção de um canal de contingência. Ele inclui o tempo para ressincronizar com outro fluxo, e o tempo médio de execução de uma operação de VCR, ponderados pela probabilidade da operação de VCR (PAUSE, REW ou FF). O número de servidores, c , é, na realidade, o tamanho do conjunto de contingência. O ponto chave é dimensionar c , de modo que nenhum usuário espere para ser servido. Sempre que um usuário achar todos os servidores ocupados, uma operação de VCR é rejeitada. Assim sendo, c tem que ser calculado de modo que leve a um baixo número de rejeições.

Para determinar o número de canais do conjunto de contingência, um algoritmo de redes de filas é executado com um certo valor de c , e o tamanho da fila do servidor dependente de carga é verificado. Este processo continua, até se encontrar um valor de c tal que o tamanho médio da fila no servidor dependente da carga seja muito pequeno. O modelo de servidor central é um modelo exato para o sistema de VoD interativo. No entanto, os algoritmos análise do valor médio (*Mean Value Analysis* - MVA) e de convolução 3.1.2.1, que são algoritmos exatos para redes de filas fechadas, apresentam instabilidades numéricas, quando o tamanho da fila nos servidores dependente da carga é muito pequeno. Alternativamente, o algoritmo de convolução normalizada [53] pode ser usado para superar tal instabilidade numérica. Entretanto, este algoritmo é também instável, quando o tempo de serviço da fila dependente de carga é uma parte significativa do tempo de ciclo (*cycle time*), ou seja, o tempo para visitar o servidor infinito e a fila dependente de carga. Os algoritmos mencionados foram utilizados para resolver o problema proposto, e a instabilidade numérica foi verificada, pois neste problema o tempo de serviço da fila dependente de carga compõe a maior parte do tempo de ciclo. Por exemplo, um usuário que permanece no estado *playback* por 30 minutos e então requisita uma operação de PAUSE por 5 minutos terá um tempo de ressincronização de aproximadamente 50 minutos, ou seja, tempo de serviço de 55 minutos, sendo que o ciclo foi de 85 minutos.

Dada a instabilidade numérica dos algoritmos exatos para redes de filas fechadas, adotou-se um modelo aberto aproximado. O sistema de VoD é modelado por uma fila Erlang B (seção 3.2). O servidor dependente de carga, nas redes de filas fechadas, corresponde a uma fila M/M/c sem fila de espera, no modelo aberto, isto é, requisições são perdidas se não houver servidor disponível. A taxa de chegada para o servidor dependente de carga, no modelo fechado, é calculada implicitamente. No entanto, no modelo aberto, a taxa média de chegada na fila M/M/c precisa ser aproximada, para ser utilizada na equação 3.7, como mostra a Figura 4.1. A aproximação é explicada a seguir.

Um usuário pode estar em dois modos: *playback* ou VCR. No modo VCR, o usuário retém um canal de contingência e, no modo *playback*, ele é parte de um grupo *multicast*.

A taxa média de chegada de requisições VCR, isto é, a taxa média de chegada na fila M/M/c é a taxa de requisições de VCR por usuário multiplicada pelo número médio de

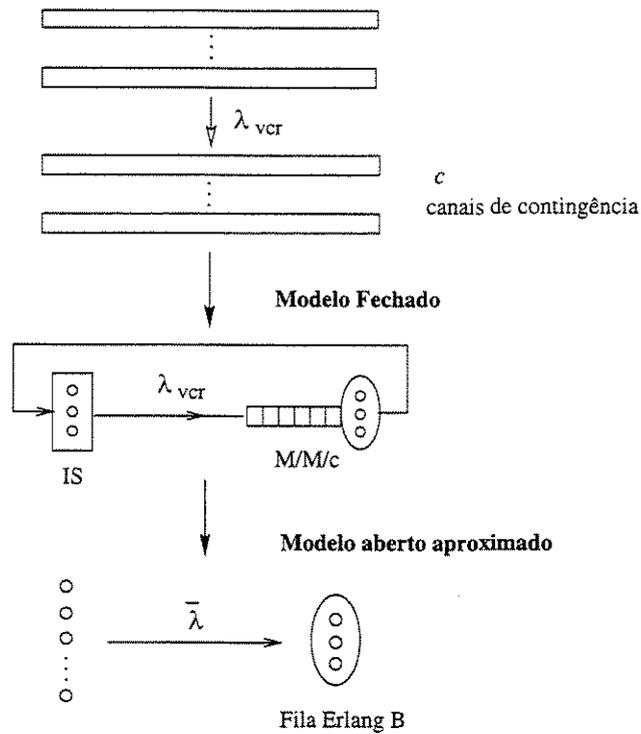


Figura 4.1: Aproximação do modelo de redes de filas fechadas para o modelo de filas aberto.

usuários que efetuam operações de VCR.

$$\bar{\lambda} = N \lambda_{vcr} P_{\text{playback}} \quad (4.1)$$

onde:

- N Número de usuários que executam operações de VCR;
- λ_{vcr} Taxa de requisições de VCR por usuário;
- P_{playback} Probabilidade de um usuário estar em modo de *playback*. Note que $P_{\text{playback}} = 1 - P_{vcr}$.

A probabilidade de estar em modo VCR, P_{vcr} , é a fração do tempo que um usuário retém um canal de contingência durante toda a exibição. A duração da exibição é a duração original do filme adicionada do tempo gasto efetuando operações de VCR. Além

disso, o tempo de retenção médio de um canal de contingência inclui o tempo médio de duração de uma operação de VCR e o tempo médio de resincronização. P_{vcr} é então dado por:

$$P_{vcr} = \frac{N_{vcr} D}{T + N_{vcr} t_{op}} \quad (4.2)$$

onde:

- N_{vcr} Número médio de operações de VCR efetuadas por um usuário;
- D Tempo de retenção médio de um canal de contingência;
- t_{op} Duração média de uma operação de VCR;
- T Duração original do filme em segundos.

O tempo de retenção médio de um canal de contingência inclui o tempo médio de resincronização com outro fluxo. Para simplificar o cálculo do valor de D , assume-se que o fluxo dessincronizado pode apenas mesclar-se com seu fluxo original. De fato, um fluxo pode mesclar-se com qualquer outro fluxo que esteja exibindo o mesmo filme. O valor calculado é, então, um limite superior para o valor de D , dado que o fluxo dessincronizado pode se mesclar com outro fluxo que esteja mais próximo do que seu fluxo original. O tempo de resincronização depende do tipo de operação de VCR, de sua duração, e da posição do quadro na qual a operação de VCR foi requisitada. Além disso, a posição na qual a operação de VCR foi requisitada depende do número de operações realizadas durante a exibição do filme. D é, então, dado por:

$$D = \sum_{n=1}^{\infty} d(n) p(n) \quad (4.3)$$

onde:

- $d(n)$ Tempo de retenção médio de um canal de contingência dado que n operações são executadas durante a exibição do vídeo;
- $p(n)$ Probabilidade de um usuário requisitar n operações de VCR durante a exibição do filme.

$p(n)$ e $d(n)$ são dados respectivamente por

$$p(n) = \frac{(\lambda_{vcr} T)^n}{n!} e^{(-\lambda_{vcr} T)} \quad (4.4)$$

$$d(n) = \frac{1}{n} \sum_{i=1}^n \sum_{op} \sum_{i+1}^{L-i} d_{op}(s) \lambda_{vcr} \frac{\lambda_{vcr}^{(i-1)}}{(i-1)!} e^{-\lambda_{vcr}} P_{op} \quad (4.5)$$

onde:

- λ_{vcr} Taxa de requisições de operações de VCR de um usuário;
- $d_{op}(s)$ Tempo de retenção do canal de contingência de uma operação de VCR op que ocorreu no s -ésimo quadro;
- P_{op} Probabilidade do tipo da operação de VCR op ;
- op Operação de VCR (PAUSE, FF, REW);
- L é o número de quadros de um vídeo.

$\lambda_{vcr} \frac{\lambda_{vcr}^{(i-1)}}{(i-1)!} e^{-\lambda_{vcr}}$ Probabilidade de que a i -ésima operação seja requisitada no s -ésimo quadro.

Cada operação de VCR possui uma duração média própria. Desta forma, $d_{op}(s)$ deve ser computado como uma função do tipo da operação. Portanto

$$d_{op}(s) = \int_0^{\text{Max}_{op}(s)} G_{op}(s, t) F_{op}(t) dt \quad (4.6)$$

onde:

- $\text{Max}_{op}(s)$ Duração máxima da operação op que ocorreu no quadro s ;
- $F_{op}(t)$ Densidade de probabilidade da duração da operação op ;
- $G_{op}(s, t)$ Duração do tempo de retenção de um canal de contingência de uma operação de VCR que ocorreu no quadro s e durou t segundos.

G_{op} é dado por:

$$G_{Pause}(s, t) = \begin{cases} t + t \times A & \text{se } t \times A \leq \frac{L-s}{V_{max}} \\ t + \frac{L-s}{V_{max}} & \text{se } t \times A > \frac{L-s}{V_{max}} \end{cases} \quad (4.7)$$

$$G_{Rew}(s, t) = \begin{cases} t + (t + t R_{Rew}) \times A & \text{se } (t + t R_{Rew}) \times A \leq \frac{L - (s - t V_{Rew})}{V_{max}} \\ t + \frac{L - (s - t V_{Rew})}{V_{max}} & \text{se } (t + t R_{Rew}) \times A > \frac{L - (s - t V_{Rew})}{V_{max}} \end{cases}$$

$$G_{FF}(s, t) = \begin{cases} t + (t R_{FF} - t) \times A & \text{se } (t R_{FF} - t) \times A \leq \frac{L - (s + t V_{FF})}{V_{max}} \\ t + \frac{L - (s + t V_{FF})}{V_{max}} & \text{se } (t R_{FF} - t) \times A > \frac{L - (s + t V_{FF})}{V_{max}} \end{cases}$$

onde:

$A = \frac{V_n}{V_{max} - V_{min}}$ Constante que, quando multiplicada pela duração da operação de VCR, produz o tempo para se mesclar ao seu fluxo original;

V_{max} Taxa máxima de exibição;

V_{min} Taxa mínima de exibição;

V_n Taxa normal de exibição em playback;

V_{Rew} Taxa de exibição quando efetua-se a operação de retrocesso (REW),

$$V_{Rew} = R_{Rew} V_n;$$

V_{FF} Taxa de exibição quando efetua-se a operação de avanço (FF), $V_{FF} =$

$$R_{FF} V_n;$$

R_{Rew} Constante de aumento da velocidade na operação REW;

R_{FF} Constante de aumento da velocidade na operação FF;

t Duração da operação de VCR.

Note que $\text{Max}_{op}(s)$ possui diferentes valores para cada tipo de operação. Estes são apresentados a seguir.

$$\text{Max}_{PAUSE}(s) = \infty$$

$$\text{Max}_{Rew}(s) = s/V_{Rew}$$

$$\text{Max}_{FF}(s) = (L - s)/V_{FF}$$

Assim, $d_{op}(s)$ é dado por:

$$d_{Pause}(s) = (1 + A) \int_0^{L'/A} t e^{\lambda_{Pause}} dt + \int_{L'/A}^{\infty} (t + (L')) e^{\lambda_{Pause}} dt$$

$$d_{Rew}(s) = (1 + A \cdot e_{Rew}) \int_0^{\text{Min}(L'/(A \cdot e_{Rew} - B_{Rew}), s/V_{Rew})} t e^{\lambda_{Rew}} dt +$$

$$\int_{\text{Min}(L'/(A \cdot e_{Rew} - B_{Rew}), s/V_{Rew})}^{s/V_{Rew}} ((1 + B_{Rew})t + (L')) e^{\lambda_{Rew}} dt \quad (4.8)$$

$$d_{FF}(s) = (1 + A \cdot e_{FF}) \int_0^{L'/(A \cdot e_{FF} - B_{FF})} t e^{\lambda_{FF}} dt +$$

$$\int_{L'/(A \cdot e_{FF} - B_{FF})}^{(L-s)/V_{FF}} ((1 + B_{FF})t + (L')) e^{\lambda_{FF}} dt$$

onde:

$$L' = \frac{L - s}{V_{max}};$$

$$B_{Rew} = \frac{V_{Rew}}{V_{max}}, \quad e_{Rew} = 1 + R_{Rew};$$

$$B_{FF} = \frac{V_{FF}}{V_{max}}, \quad e_{FF} = 1 + R_{FF}.$$

4.4 Precisão do Modelo Aproximado

Para avaliar a precisão do modelo aproximado, o número estimado de canais demandados foi confrontado com resultados derivados via simulação. Ao invés de simular todo o sistema, apenas a chegada de requisições de operações de VCR ao conjunto de contingência foi simulada, visto que isto é o que realmente influencia o valor estimado. Para variar $\bar{\lambda}$, que representa a taxa média de chegada de requisições de operações de VCR, diferentes valores de N e λ_{vcr} foram escolhidos. Desde que $\lambda_{vcr} = N_{vcr}/T$, diversos valores de λ_{vcr}

foram obtidos através da variação de N_{vcr} . Variou-se N no intervalo $[50, 2000]$ e N_{vcr} em $[1,5]$. Diferentes valores de P_{playback} foram obtidos alterando a duração média das operações de VCR, isto é, a média de $F_{op}(t)$.

O número de canais necessários estimado pelo modelo aproximado é um limite superior do número médio de canais demandados, e é um limite inferior do número máximo de canais demandados.

Assume-se que um fluxo de vídeo que retém um canal de contingência pode apenas se ressincronizar ao fluxo associado ao seu grupo *multicast* original. Esta suposição superestima o tempo médio de retenção dos canais de contingência e, conseqüentemente, leva a uma estimativa conservadora do número requerido de canais de contingência.

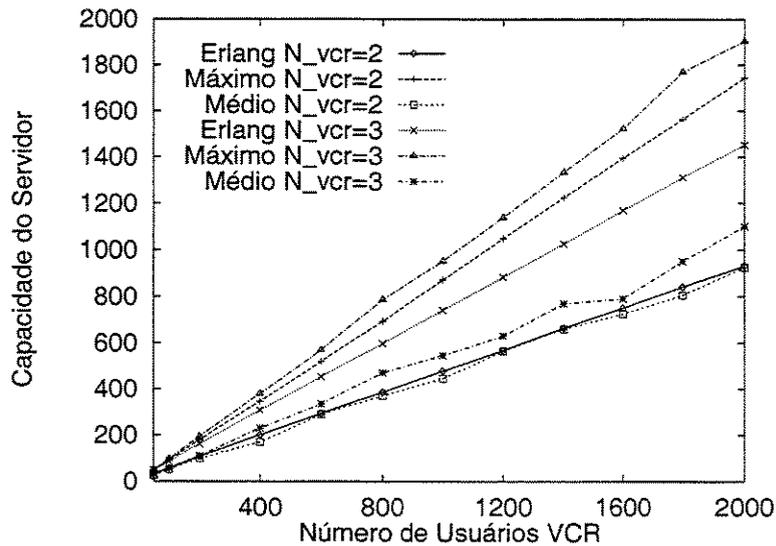


Figura 4.2: Comparação entre número de canais de contingência estimado e o número médio e máximo de canais obtidos via simulação.

A Figura 4.2 ilustra o número de canais estimado, o número médio e o número máximo de canais obtidos via simulação, em função de N , para diferentes valores de N_{vcr} . N_{vcr} é o parâmetro de maior influência para que o valor estimado seja mais próximo ao valor médio ou ao valor máximo. Quanto maior o valor de N_{vcr} , mais próximo do valor máximo está o valor estimado. Esta tendência pode ser entendida pelo fato de que quanto maior o número de visitas ao conjunto de contingência, maior é a possibilidade da requisição

de operação de VCR ser emitida em uma posição próxima ao final do vídeo, o que não permite futuras mesclagens com o fluxo *multicast* original. Conseqüentemente, maior é o tempo de retenção médio de um canal de contingência.

A fórmula de Erlang é calculada sempre que um grupo de usuários é admitido no sistema, ou seja, ela tem que ser calculada em tempo real. A tabela 4.1 apresenta o tempo médio para se calcular a fórmula de Erlang, para diferentes valores de N e N_{vcr} . Estes tempos foram obtidos em uma máquina Sparc com sistema operacional SunOS e 512 Mb de memória.

N° Usuários VCR	Tempo Médio de Execução (μs)			
	N° de Operações de VCR por filme			
	1	2	3	4
50	5	10	10	10
100	10	15	20	30
400	20	50	70	90
1000	45	110	170	220
1600	70	180	260	350
2000	100	220	330	420

Tabela 4.1: Tempo médio do cálculo da fórmula de Erlang, em microssegundos

4.5 Descrição Simulador

Para se comparar diferentes sistemas de VoD, um simulador de um sistema de VoD interativo foi desenvolvido. Os parâmetros de desempenho nos quais se tem interesse são *i*) o número de usuários admitidos no sistema, *ii*) a probabilidade de abandono, isto é, a razão entre o número de usuários que desistem de assistir ao filme e o número total de requisições que chegaram no sistema, e *iii*) a porcentagem de operações VCR rejeitadas. Provedores de serviço de VoD almejam aumentar o número de usuários admitidos no sistema, isto é, o número de requisições de exibição de filme aceitas. A taxa de abandono traduz o prejuízo causado pela não admissão de requisições de exibição de vídeo. A porcentagem

de operações de VCR rejeitadas representa a porcentagem de requisições de operações de VCR que são recusadas devido a falta de canal para o seu suporte. Ela representa o grau de quebra de contrato entre usuários e provedor de serviço, e é uma medida da qualidade do serviço fornecido.

Chegadas de requisições seguem um processo de Poisson com média λ . Ao chegar uma requisição, o vídeo associado a ela é escolhido de acordo com a distribuição de Zip, e a requisição entra na fila específica. A distribuição de Zip modela de forma precisa a preferência de usuários em locadoras de vídeo. De tempos em tempos, uma decisão é tomada sobre a qual filme um canal deve ser alocado, de acordo com a política de *batching* adotada. Uma vez que nem todos os usuários emitem requisições de operações de VCR, quando um grupo de usuários é aceito no sistema, isto é, um canal é alocado a eles, o número de usuários naquele grupo que efetuam operações de VCR é determinado aleatoriamente. A probabilidade de um usuário executar operações de VCR é dado por $p_{u_{vcr}}$. O número médio de operações de VCR efetuadas por um usuário durante a exibição de um vídeo é dado por N_{vcr} operações. O intervalo de tempo entre requisições de operações de VCR é obtido de uma distribuição exponencial com média dada pela duração do vídeo dividida pelo número médio de operações efetuadas por um usuário, N_{vcr} . O tipo da operação de VCR é determinado por P_{op} e sua duração é exponencialmente distribuída com média λ_{op} , onde *op* pode ser REW, FF ou PAUSE.

A política de *batching* usada no experimento de simulação é a *Look-ahead-maximize-batch* (LAMB), vista na seção 2.4.3.1, e a política de *piggybacking* é a par-ímpar, seção 2.4.3.2. LAMB foi escolhida porque admite o maior número de usuários quando comparada a outras políticas de *batching*. Quanto maior o número de usuários admitidos no sistema, para um valor fixo de $p_{u_{vcr}}$, maior será o número de usuários que requisitam operações de VCR. Conseqüentemente, uma avaliação mais precisa da abordagem proposta é obtida.

Sob LAMB, pontos de escalonamento são definidos pelo tempo de abandono de usuários. LAMB maximiza o número de usuários admitidos em uma janela de tempo definida pelo tempo de escalonamento e o tempo de abandono mais longe de um usuário no sistema. LAMB leva em consideração todas as liberações de canais dentro desta janela de

tempo. Quando a decisão de aceitar um novo grupo de usuários é tomada, o número de canais extras necessários para suportar a demanda por operações de VCR deste novo grupo de usuários é calculado, usando a aproximação desenvolvida na seção 4.3. Então, a disponibilidade de tal número de canais é verificada. Se o número de canais disponíveis não é suficiente para suportar o novo grupo, este não é admitido no sistema neste momento. Caso contrário, o número de canais do conjunto de contingência é ajustado para o valor corrente, adicionado dos canais extras exigidos pelo novo grupo.

Adotou-se a política de *piggybacking* par-ímpar, que mescla fluxos em pares. Algumas políticas de *piggybacking*, como *Snapshot* e *S2* (seção 2.4.3.2), produzem um menor número de quadros exibidos do que a política adotada, entretanto assumem um processo de chegada de Poisson, suposição que não é válida em um sistema sob *batching*. As políticas par-ímpar e mesclagem simples produzem os mesmos resultados quando usadas em conjunto com LAMB [2].

4.6 Resultados Numéricos

Para avaliar o desempenho do sistema de VoD, utilizou-se simulação de eventos discretos. O método de replicação independente foi utilizado para derivar intervalos de confiança com 99% de nível de confiança. O número de replicações para cada ponto das curvas exibidas nesta seção foi tal que a largura dos intervalos de confiança é de no máximo 8% do valor médio. Intervalos de confiança não são exibidos nos gráficos para uma melhor interpretação visual dos resultados. Sistemas de VoD com *batching* e sistemas com *batching* e *piggybacking* foram estudados. A eficiência da adoção de canais de contingência é também investigada.

Foram realizadas simulações para variadas configurações, graus de interatividade e sob diferentes cargas. Os resultados são apresentados em função da capacidade do sistema, isto é, número de canais disponíveis para transmitir fluxos de vídeo. O impacto do tamanho do servidor, ou seja, o número de vídeos armazenados, no desempenho é também avaliado. Para uma carga fixa, variações na taxa de requisições de operações de VCR foram obtidas através de alterações no grau de interatividade, isto é, na fração de usuários que fazem

operações de VCR, p_{vcr} . O valor desejado para a probabilidade de abandono, bem como para a rejeição de operações de VCR, é 1%.

Inicialmente são apresentados os resultados para servidores pequenos, com 100 filmes armazenados com duas horas de duração cada. São mostrados resultados para baixas cargas de 10 requisições de vídeo por minuto, e para cargas elevadas de 60 requisições de vídeo por minuto. A probabilidade de uma requisição de PAUSE é 0.5, e sua duração média é 5 minutos. As probabilidades de FF e REW são ambas de 0.25 e sua duração média é 30 segundos. O número médio de operações de VCR por usuário é 2. A sensibilidade do desempenho do sistema ao comportamento do usuário é descrita ao final desta seção.

Sempre que um grupo de usuários é admitido no sistema, o número de canais necessários para operações de VCR é calculado utilizando o modelo aproximado introduzido na seção 4.3.

4.6.1 Sistema de VoD com *Batching*

As Figuras 4.3, 4.4 e 4.5 mostram, respectivamente, o número de usuários admitidos no sistema, a probabilidade de abandono e a porcentagem de operações de VCR rejeitadas, em função da capacidade do servidor. Cada uma delas exibe diferentes graus de interatividade, para um servidor pequeno sob baixas cargas. Em todas as figuras aqui exibidas, CC e U denotam “com canais de contingência” e grau de interatividade, respectivamente.

O número de usuários admitidos no sistema é sempre maior para um servidor sem o conjunto de canais de contingência do que para servidores com este conjunto. Para um grau de interatividade de 10%, não há diferença entre os sistemas com e sem conjunto de contingência. Entretanto, conforme o grau de interatividade aumenta, a diferença entre o número de usuários admitidos em ambos os sistemas aumenta. Para um grau de 80%, e para um sistema com capacidade de 1000 canais, o número de usuários admitidos no sistema sem conjunto de contingência é maior que o dobro do número de usuários admitidos em um sistema com conjunto de contingência. Esta diferença diminui conforme a capacidade do sistema aumenta, pois o número de usuários admitidos no sistema converge para seu valor máximo, para uma carga fixa. Em um sistema com conjunto de contingência,

canais permanecem ociosos para suportar operações de VCR. Tais canais são utilizados para admitir mais usuários no sistema sem canais de contingência.

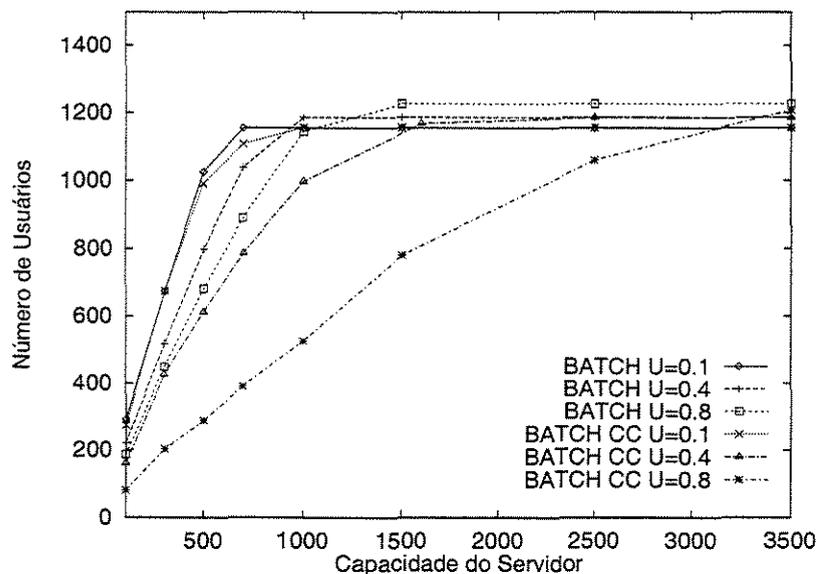


Figura 4.3: N° de Usuários Admitidos \times Capacidade do Servidor, taxa de 10 req/min.

Note que o número de usuários admitidos no sistema não cresce significativamente apesar do aumento da capacidade do sistema. Este efeito de limite superior aparece em todos os tipos de sistemas investigado neste estudo, e pode ser atribuído ao comportamento regulador das políticas de *batching*. Sob *batching*, os usuários são admitidos de maneira discreta, e não de forma contínua. Usuários são admitidos de tempos em tempos, e o tempo entre admissões é determinado pela política de *batching* adotada. Então, não é vantajoso aumentar a disponibilidade de canais além de um certo valor já que a demanda por novos canais é regulada pelo intervalo entre admissões da política de *batching*. Por exemplo, em um sistema com canais de contingência e com grau de interatividade de 40%, o número de usuários admitidos permanece constante após 1600 canais disponíveis.

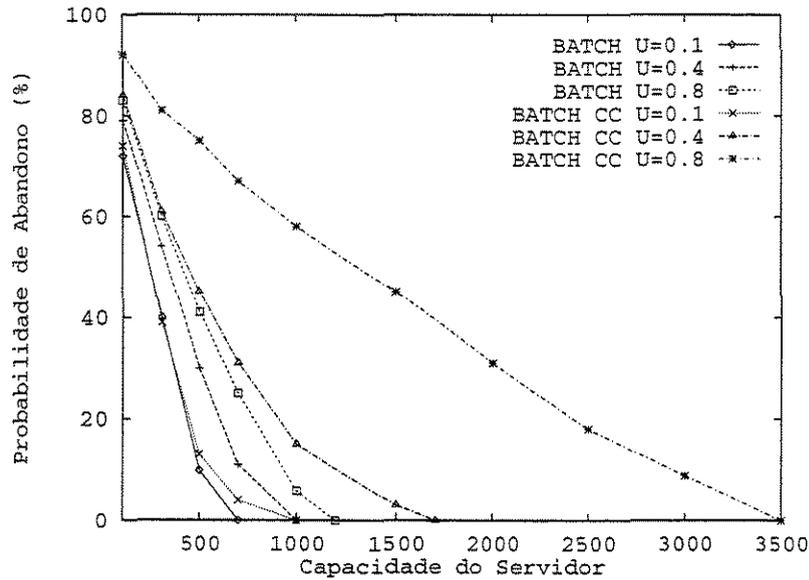


Figura 4.4: Probabilidade de Abandono \times Capacidade do Servidor, taxa de 10 req/min.

A probabilidade de abandono para sistemas sem conjunto de contingência é sempre menor que para sistemas com conjunto de contingência (Figura 4.4). A diferença entre o número de canais necessários para suportar uma probabilidade de abandono de 1%, por um sistema com conjunto de contingência e por um sistema sem este conjunto, aumenta conforme o grau de interatividade aumenta. Para um baixo grau de interatividade (10%), esta diferença é de 40%, enquanto que, para um grau de interatividade de 80%, o número de canais demandados por um sistema com conjunto de contingência é o dobro do número de canais demandados por um sistema sem conjunto de contingência.

O número de operações de VCR rejeitadas em um sistema com canais de contingência está abaixo de 1%, para uma capacidade de sistema acima de 300 canais. Por outro lado, para um grau de interatividade de 10%, em um sistema sem canais de contingência, tal porcentagem é atingida com uma capacidade de sistema de 700 canais. Para um grau de interatividade de 80%, esta porcentagem é obtida para uma capacidade de sistema de 1300 canais. Apesar disso, sistemas com canais de contingência não são preferíveis sobre sistemas sem canais de contingência. Note que, para graus de interatividade baixos (10%), em um sistema sem conjunto de contingência, 700 canais são necessários para prover uma

porcentagem de requisições de VCR rejeitadas de 1%, enquanto que em um sistema com conjunto de contingência, 1000 canais são necessários para obter uma probabilidade de abandono de 1%. Em outras palavras, os valores de desempenho desejados são obtidos com uma menor capacidade em sistemas sem conjunto de contingência do que em sistemas com conjunto de contingência. Para elevados graus de interatividade (80%), em um sistema sem canais de contingência, a porcentagem de requisições de VCR rejeitadas, bem como a probabilidade de abandono, estão abaixo dos valores alvo para uma capacidade de sistema de 1400 canais, enquanto que em um sistema com conjunto de contingência, a probabilidade de abandono está abaixo do valor alvo para uma capacidade de sistema de 3500 canais. Portanto, para elevados graus de interatividade, sistemas sem canais de contingência possuem um desempenho melhor que sistemas com canais de contingência.

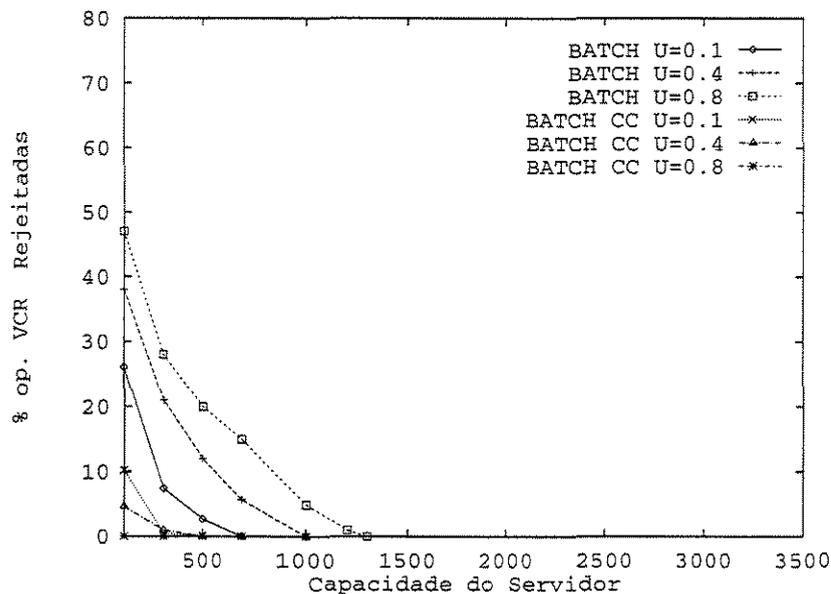


Figura 4.5: Porcentagem de operações de VCR Rejeitadas \times Capacidade do Servidor, taxa de 10 req/min.

Sob altas cargas (60 requisições por minuto), a diferença entre o número de usuários admitidos em um sistema sem conjunto de contingência e em um sistema com conjunto de contingência é superior a diferença sob baixas cargas, como ilustrado na Figura 4.6. Por exemplo, para 1000 canais e grau de interatividade de 40%, a diferença é de 200 usuários

sob baixas cargas, e de 900 usuários sob altas cargas. A diferença aumenta conforme o grau de interatividade aumenta. Por exemplo, para 1000 canais e 10% de interatividade, a diferença é 300 usuários, enquanto que, para 80% de interatividade, ela é 1100 usuários.

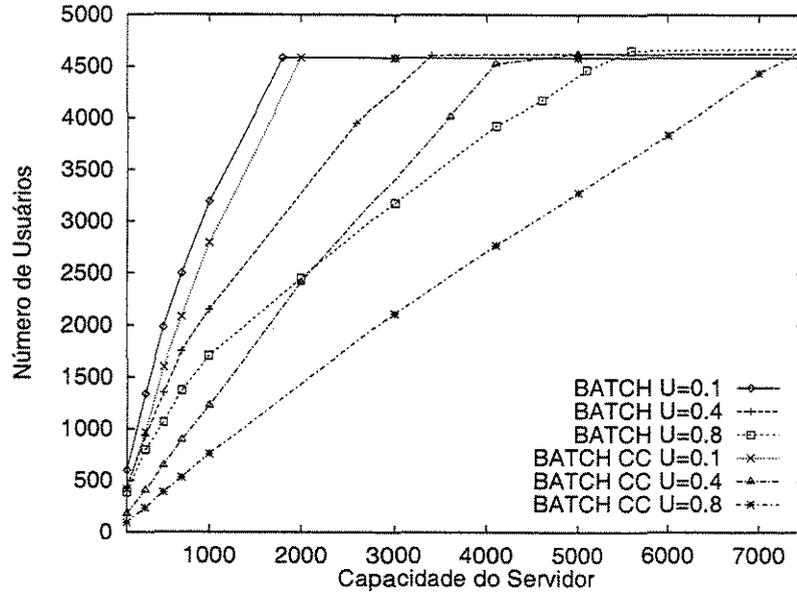


Figura 4.6: N° de Usuários Admitidos × Capacidade do Servidor, taxa de 60 req/min.

Para atingir a meta de probabilidade de abandono, um maior número de canais é necessário que sob baixas cargas, como pode ser visto na Figura 4.7. Por exemplo, para um sistema com canais de contingência e 10% de interatividade, são necessários 1000 e 2000 canais sob baixas e sob altas cargas, respectivamente. A probabilidade de abandono para sistemas com conjunto de contingência é maior que para sistemas sem conjunto de contingência, sem levar em consideração o grau de interatividade. No entanto, a diferença entre as probabilidades de abandono é proporcionalmente menor que esta diferença sob baixas cargas.

Sob altas cargas, o valor alvo para a porcentagem de operações de VCR rejeitadas é atingido com uma capacidade de sistema três vezes maior que a capacidade necessária sob baixas cargas, para um sistema sem canais de contingência (Figura 4.8). Por exemplo, para 80% de interatividade, o valor alvo é atingido com 1250 canais sob baixas cargas,

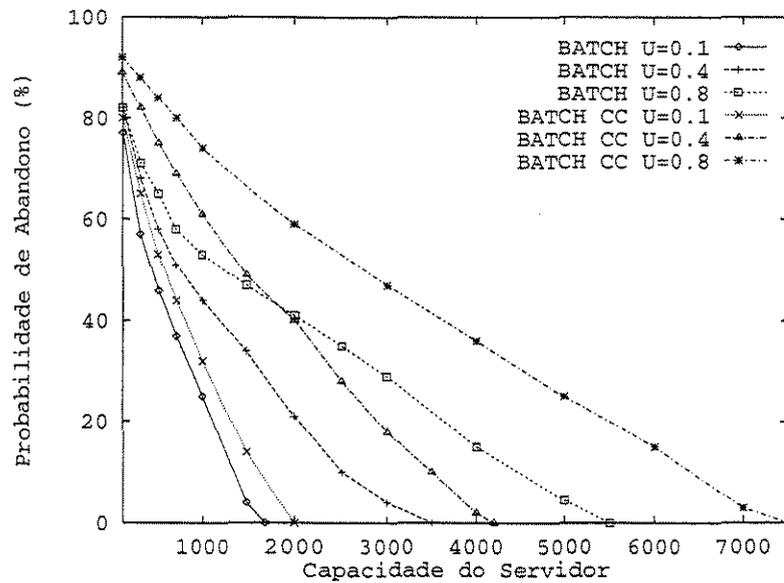


Figura 4.7: Probabilidade de Abandono \times Capacidade do Servidor, taxa de 60 req/min.

enquanto que sob altas cargas, são necessários 5600 canais.

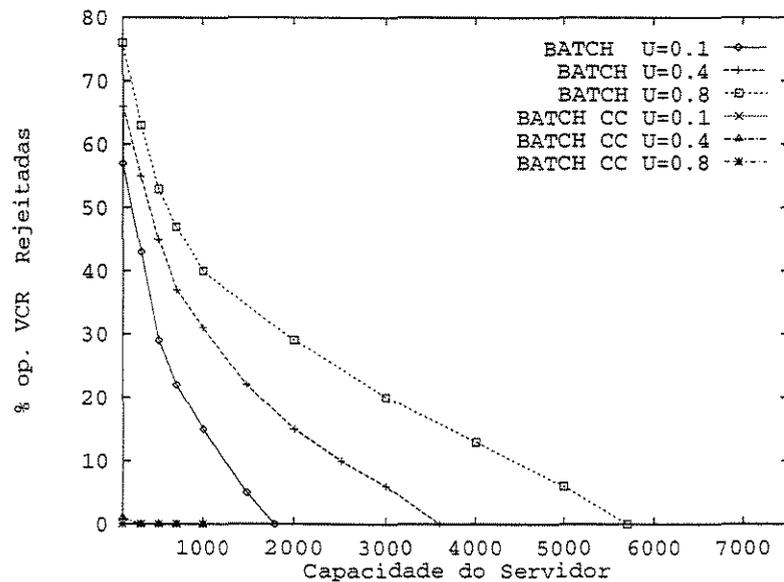


Figura 4.8: Porcentagem de operações de VCR Rejeitadas \times Capacidade do Servidor, taxa de 60 req/min.

Embora um maior número de canais sejam demandados para se atingir a porcentagem alvo de requisições de operações de VCR rejeitadas para um sistema sem canais de contingência (Figura 4.8), o número de canais exigidos por um sistema com canais de contingência para atingir a probabilidade alvo de abandono (Figura 4.7) é maior que o número de canais para atingir os valores alvos de desempenho por um sistema sem canais de contingência. Por exemplo, para um grau de 40% de interatividade, um sistema sem canais de contingência necessita de 3500 canais para atingir suas metas, enquanto que um sistema com canais de contingência precisa de 4100 canais.

As Figuras 4.9 e 4.10 apresentam, respectivamente, a capacidade do servidor necessária para obter a probabilidade de abandono e a porcentagem de operações de VCR rejeitadas abaixo de 1%, em função da taxa de chegada de requisições de vídeo. Estes gráficos sintetizam as conclusões acima. Por exemplo, pode-se observar que a capacidade do servidor aumenta conforme o aumento da taxa de chegada, e que a reserva de canais não é uma solução atrativa.

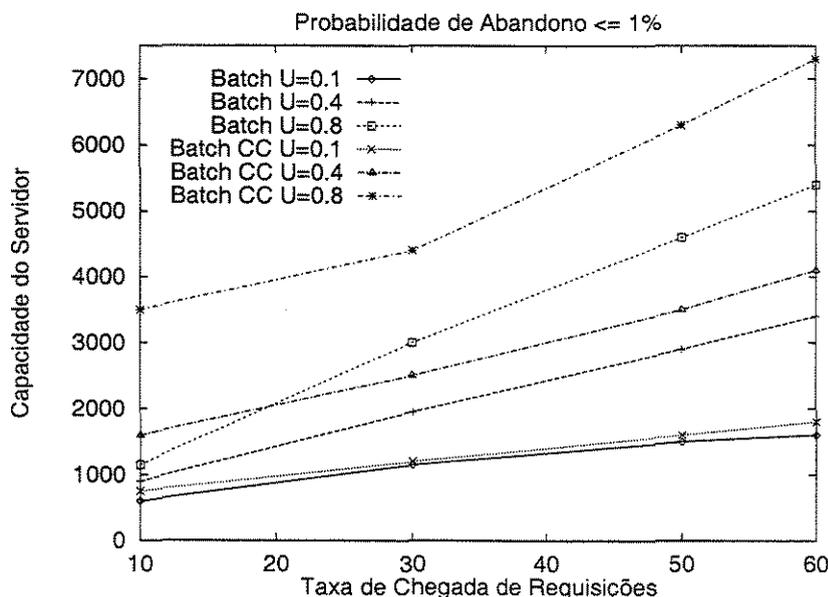


Figura 4.9: Capacidade do Servidor \times Taxa de Chegada, para probabilidade de abandono inferior a 1%.

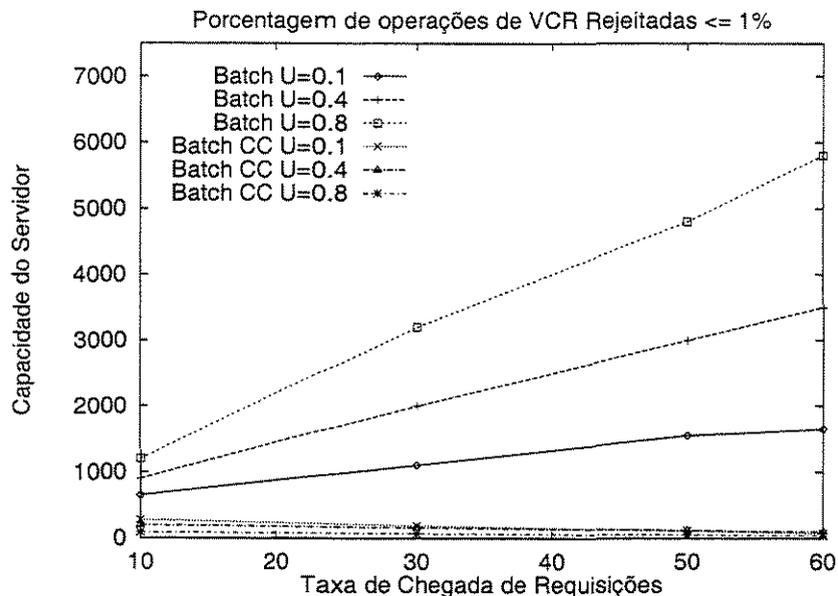


Figura 4.10: Capacidade do Servidor \times Taxa de Chegada, para porcentagem de requisições de VCR rejeitadas inferior a 1%.

Para um servidor grande, o número de canais necessários para obter a probabilidade de abandono desejada é tipicamente maior que para um servidor pequeno. Isto ocorre porque probabilisticamente há um maior número de canais alocados a um único usuário em um sistema grande do que há em um servidor pequeno.

As Figuras 4.11 e 4.12 mostram, respectivamente, a capacidade do servidor, sob altas cargas, necessária para obter a probabilidade de abandono e a porcentagem de requisições de VCR rejeitadas inferiores a 1%, como função do número de vídeos armazenados (tamanho do servidor).

O aumento da capacidade do servidor acompanha o crescimento do número de vídeos armazenados. Este aumento é mais acentuado em sistemas com canais de contingência do que em sistemas sem estes canais, em especial para graus de interatividade médio a alto. Por exemplo, um aumento no tamanho do servidor de 100 para 300 filmes armazenados, em sistema com contingência, acarreta um aumento de 2200 canais para 80% de interatividade e um aumento de 1800 canais para 40% de interatividade, enquanto que, em sistemas sem canais de contingência, este aumento fica em torno de 400 canais (Figura 4.11).

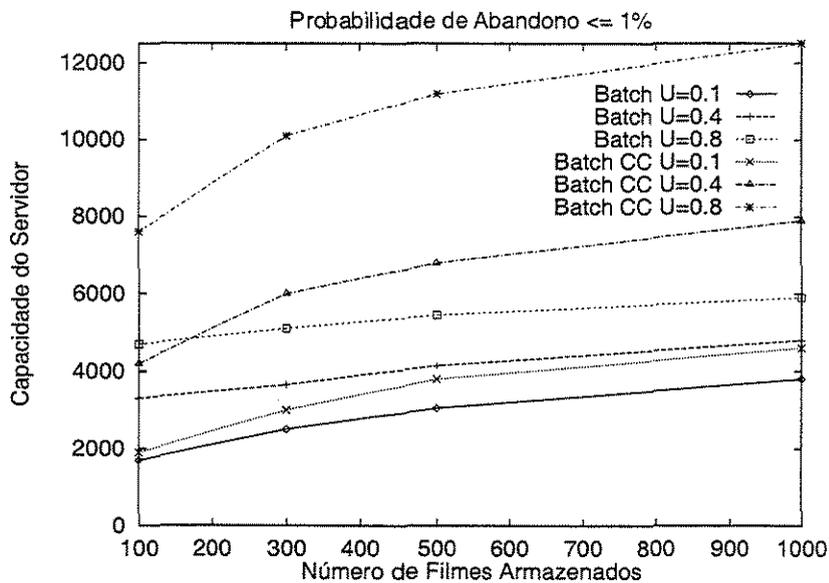


Figura 4.11: Capacidade do Servidor \times Número de Vídeos Armazenados, para probabilidade de abandono inferior a 1%, à taxa de 60 req/min.

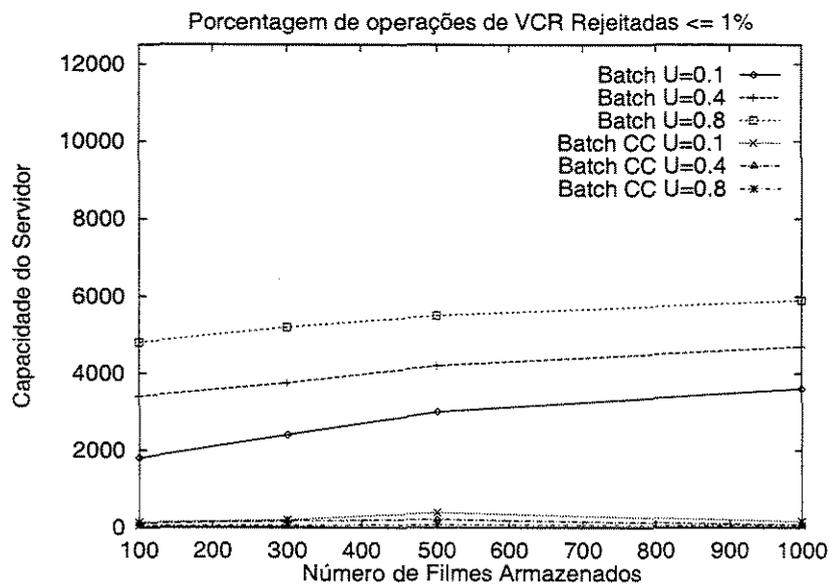


Figura 4.12: Capacidade do Servidor \times Número de Vídeos Armazenados, para porcentagem de requisições de VCR rejeitadas inferior a 1%, à taxa de 60 req/min.

Comparando as Figuras 4.11 e 4.12, fica evidente que um sistema sem canais de contingência é preferível, visto que o aumento da capacidade do servidor, nestes sistemas, para manter a porcentagem de requisições de operações de VCR rejeitadas abaixo de 1% é muito menor que o aumento da capacidade do servidor para manter a probabilidade de abandono abaixo de 1% em sistemas com canais de contingência.

Sob baixas cargas, a demanda extra de capacidade do servidor é significativa apenas para sistemas com canais de contingência com grau de interatividade médio a alto, como pode ser observado nas Figuras 4.13 e 4.14. No entanto, o aumento da capacidade do sistema é menor sob baixas cargas do que sob altas cargas. Por exemplo, para um aumento no tamanho do servidor de 100 para 300 vídeos armazenados, o aumento da capacidade é de 1500 canais e de 1000 canais para graus de interatividade de 80 % e 40% respectivamente.

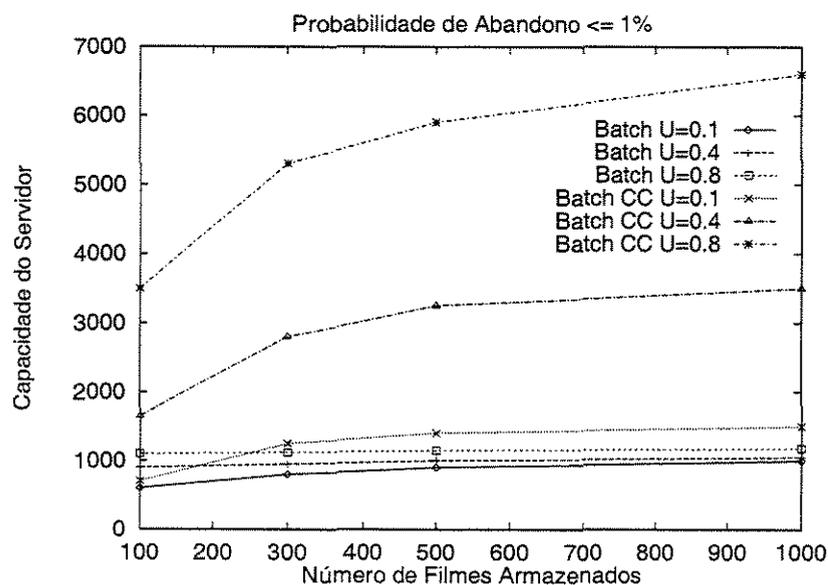


Figura 4.13: Capacidade do Servidor \times Número de Vídeos Armazenados, para probabilidade de abandono inferior a 1%, à taxa de 10 req/min.

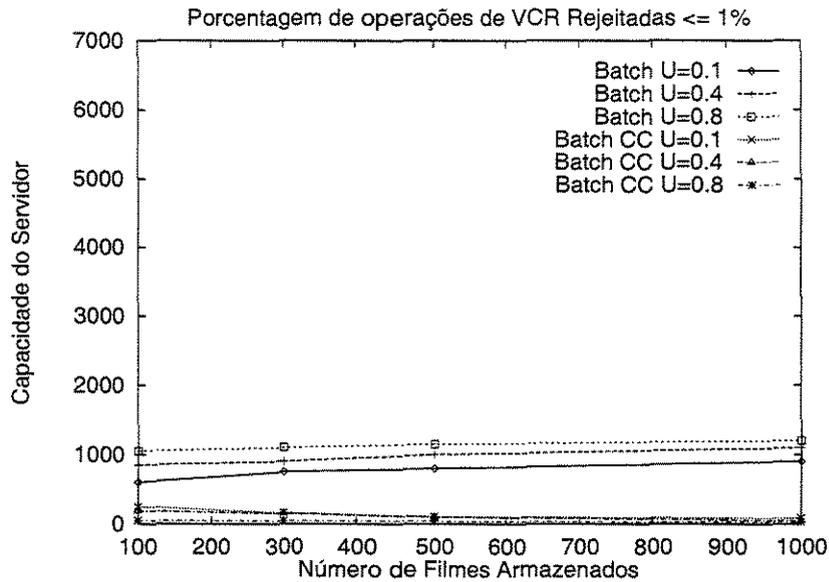


Figura 4.14: Capacidade do Servidor \times Número de Vídeos Armazenados, para porcentagem de requisições de VCR rejeitadas inferior a 1%, à taxa de 10 req/min.

4.6.2 Sistemas de VoD com *Batching* e *Piggybacking*

As Figuras 4.15, 4.16 e 4.17 apresentam, respectivamente, o número de usuários admitidos no sistema, a probabilidade de abandono e a porcentagem de operações de VCR rejeitadas, para um servidor pequeno sob cargas baixas. Figuras 4.18, 4.19 e 4.20 mostram os mesmos gráficos para um servidor pequeno, sob cargas altas.

Sob cargas baixas, o número de usuários admitidos no sistema com *batching* e *piggybacking* segue a mesma tendência encontrada em um servidor com *batching* apenas. A diferença entre o número de usuários admitidos em um sistema com conjunto de canais de contingência, e o número de usuários admitidos em um sistema sem conjunto de contingência aumenta com o grau de interatividade.

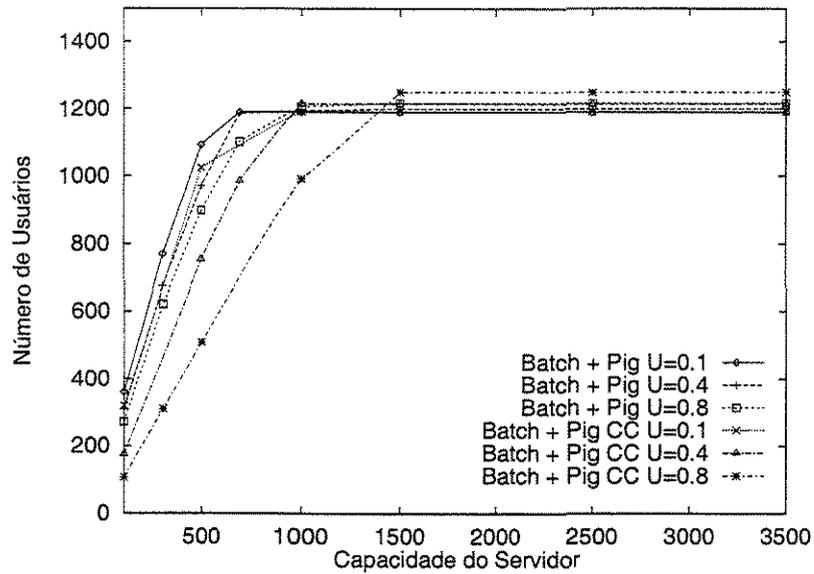


Figura 4.15: N° de Usuários Admitidos \times Capacidade do Servidor, taxa de 10 req/min.

No entanto, em um sistema com *batching* e *piggybacking*, esta diferença é menor do que em um sistema com *batching*, para graus de interatividade altos, dado que o número de usuários admitidos converge mais rapidamente para o seu valor máximo. Por exemplo, para 80% de interatividade e 1000 canais esta diferença é de 650 usuários em sistemas com *batching*, enquanto que para sistemas com *batching* e *piggybacking* ela é de 200 usuários. Para um grau de interatividade baixo esta diferença não é significativa.

Para um grau de interatividade fixo, sistemas sem conjunto de canais de contingência fornecem uma probabilidade de abandono menor que sistemas com conjunto de canais de contingência (Figura 4.16). Entretanto, em um sistema com *batching* e *piggybacking*, valores específicos de probabilidade de abandono são obtidos com um número menor de canais do que em um sistema com apenas *batching* (Figura 4.4). Para sistemas com canais de contingência e para graus de interatividade médio e alto, necessita-se do dobro do número de canais em um sistema com *batching*, para se obter um valor da probabilidade de abandono de 1%, do que se necessita em um sistema com *batching* e *piggybacking*. Por exemplo, em um sistema com canais de contingência para 40% de interatividade, a probabilidade de abandono de 1% é atingida com 850 e 1700 canais em sistemas com

batching e *piggybacking* e em sistemas com apenas *batching*, respectivamente.

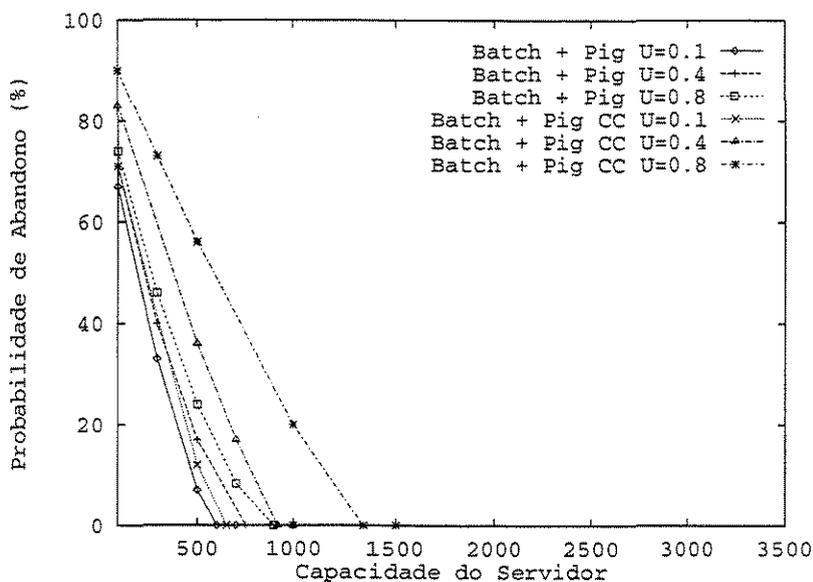


Figura 4.16: Probabilidade de Abandono \times Capacidade do Servidor, taxa de 10 req/min.

O mesmo padrão da porcentagem de operações de VCR rejeitadas observado em um sistema com *batching* apenas (Figura 4.5) é observado em um sistema com *batching* e *piggybacking* (Figura 4.17). Em um sistema com somente *batching*, porcentagens inferiores ao valor alvo são obtidas para capacidade de sistema de 300 canais, enquanto que, em sistemas com *batching* e *piggybacking*, elas são atingidas para uma capacidade de sistema de 50 canais. Similarmente a sistemas com *batching*, em sistemas com *batching* e *piggybacking*, é pouco atrativo reservar um conjunto de contingência. Note que, em sistemas sem canais de contingência, tanto a probabilidade de abandono, quanto a porcentagem de operações de VCR rejeitadas, obtém-se valores alvos para capacidade menores que a necessária em sistemas com conjunto de contingência (Figura 4.16).

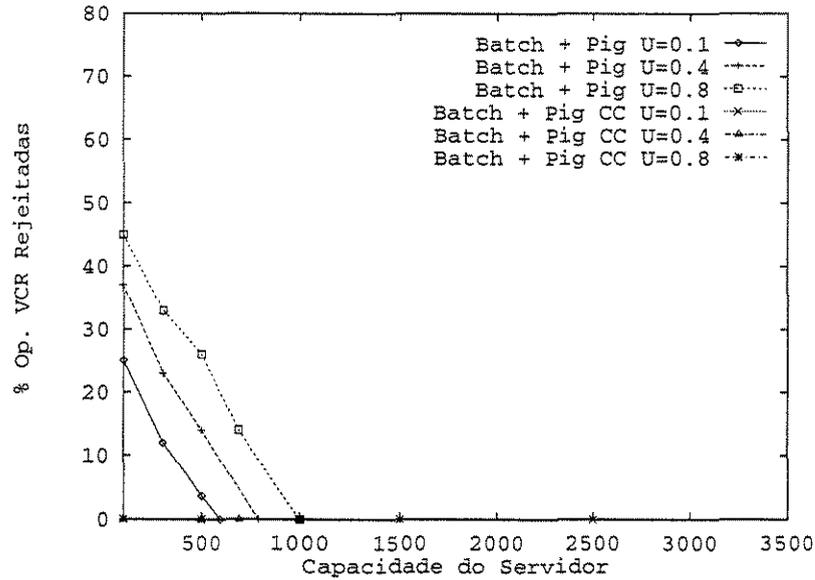


Figura 4.17: Porcentagem de operações de VCR Rejeitadas \times Capacidade do Servidor, taxa de 10 req/min.

As vantagens de se adotar *piggybacking* em conjunto com *batching* tornam-se evidentes ao se observar o sistema sob cargas altas. Enquanto que em um sistema com *batching* o número máximo de usuários admitidos no sistema é da ordem de 4500 (Figura 4.6), em um sistema com *batching* e *piggybacking* o número máximo de usuários admitidos chega a 8000 (Figura 4.18).

Sob altas cargas, obtém-se um valor para probabilidade de abandono de 1% com capacidade do sistema significativamente maior que em baixas cargas. Por exemplo, se, sob baixas cargas, para um grau de interatividade de 80%, em um sistema com canais contingentes, a demanda de canais é menor que 1500, sob altas cargas (Figura 4.19) ela é de 5800 canais. A vantagem de se adotar conjuntamente *batching* e *piggybacking* pode, também, ser apreciada ao se comparar a menor demanda de canais para se obter o valor alvo de probabilidade de abandono em um sistema com *batching* e *piggybacking* (Figura 4.19) do que a demanda em sistemas com *batching* apenas (Figura 4.7).

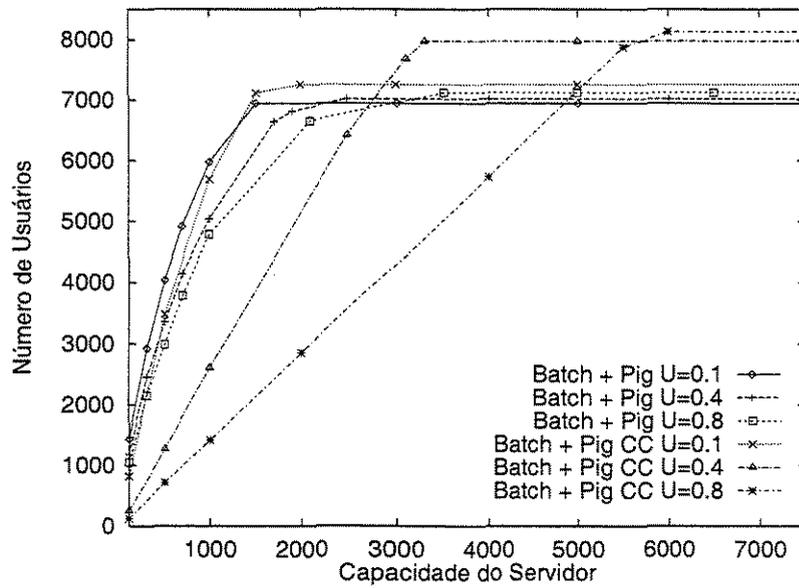


Figura 4.18: N° de Usuários Admitidos \times Capacidade do Servidor, taxa de 60 req/min.

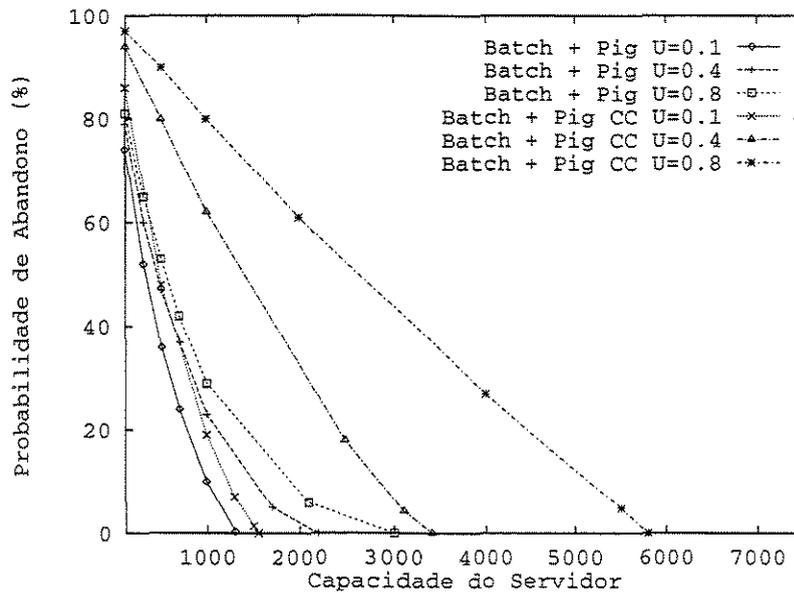


Figura 4.19: Probabilidade de Abandono \times Capacidade do Servidor, taxa de 60 req/min.

Para um grau de interatividade baixo (10%), um sistema com canais de contingência é preferível do que um sistema sem canais de contingência, dado que os valores alvos da probabilidade de abandono, para sistema com contingência (Figura 4.19), bem como da percentagem de operações VCR rejeitadas, para sistema sem contingência, (Figura 4.20) são obtidas com aproximadamente o mesmo número de canais. No entanto, para valores médio e alto do grau de interatividade, os valores alvos são obtidos com um menor número de canais, em um sistema sem canais de contingência. Por exemplo, para 80% de interatividade o conjunto das métricas a 1% é atingido com 3200 canais para sistemas sem canais de contingência, e com 5900 canais para sistemas com canais de contingência.

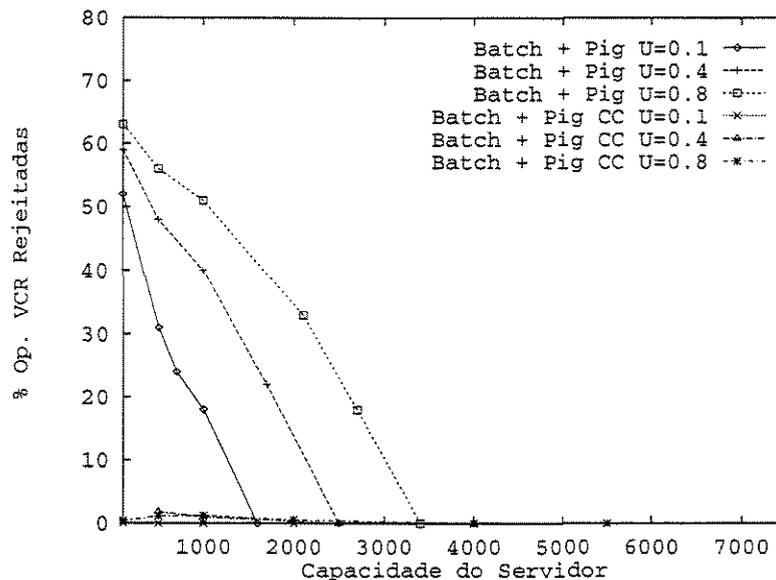


Figura 4.20: Porcentagem de operações de VCR Rejeitadas \times Capacidade do Servidor, taxa de 60 req/min.

O comportamento do sistema, conforme se aumenta taxa de chegada de requisições de vídeo, pode ser observado mais claramente nas Figuras 4.21 e 4.22. Percebe-se que para altas cargas o sistema sem canais de contingência é mais apropriado. Além disso, comparando as Figuras 4.21 e 4.9, fica evidente que um sistema que utiliza em conjunto *batching* e *piggybacking* atinge as metas de desempenho com uma capacidade de sistema significativamente menor.

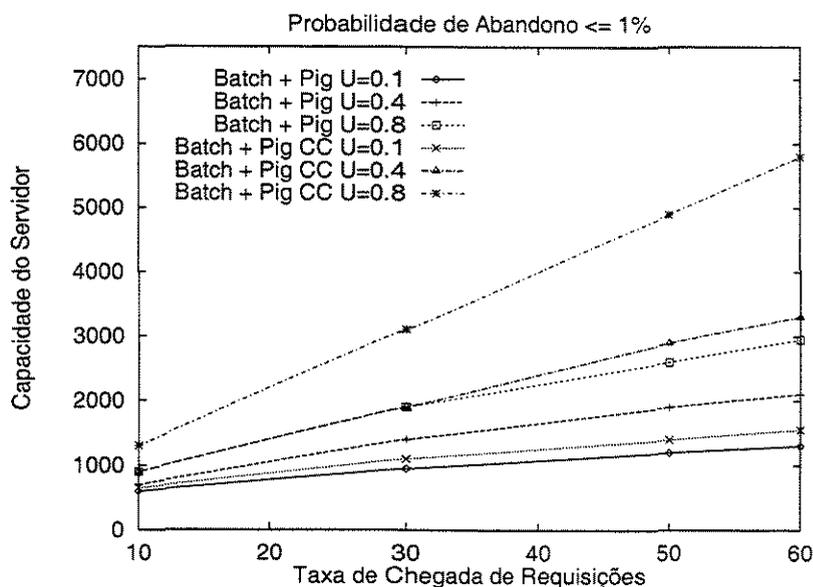


Figura 4.21: Capacidade do Servidor \times Taxa de Chegada, para probabilidade de abandono inferior a 1%.

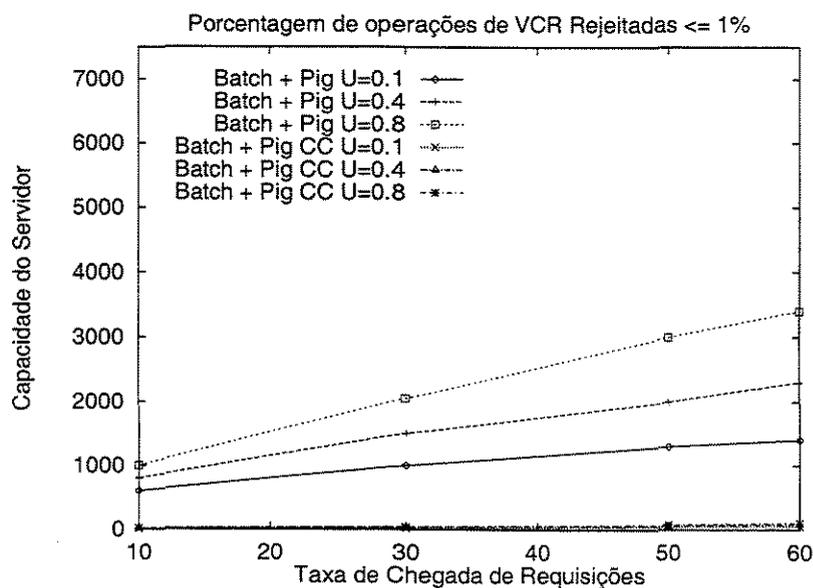


Figura 4.22: Capacidade do Servidor \times Taxa de Chegada, para porcentagem de requisições de VCR rejeitadas inferior a 1%.

As Figuras 4.23 e 4.24 mostram, respectivamente, a capacidade do servidor necessária para manter a probabilidade de abandono e a porcentagem de operações de VCR rejeitadas abaixo de 1%, em função do tamanho do servidor, sob altas cargas. As Figuras 4.25 e 4.26 mostram os mesmos gráficos sob baixas cargas.

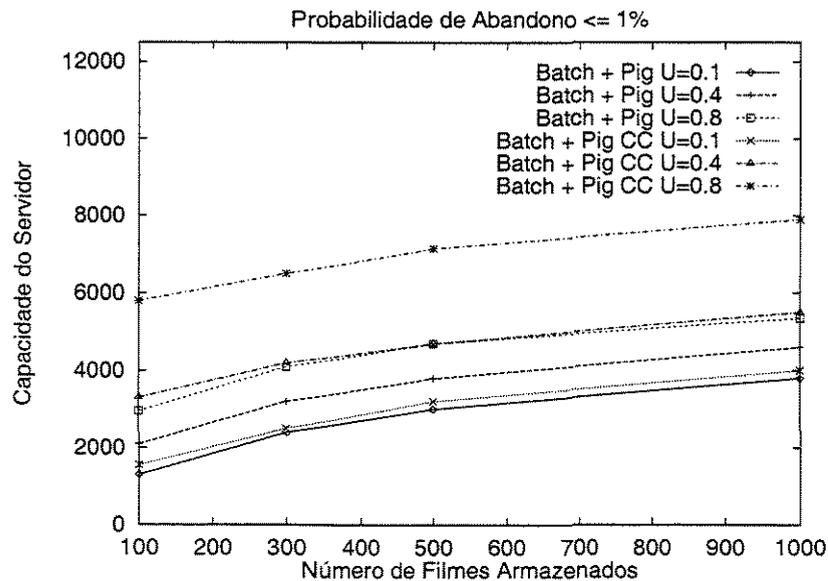


Figura 4.23: Capacidade do Servidor \times Número de Vídeos Armazenados, para probabilidade de abandono inferior a 1%, à taxa de 60 req/min.

A vantagem de se utilizar conjuntamente *batching* e *piggybacking* pode ser avaliada pela menor demanda de capacidade em sistemas com estas duas técnicas (Figura 4.23) do que em sistemas com somente *batching* (Figura 4.11). Por exemplo, em um sistema com canais de contingência, para grau de interatividade de 80%, um aumento no tamanho do servidor de 100 para 300 filmes leva a um aumento de 800 canais e de 2500 canais para sistemas com *batching* e *piggybacking* e sistemas com apenas *batching* respectivamente.

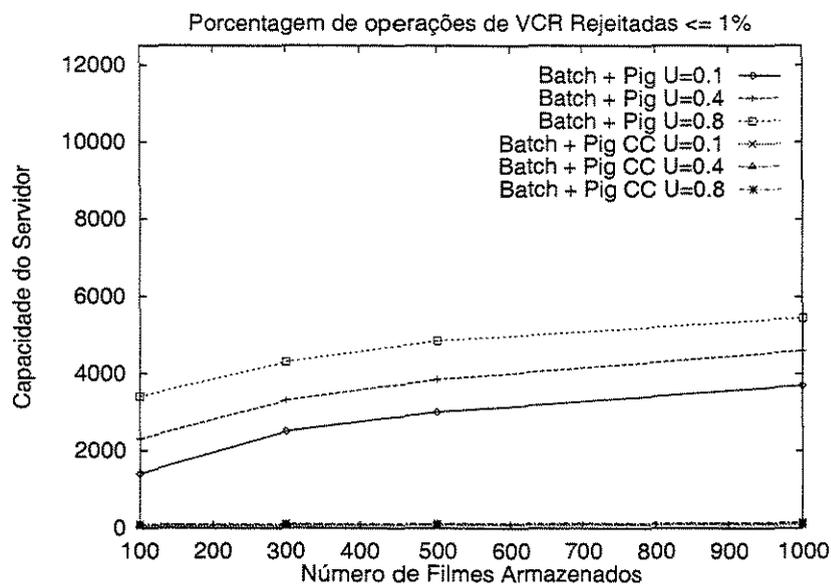


Figura 4.24: Capacidade do Servidor \times Número de Vídeos Armazenados, para porcentagem de requisições de VCR rejeitadas inferior a 1%, à taxa de 60 req/min.

O impacto da utilização de *batching* e *piggybacking* na porcentagem de operações de VCR rejeitadas é pouco significativa em ambos os sistemas com e sem canais de contingência.

Sob baixas cargas, apenas para sistemas com canais de contingência e graus de interatividade médio e alto, o uso em conjunto de *batching* e *piggybacking* é vantajoso. Por exemplo, para um aumento no tamanho do servidor de 100 para 500 vídeos armazenados, para grau de interatividade de 80%, o aumento da capacidade é de 2400 canais para sistemas com apenas *batching* (Figura 4.13), enquanto que, para sistemas com *batching* e *piggybacking*, este aumento é de apenas 350 canais (Figura 4.25).

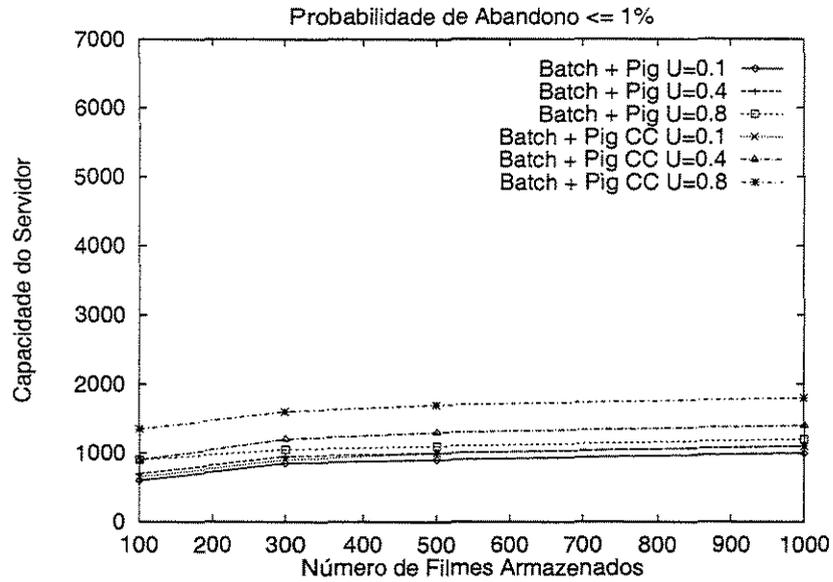


Figura 4.25: Capacidade do Servidor \times Número de Vídeos Armazenados, para probabilidade de abandono inferior a 1%, à taxa de 10 req/min.

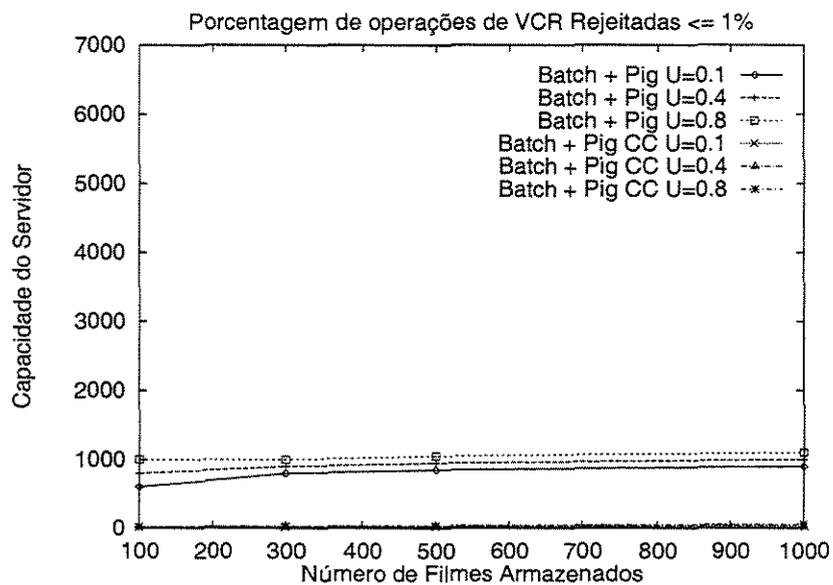


Figura 4.26: Capacidade do Servidor \times Número de Vídeos Armazenados, para porcentagem de requisições de VCR rejeitadas inferior a 1%, à taxa de 10 req/min.

É interessante notar que a suposição de que um fluxo de vídeo que retém um canal de contingência pode apenas se unir com o fluxo *multicast* original conduz a estimações conservadoras do número de canais demandados. Conseqüentemente, a conclusão de que sistemas sem reserva de canais de contingência são mais atrativos que sistemas com estes canais pode ser contestada. Entretanto, pode-se notar que os resultados aqui apresentados correspondem a um conjunto de experimentos com $N_{ver} = 2$. Estimções baseadas em tal valor de N_{ver} são próximas ao número médio de canais demandados, derivado via simulação (veja Figura 4.2), e, portanto, reforçam as conclusões apresentadas acima.

Capítulo 5

Conclusões

Com o advento do paradigma da integração de serviços no contexto das redes de telecomunicações, novos serviços têm sido criados e oferecidos aos usuários. Estes serviços, que se enquadram na classe das aplicações multimídia, despertam grande interesse em seus usuários em virtude de sua principal característica: o intercâmbio de dados através da utilização de diversas mídias de informação.

Novas aplicações multimídia surgem continuamente, mantendo-se sempre crescente o número de usuários potenciais de seus serviços. Nesta classe de aplicações destacam-se principalmente aquelas em que as mídias de áudio e, notadamente, de vídeo encontram-se presentes. As técnicas para a digitalização de vídeo, que surgiram na década de 80, possibilitaram o desenvolvimento destas aplicações. Destacando-se entre as demais encontra-se o Vídeo sob Demanda, esta aplicação permite a um usuário selecionar um vídeo dentre uma grande coleção e controlá-lo através de operações de VCR, como se o mesmo assistisse a um filme utilizando um aparelho de vídeo cassete em sua residência. O mercado para os serviços de VoD é extremamente abrangente, o que sugere um elevado potencial comercial.

No entanto, o oferecimento de um serviço de VoD em larga escala ainda não é possível em decorrência das limitações tecnológicas que existem atualmente. Diversas iniciativas envolvendo universidades e empresas têm sido implementadas com o objetivo de tornar operacional o oferecimento de serviços de VoD. Porém, estas possuem caráter experimental

e não abrangem uma grande população de usuários.

Uma vez que VoD pressupõe a distribuição de vídeo de alta qualidade, muitas vezes no formato de HDTV, um dos pontos críticos nestes sistemas é a largura de banda requerida para a transmissão. O emprego da tecnologia de fibra ótica na rede de transmissão pode suavizar esta limitação. Deste modo, os servidores tornam-se o ponto crítico do sistema. Mais precisamente, o subsistema de armazenamento (hierarquia de armazenamento) precisa lidar com um grande número de transações e suportar uma quantidade elevada de fluxos de dados.

As restrições existentes nos servidores, notadamente no subsistema de armazenamento, motivaram a proposição de um conjunto de técnicas que objetivam a redução do montante de banda passante requerido para a transmissão de fluxos de dados. Neste contexto, destacam-se as técnicas de *Batching* e *Piggybacking*, as quais propõem o compartilhamento de fluxos como mecanismo para o gerenciamento mais eficiente da demanda de banda passante em servidores de vídeo.

Nesta dissertação apresentou-se um estudo sobre o emprego de ambas as técnicas. O trabalho aborda problemas atuais e sua realização justifica-se pela importância da redução da demanda de banda passante para o atendimento de uma grande população de usuários.

O estudo envolveu a proposição de um modelo aproximado para determinar o número de canais necessários para dar suporte a operações de VCR em sistemas de VoD. O modelo é uma fila Erlang B cuja taxa de chegada é uma aproximação da taxa de chegada de requisições de operações de VCR. A precisão da estimativa do número de canais foi verificada por resultados de simulações. O número de canais estimado é um limite superior do número médio de canais demandados, e é limite inferior do número máximo de canais exigidos.

Além disso, avaliaram-se aspectos de desempenho em servidores considerando-se diferentes cenários. Também foi analisado o desempenho de diversos sistemas de VoD. A eficiência de reservar canais de contingência para dar suporte a execução de operações de VCR foi investigada. Mostrou-se que, em sistemas sem canais de contingência, a meta de 1% para os valores da probabilidade de abandono, bem como da porcentagem de operações de VCR rejeitadas, é obtida com uma capacidade do sistema menor que em

sistemas com o conjunto de canais de contingência. Além disso, um sistema sem conjunto de contingência aceita um maior número de usuários que um sistema com canais de contingência. Adicionalmente, mostrou-se que um sistema interativo com *Batching* e *Piggybacking* admite um maior número de usuários, e provê menores probabilidades de abandono e de rejeição de operações de VCR que um sistema com apenas *Batching*.

O sistema proposto por Dey Sircar et all [57], contempla as operações de FF e REW, mas não utiliza técnicas de compartilhamento de fluxos. O sistema aqui proposto apresenta melhores resultados em relação ao número de usuários admitidos no sistema, enfatizando assim a vantagem da utilização de *batching* e *piggybacking*. Outra diferença é que as requisições de operações de VCR podem sofrer atrasos, o que não ocorre no nosso sistema em que as requisições são imediatamente atendidas ou rejeitadas.

O sistema de Towsley et all [56] utiliza *batching* e trata apenas a operação PAUSE/RESUME. Seus resultados, assim como os nossos, mostram que o uso de *batching* melhora o desempenho do sistema. O número total de canais utilizados nos dois esquemas foram similares, mas no sistema de Towsley o uso de canais de contingência reduz o uso total de canais utilizados, o que não ocorre em nosso sistema. Esta diferença ocorre pelo fato que o usuário não utiliza canal durante a operação de VCR, além disso tolera-se um atraso de pelo menos 30s para a retomada da exibição e também a probabilidade de abandono fica em torno de 5%, e não abaixo de 1% como no nosso caso.

Os experimentos realizados evidenciam a extensão dos resultados contidos na literatura, contribuindo para o avanço do estado da arte da pesquisa sobre *Batching* e *Piggybacking*, sua integração e suporte a operações de VCR.

Como trabalho futuro, sugere-se uma comparação entre sistemas em que requisições de operações de VCR são atrasadas quando não houver canais disponíveis, e sistemas nos quais requisições de VCR são rejeitadas ao invés de atrasadas. Outra sugestão é a implementação de suporte à técnica de *piggybacking*, sendo necessário analisar o custo computacional e o impacto no número de canais que o servidor é capaz de sustentar, com utilização desta técnica. Finalmente, pode-se dizer que o oferecimento de serviços de VoD em larga escala requer uma maturação da tecnologia envolvida, isto é, pesquisas precisam ser desenvolvidas abordando assuntos como: técnicas de armazenamento de

arquivos multimídia, suporte a aplicações multimídia em sistemas operacionais através de políticas para o escalonamento de processos e/ou *threads*, esquemas para o gerenciamento de *buffers*, balanceamento de carga em discos, suporte a operações de VCR, a *set top box*, além do próprio gerenciamento da demanda de banda passante, dentre outros.

Apêndice A

Tabelas de Símbolos

Símbolo	Descrição
λ	Taxa de chegada de requisições de vídeo.
λ_{op}	Média da distribuição exponencial que determina a duração das operações de VCR.
λ_{vcr}	Taxa de chegada de requisições de operações de VCR de um usuário.
$\bar{\lambda}$	Taxa média de chegada de requisições de de operações de VCR.
$\mu_i(j)$	Taxa de serviço do servidor no dispositivo i em função do número j de usuários em sua fila em dado instante de tempo.
θ	Parâmetro para a distribuição Zipf.
ω	Tempo mínimo de espera da política <i>Batch</i> MBQ.
ζ	Parâmetro de confiança para a determinação de ω na política <i>Batch</i> MBQ.
Δt	Intervalo entre alocações de fluxos de vídeo no esquema NVOD.
\mathfrak{S}	Janela de <i>Batching</i> .

Tabela A.1: Letras Gregas.

Símbolo	Descrição
$\mathcal{A}(i, j)$	Árvore de mesclagem dos fluxos i, \dots, j .
A	Constante de multiplicação usada para o cálculo do tempo para a mesclagem de um fluxo, cujo usuário está efetuando op. de VCR, com seu fluxo original (grupo de <i>batching</i>).
A_{ki}	Número de fluxos que se tornaram disponíveis até o instante de alocação da i -ésima requisição do vídeo k (política LAMB).
Abd	Tempo de Abandono dos usuários, modelado através da distribuição normal, para o estudo de <i>Batching</i> .
$B(N_d, \vec{k})$	Rede de filas fechada multiclasse com solução em forma de produtos, com N_d dispositivos e com \vec{k} usuários.
$C(i, j)$	Custo para mesclagem dos fluxos i, \dots, j .
C	Capacidade do Servidor (número máximo de canais suportados por um servidor)
\mathcal{C}	Conjunto de vídeos não populares no esquema de <i>Batch Min_Idle</i> .
D	Tempo de retenção médio do canal de contingência.
E_c	Valor da fórmula Erlang B.
F	Número de vídeos armazenados no sistema.
$F_{op}(t)$	Função densidade de probabilidade da duração da operação de VCR <i>op</i> .
$G_{op}(s, t)$	Duração do tempo de retenção de um canal de contingência de uma operação de VCR <i>op</i> que ocorreu no s -ésimo quadro e durou t segundos.
\mathcal{H}	Conjunto de vídeos populares no esquema de <i>Batch Min_Idle</i> .
I	Intervalo <i>Snapshot</i> .
K	População total no sistema modelado por redes de filas fechadas.
L	Número de quadros de um vídeo.
	Continua na próxima página...

Tabela A.2: Letras Maiúsculas.

Símbolo	Descrição
$\text{Max}_{op}(s)$	Duração máxima da operação op que ocorra no quadro s .
N	Número de usuários que executam operações de VCR.
N_d	Número de dispositivos no modelo de redes de filas fechadas.
N_{vcr}	Número médio de operações de VCR executadas por usuário.
$P_i(j, \vec{k})$	Probabilidade de haver j usuários no dispositivo i , no modelo de rede de filas fechada.
P_{op}	Probabilidade da operação de VCR op .
P_{playback}	Probabilidade de um usuário estar assistindo a um vídeo em modo <i>playback</i> .
P_{vcr}	Probabilidade de um usuário estar em modo VCR, ou seja, efetuando operação de VCR.
$\text{Pos}(i, j)$	Posição de mesclagem dos fluxos de vídeo i e j .
Q_{ir}	Número médio de usuários da classe r no dispositivo i no modelo de redes de filas fechado.
$Q(\ell)$	Número de quadros exibidos por um vídeo dentro de um intervalo <i>Snapshot</i> .
R	Número de classes no modelo fechado de redes de filas.
R_{FF}	Constante de multiplicação para produzir a velocidade de exibição do vídeo durante a operação de avanço (FF).
R_{Rew}	Constante de multiplicação para produzir a velocidade de exibição do vídeo durante a operação de retrocesso (REW).
S	Número de fluxos de vídeo disponíveis no instante de definição de uma janela de estudo (intervalo de <i>Batching</i> esgotado).
S_{ir}	Tempo de serviço médio dos usuários da classe r no dispositivo i , em redes de filas fechadas.
T	Duração original de um filme, em segundos.
	Continua na próxima página. . .

Tabela A.2: Letras Maiúsculas.

Símbolo	Descrição
T_{ir}	Vazão da classe r no dispositivo i , em redes de filas fechadas.
U_{ir}	Utilização do dispositivo i pelos usuários da classe r , em redes de filas fechadas.
V_{FF}	Velocidade de exibição de um fluxo de vídeo durante a operação de avanço (FF), $V_{FF} = R_{FF} \times V_n$.
V_{ir}	Taxa média de visita (<i>visit ratio</i>) dos usuários da classe r no dispositivo i em redes de filas fechadas.
V_{max}	Velocidade máxima de exibição de um fluxo de vídeo no contexto de <i>Piggybacking</i> , $V_{max} = 1,05 \times V_n$.
V_{min}	Velocidade mínima de exibição de um fluxo de vídeo no contexto de <i>Piggybacking</i> , $V_{min} = 0,95 \times V_n$.
V_n	Velocidade normal de exibição de um fluxo de vídeo, $V_n = 30$ quadros/segundo. Também chamada de V_{playback} .
V_{Rew}	Velocidade de exibição de um fluxo de vídeo durante a operação de retrocesso (REW), $V_{Rew} = R_{Rew} \times V_n$.
W	Janela ótima de mesclagem.
W_{ir}	Tempo médio de espera, incluindo o tempo de serviço, dos usuários da classe r no dispositivo i , em redes de filas fechadas.
W_m	Janela máxima de mesclagem.
W'_m	Janela máxima alterada de mesclagem (política S^2).
$W_{Pol}(f_i)$	Janela de mesclagem para a política Pol calculada em relação à posição f_i .

Tabela A.2: Letras Maiúsculas.

Símbolo	Descrição
c	Número de canais de contingência.
$d(n)$	Tempo de retenção médio do canal de contingência dado que n operações de VCR são executadas durante a exibição do vídeo.
$d_{op}(s, t)$	Tempo de retenção do canal de contingência dado que a operação de VCR op ocorreu no s -ésimo quadro e durou t segundos.
e_{ki}	Tempo de abandono di i -ésimo usuário da k -ésima fila (política LAMB).
f_i	Quadro atualmente exibido pelo fluxo de vídeo i .
\vec{k}	Vetor de população em redes de filas fechadas.
op	Operação de VCR (Pause, FF, REW).
p_k	Probabilidade de escolha do vídeo k conforme a distribuição Zipf.
p_n	Probabilidade de um usuário requisitar n operações de VCR durante o vídeo.
p_{vcr}	Probabilidade de um usuário efetuar operações de VCR.
q_k	Número de requisições da fila do vídeo k .
τ_k	Custo (taxa monetária) para assistir ao vídeo k .
s	Quadro em que ocorreu uma op. de VCR.
t	Duração da operação de VCR.
$t_{k,i}$	Instante de alocação (abandono) da i -ésima requisição da fila do vídeo k .
t_{op}	Duração média de uma operação de VCR.
$x_{k,i}$	Variável que representa a i -ésima requisição da fila do vídeo k nos problemas de otimização formulados pela política LAMB.
z	Constante de normalização da distribuição Zipf.

Tabela A.3: Letras Minúsculas.

Referências Bibliográficas

- [1] Jean M. McManus and Keith W. Ross. Video on Demand over ATM: Constant-Rate Transmission and Transport. *IEEE Journal on Selected Areas in Communications*, 14(6):1087–1098, Agosto 1996.
- [2] N. L. S. Fonseca and R. A. Façanha. Integrating Batching and Piggybacking in Video Server. In *IEEE Global Telecommunications Conference*, pages 1334–1338, 2000.
- [3] Andrew S. Tanenbaum. *Computer Networks*. Computer Networking. Prentice Hall, 3 edition, 1996.
- [4] Y. H. Chang, D. Coggins, D. Pitt, D. Skellern, M. Thapar, and C. Venkatraman. An open-system approach to video-on-demand. *IEEE Communications Magazine*, pages 68–80, Maio 1997.
- [5] Daniel Minoli. *Video Dialtone Technology: Digital Video over ADSL, HFC, FTTC & ATM*. McGraw Hill, 1995.
- [6] D. Le Gall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):47–58, Abril 1991.
- [7] Luiz Fernando Gomes Soares, Guido Lemos, and Sérgio Colcher. *Redes de Computadores: Das Lans, Mans e Wans às Redes ATM*. Redes e Conectividade. Editora Campus, 1995.
- [8] J. A. S. Monteiro. *Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL)*. IX Escola de Computação, Recife, PE, 1994.

- [9] Martin de Prycker. *Asynchronous Transfer Mode: Solution for Broadband ISDN*. Prentice Hall, 3 edition, 1995.
- [10] Ralf Steinmetz. Multimedia File Systems Survey: Approaches for Continuous Media Disk Scheduling. *Computer Communications*, 18(3):133–144, Março 1995.
- [11] C. L. Jonathan Liu, Jenwei Hsieh, David H. C. Du, and Mengjou Lin. Performance of A Storage System for Supporting Different Video Types and Qualities. In *Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer and Communications Societies: Networking the Next Generation*, volume 1, pages 2–9, Março 1996.
- [12] Banu Özden and Rajee Rastogi and Avi Silberschatz. On the Design of a Low-Cost Video-on-Demand Storage Server. *Multimedia Systems*, 4(1):40–54, Fevereiro 1996.
- [13] Asit Dan, Martin Kienzle, and Dinkar Sitaram. A Dynamic Policy of Segment Replication for Load-Balancing in Video-on-Demand Servers. *Multimedia Systems*, 3(3):93–103, 1995.
- [14] ChengFu Chou, Leana Golubchik, and John C. S. Lui. A Performance Study of Dynamic Replication Techniques in Continuous Media Servers. In *IEEE MASCOTS - International Symposium on Modeling Analysis and Simulation of Computer and Telecommunication Systems*, Agosto 2000.
- [15] Kun-Lung Wu and Philip S. Yu. Consumption-Based Buffer Management for Maximizing System Throughputs of News-on-Demand Multimedia Systems. Technical Report RC 20143 (98156), IBM Research Report, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, Janeiro 1995.
- [16] Leana Golubchik, John C. S. Lui, Edmundo de Souza e Silva, and H. Richard Gail. Evaluation of Tradeoffs in Resource Management Techniques for Multimedia Storage Servers. 1998.
- [17] Antoine N. Mourad. Issues in the Design of a Storage Server for Video-on-Demand. *Multimedia Systems*, 4(2):70–86, Abril 1996.

- [18] Harrick M. Vin, Alok Goyal, and Pawan Goyal. Algorithms for Designing Multimedia Servers. *Computer Communications*, 18(3):192–203, Março 1995.
- [19] Chiung-Shien Wu, Gin-Kou Ma, and Bao-Shuh P. Lin. On Scalable Design of an ATM-based Video Server. In *Proceedings of the ICC'96: Converging Technologies for Tomorrow's Applications*, volume 1, pages 1335–1340, Dallas, Junho 1996. IEEE Communications Society.
- [20] Y. B. Lee and P. C. Wong. A Server Array Approach for Video-on-Demand Service on Local Area Networks. In *Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer and Communications Societies: Networking the Next Generation*, volume 1, pages 27–34, Março 1996.
- [21] Victor O. K. Li, Wanjiun Liao, Xiaoxin Qiu, and W. M. Wong. Performance Model of Interactive Video-on-Demand Systems. *IEEE Journal on Selected Areas in Communications*, 14(6):1099–1109, Agosto 1996.
- [22] Kelvin C. Almeroth and Mostafa H. Ammar. The Use of Multicast Delivery to Provide a Scalable and Interactive Video-on-Demand Service. *IEEE Journal on Selected Areas in Communication*, 14(6):1110–1122, Agosto 1996.
- [23] Jean-Paul Nussbaummer and Frank Schaffa. Capacity Analysis of CATV for on-Demand Multimedia Distribution. In *Proceedings of the First ISMM International Conference on Distributed Multimedia Systems Applications*, Agosto 1994.
- [24] Jean-Paul Nussbaummer and Frank Schaffa. Impact of Channel Allocation Policies on Video on Demand Over CATV. Private Communication.
- [25] S. H. Gary Chan and F. A. Tobagi. On achieving profit in providing near video-on-demand services. In *IEEE International Conference on Communications*, volume 2, pages 988–999, June 1999.
- [26] J. P. Nussbaummer, B. V. Patel, F. Schaffa, and J.P.G. Sterbenz. Networking Requirements for Interactive Video on Demand. *IEEE Journal on Selected Areas in Communications*, 13(No. 5):779–787, Junho 1995.

- [27] Chatschik C. Bisdikian and Baiju V. Patel. Issues on Movie Allocation in Distributed Video-on-Demand Systems. In *Proceedings of the ICC'95*, volume 3, pages 250–255, Seattle, Junho 1995. IEEE Communications Society: Communications — Gateway to Globalization.
- [28] Frank Schaffa and Jean Paul Nussbaumer. On Bandwidth and Storage Tradeoffs in Multimedia Distribution Networks. In *Proceedings of the Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies: Bringing Information to People*, volume 3, pages 1020–1026.
- [29] S. Sen, J. Rexford, and D. Towsley. Proxy Prefix Caching for Multimedia Streams. In *IEEE Infocom*, pages 1310–1319, 1999.
- [30] Christos Papadimitriou, Srinivas Ramanathan, P. Venkat Rangan, and Srihari SampathKumar. Multimedia Information Caching for Personalized Video-on-Demand. *Computer Communications*, 18(3), Março 1995.
- [31] Asit Dan and Dinkar Sitaram. A Generalized Interval Caching Policy for Mixed Interactive and Long Video Workloads. Technical Report RC 20206 (89404), IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598, Setembro 1995.
- [32] Asit Dan and Dinkar Sitaram. Multimedia Caching Strategies for Heterogeneous Application and Server Environments. Technical Report RC 20670 (91482), IBM Research Division, T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598, Dezembro 1996.
- [33] Philip S. Yu, Joel L. Wolf, and Hadas Shachnai. Design and Analysis of a Look-Ahead Scheduling Scheme to Support Pause-Resume for Video-on-Demand Applications. Research Report RC 19683 (87236), IBM Research Division, Agosto 1994.
- [34] Phikip S. Yu, Joel L. Wolf, and Hadas Shachnai. Design and Analysis of a Look-Ahead Scheduling Scheme to Support Pause-Resume for Video-on-Demand Applications. *Multimedia Systems*, 3(4):137–149, Setembro 1995.

- [35] Leana Golubchik, John C. S. Lui, and Richard Muntz. Reducing I/O Demand in Video-on-Demand Storage Servers. *ACM Sigmetrics*, pages 25–36, 1995. Ottawa, Canada.
- [36] Leana Golubchik, John C. S. Lui, and Richard Muntz. Adaptive Piggybacking: A Novel Technique for Data Sharing in Video-on-Demand Storage Servers. *Multimedia Systems*, 4(3):140–155, 1996.
- [37] W. F. Poon and K. T. Lo. New Batching Policy for Providing True Video-on-Demand (T-VoD) in Multicast System. In *IEEE International Conference on Communication*, 1999.
- [38] Hadas Shachnai and Philip S. Yu. The Role of Wait Tolerance in Effective Batching: A Paradigm for Multimedia Scheduling Schemes. Technical Report RC 20038 (88607), IBM Research Division, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, Abril 1995.
- [39] Charu C. Aggarwal, Joel L. Wolf, and Philip S. Yu. The Maximum Factor Queue Length Batching Scheme for Video-on-Demand Systems. Technical Report RC 20261 (91305), IBM Research Division, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, Novembro 1996.
- [40] Asit Dan Dinkar Sitaram and Perwez Shahabuddin. Dynamic Batching Policies for an on-Demand Video Server. *Multimedia Systems*, 4:112–121, 1996.
- [41] Hadas Shachnai and Philip S. Yu. An Analytical Study of Multimedia Batching Schemes. Technical Report RC 20662 (91420), IBM Research Division, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, Dezembro 1996.
- [42] N. L. S. Fonseca and R. A. Façanha. The Look-Ahead-Maximize-Batch Batching Policy. In *IEEE Global Telecommunications Conference*, pages 354–358, 1999.
- [43] Charu C. Aggarwal, Joel L. Wolf, and Philip S. Yu. Adaptive Piggybacking Schemes for Video-on-Demand Systems. Technical Report RC 20635 (91350), IBM Research

- Division, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, Novembro 1996.
- [44] Charu C. Aggarwal, Joel L. Wolf, and Philip S. Yu. On Optimal Piggyback Merging Policies for Video-on-Demand Systems. Technical Report RC 20337 (90078), IBM Research Division, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, Fevereiro 1996.
- [45] R. A. Façanha and N. L. S. Fonseca. APiggybacking Policy for Reducing the Bandwidth Demand of Video Servers. In *Managing QoS in Multimedia Networks and Services*, pages 225–236. Kluwer Academic Publishers, 2000.
- [46] Martin Gardner. Catalan numbers. *Scientific American*, pages 120–124, Junho 1976.
- [47] Byron J. T. Morgan. *Elements of Simulation*. Chapman & Hall, 1992.
- [48] Y. S. Chen. Mathematical Modeling of Empirical Laws in Computer Application: a Case of Study. In *Comp. Math. Applicat.*, pages 77–87, Outubro 1992.
- [49] Philip Branch, Greg Egan, and Bruce Tonkin. Modeling Interactive Behavior of a Video Based Multimedia System. In *IEEE International Conference on Communication*, volume 2, pages 978–982, 1999.
- [50] D. A. Menascé, V. A. F. Almeida, and L. W. Dowdy. *Capacity Planning and Performance Modeling*. PTR Prentice Hall, 1994.
- [51] E. de Souza e Silva and R.R. Muntz. Queueing Networks: Solutions and Applications. *Stochastic Analysis of Computer and Communication Systems*, 1990.
- [52] A.E. Conway and N.D. Georganas. *Queueing Networks - Exact Computational Algorithms: A Unified Theory Based on Decomposition and Aggregation*. The MIT Press, 1989.
- [53] M. Reiser. Mean-Value-Analysis and Convolution Method for Queue-Dependent Servers in Closed Queueing Networks. *Performance Evaluation*, 1:7–18, 1981.

- [54] Theodore S. Rappaport. *Wireless Communications - Principles and Practice*. Prentice Hall PTR, 1996.
- [55] Victor O. K. Li and Wanjiun Liao. The Split and Merge Protocol for Interactive Video-on-Demand. *IEEE Multimedia*, 4(4):51–62, Dezembro 1997.
- [56] Asit Dan, Perwez Shahabuddin, Dinkar Sitaram, and Don Towsley. Channel Allocation under Batching and VCR Control in Video-on-Demand Systems. *Journal of Parallel and Distributed Computing*, 30:168–179, 1995.
- [57] Jayanta K. Dey-Sircar, James D. Salehi, James F. Kurose, and Don Towsley. Providing VCR Capabilities in Large-Scale Video Servers. In *Proceedings of the 2nd ACM International Conference on Multimedia*, pages 25–36, San Francisco, CA, 1994. ACM.