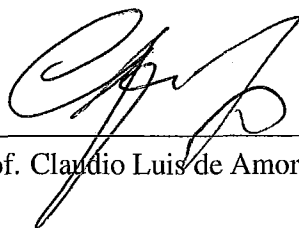


IMPLEMENTAÇÃO DE UMA SOLUÇÃO INTEGRADA PARA TRANSMISSÃO DE
FLUXOS EM SISTEMAS DE VÍDEO SOB DEMANDA

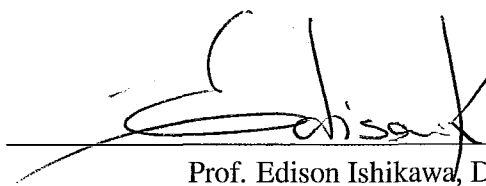
João Mauricio de Oliveira Alves

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

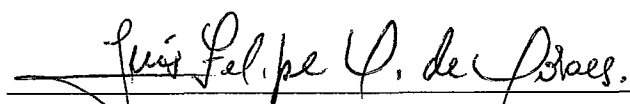
Aprovada por:



Prof. Claudio Luis de Amorim, Ph.D.



Prof. Edison Ishikawa, D.Sc.



Prof. Luis Felipe Magalhães de Moraes, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2007

ALVES, JOÃO MAURICIO DE OLIVEIRA

Implementação de uma Solução Integrada
para Transmissão de Fluxos em Sistemas de Ví-
deo sob Demanda [Rio de Janeiro] 2007

XV, 125 p. 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia de Sistemas e Computação, 2007)

Dissertação – Universidade Federal do Rio de
Janeiro, COPPE

1 - Sistemas Multimídia

2 - Vídeo sob Demanda

3 - Cache

4 - Escalabilidade

I. COPPE/UFRJ II. Título (série)

*Aos que acreditaram no impossível e contribuíram para que ele se
tornasse realidade*

Agradecimentos

Muitos foram os que contribuíram, de uma forma ou de outra, durante o curso de Mestrado e a elaboração desta dissertação. Dedico especial agradecimento à minha esposa, Tatiana, que esteve do meu lado este tempo todo, incentivando quando os ânimos esgotavam, chamando à responsabilidade quando o nível de esforço não era adequado ou simplesmente sorrindo quando tudo parecia perdido. Sua força foi o que me fez, e me faz, prosseguir. Sem ela, nada disso aconteceria.

Meus pais, Marli e João, tiveram também uma participação fundamental por sempre acreditarem que seria possível. Nossas longas conversas e o exemplo que sempre pude encontrar neles, seja no campo da ética, da dedicação, da atenção, da esperança ou da fé, foram essenciais para que eu possa perseguir ser uma pessoa um pouco melhor a cada dia. Conseguir este objetivo é outra história, mas isso não é culpa deles.

Minha família como um todo, e aí incluo a família que ganhei de presente ao conhecer minha esposa, ajudou com sua compreensão dos longos períodos de ausência, por sua torcida, algumas vezes à distância, outras bem de perto. Os caminhos da vida são difíceis e algumas vezes os que nos apoiam não podem estar sempre por perto ou nós não podemos estar perto deles. Mas a certeza de que há um porto seguro em algum lugar nos conforta e dá forças para seguir.

Meu orientador, Prof. Claudio Amorim, foi fundamental na condução deste trabalho. Não só na tarefa de mostrar o caminho acadêmico a ser seguido mas também como “psicólogo” nos momentos cruciais. As partes boas e bem colocadas nesta dissertação têm sua contribuição irrevogável. As partes ruins foram os momentos em que eu não consegui entender direito o que era para fazer.

Um agradecimento especial aos amigos do Laboratório de Computação Paralela, em particular aos membros do grupo de Multimídia, Leonardo Pinho, Lauro Whately e

Leonardo Bragato. Todos os amigos de LCP ajudaram muito nesta dissertação e o grupo que o Prof. Amorim conseguiu montar no laboratório é excelente de se conviver. Pessoal e profissionalmente. Valeu mesmo e Tá Lindo!

Os amigos, desde os mais antigos até os mais recentes, foram determinantes para a conclusão desta dissertação. Alguns ajudaram na revisão do conteúdo ou da forma, no apoio psicológico e até na “criação” de tempo. Outros amigos participaram no incentivo e acreditando que as dificuldades estavam exageradas, que o término era só questão de tempo. O esforço foi grande mas receber esta fé no resultado positivo ajudou muito. Injusto seria citar apenas alguns aqui, mesmo que uns poucos tenham sido citados em outros parágrafos. De qualquer forma, obrigado a todos. Mesmo!

Aos professores do PESC que contribuíram com sua experiência, dedicação e excelentes aulas. Em particular aos professores Felipe França, Walmir Barbosa, Paulo Veloso, Claudio Amorim e Vitor Costa. Aos membros da banca, Prof. Edison Ishikawa e Prof. Luis Felipe Moraes, pelo tempo e atenção dedicados à análise desta dissertação.

Ao CNPq e à Finep, que patrocinaram parcialmente este trabalho.

Alguns não consideram adequada a mistura entre ciência e religião. Eu considero que somos todos pessoas integrais e que a nossa vivência e personalidade são unas e indivisíveis. Desta forma, não posso deixar de agradecer Àquele que zelou por mim e por todos nós, e que mostrou sua influência em diversas e inúmeras oportunidades, além de qualquer esperança e a favor de todas as coincidências. Nas vezes em que tudo parecia perdido, quando as minhas forças não eram mais suficientes. Deus colocou um dos seus anjos mencionados acima, ou outros que não me dei conta, em meu caminho.

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

IMPLEMENTAÇÃO DE UMA SOLUÇÃO INTEGRADA PARA TRANSMISSÃO DE FLUXOS EM SISTEMAS DE VÍDEO SOB DEMANDA

João Mauricio de Oliveira Alves

Setembro/2007

Orientador: Claudio Luis de Amorim

Programa: Engenharia de Sistemas e Computação

Neste trabalho as técnicas de CVC (*Cooperative Video Cache* - Cache de Vídeo Cooperativa) e C-CVC (*Collapsed CVC*) são integradas para melhorar a usabilidade e a escalabilidade de uma solução para distribuição de Vídeo sob Demanda em ambientes de Rede Ampla (WAN) e Rede Local (LAN).

Algumas questões sobre garantia de Qualidade de Serviço são discutidas, preparando as fundações para uma implementação futura de uma solução de Vídeo sob Demanda tolerante a falhas, utilizando as já mencionadas técnicas do projeto GloVE, a CVC e C-CVC.

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

AN INTEGRATED SOLUTION IMPLEMENTATION FOR STREAMING IN VIDEO
ON DEMAND SYSTEMS

João Mauricio de Oliveira Alves

September/2007

Advisor: Claudio Luis de Amorim

Department: Computing and Systems Engineering

In this work, the CVC (Cooperative Video Cache) and the C-CVC (Collapsed CVC) techniques are merged in a single implementation. They are applied altogether to improve the usability and scalability of a Video on Demand (VoD) solution to be used in WAN (Wide Area Network) and LAN (Local Area Network) environments. Some issues about Quality of Service guarantees are discussed, laying the foundation of a future fault-tolerant VoD implementation using the GloVE project techniques, the already mentioned CVC and C-CVC.

Sumário

Resumo	vi
Abstract	vii
Lista de Figuras	xi
Lista de Tabelas	xiv
1 Introdução	1
1.1 Introdução	1
2 Redes de Distribuição de Mídia Contínua	6
2.1 Introdução	6
2.2 Rede de Distribuição de Conteúdo	6
2.3 Transmissão em árvore	8
2.3.1 Introduzindo tratamento de falhas	10
2.3.2 Distribuição em árvore com possibilidade de balanceamento	12
2.3.3 Requisitos do cenário	12
2.4 Situações de mobilidade	13
2.4.1 Distribuidores integrados aos APs	14
2.4.2 Uma variação na conexão sem fio	16
2.4.3 Requisitos do cenário	17
2.5 Transmissão <i>peer-to-peer</i>	18
2.5.1 Requisitos do cenário	19
2.6 Consolidando os cenários de transmissão	19
2.6.1 Simplificando a representação	22
2.7 Conclusões	24

3	Versões do Projeto GloVE	25
3.1	Introdução	25
3.2	Atendendo clientes em uma rede local - Opção <i>peer-to-peer</i>	27
3.3	Clientes em uma rede pública - Opção com distribuidores	29
3.4	Vantagens e desvantagens das versões isoladas	31
3.4.1	Vantagens	31
3.4.1.1	GloVE-LAN	31
3.4.1.2	GloVE-WAN	32
3.4.2	Desvantagens	33
3.4.2.1	GloVE-LAN	33
3.4.2.2	GloVE-WAN	34
3.5	Cenários insatisfatórios para as versões isoladas	34
4	Integrando as versões LAN+WAN - GloVE MIX	37
4.1	Introdução	37
4.2	Situação encontrada antes da integração	37
4.3	Estrutura do GloVE	39
4.3.1	Estrutura integrada	40
4.4	Supervisor	41
4.4.1	Atendimento das requisições	41
4.4.2	Armazenamento das informações	42
4.5	Distribuidor	43
4.6	Servidor	44
4.7	Cliente GloVE	45
4.7.1	<i>GloVE Client Engine</i>	47
4.7.1.1	Visão de janela deslizante	48
4.8	Adaptações para rede sem fio	51
4.9	Conclusões	52
5	Análise Experimental	54
5.1	Objetivo dos experimentos	54
5.2	Ambiente Experimental	55
5.2.1	Servidor e Supervisor	55
5.2.1.1	Clientes e Distribuidores	56
5.3	Métricas de desempenho	56

5.3.1	Fluxos oriundos do Servidor	56
5.3.2	Fluxos oriundos dos Distribuidores	57
5.3.3	Fluxos oriundos de outros Clientes	57
5.3.4	Custo de comunicação	57
5.3.5	Resumo das métricas	57
5.4	Problemas encontrados durante os experimentos	58
5.4.1	Quantidade de clientes por máquina	58
5.4.2	Estouro do <i>buffer</i>	58
5.5	Carga de Trabalho	59
5.6	Análise dos Resultados	62
5.6.1	Sobrecarga de clientes por máquina	62
5.6.2	Opções de vídeo disponíveis	63
5.6.2.1	Análise das métricas	68
5.6.3	Taxa de chegada de clientes	72
5.6.4	Tamanho do <i>buffer</i>	76
5.6.5	Custo de comunicação	81
5.7	Discussão	83
6	Qualidade de Serviço	85
6.1	Introdução	85
6.2	Garantia de não interrupção	86
6.2.1	Tolerância a falhas	87
6.2.2	Controle de Admissão	89
6.2.3	Balanceamento de Carga	90
6.2.4	Mobilidade	91
6.3	Segurança	92
7	Trabalhos relacionados	94
7.1	Introdução	94
7.2	Migração no contexto de rede	94
7.2.1	Branch connections	94
7.3	Migração no contexto de aplicação	95
7.3.1	Nogueras 2001	95

8	Conclusões e trabalhos futuros	96
8.1	Conclusões finais	96
8.2	Trabalhos futuros	97
8.2.1	Aproveitamento do Grafo de Distribuição	97
8.2.2	Grupos de Distribuidores	98
8.2.3	Tolerância a falhas	99
8.2.4	Reserva de recursos	100
	Referências Bibliográficas	102
A	Conceitos básicos	105
A.1	Introdução	105
A.1.1	Redes de computadores para entrega de mídia contínua	106
A.1.2	Utilização de redes sem fio	108
A.2	Mídia contínua sob demanda	110
A.2.1	Cenário inicial	111
A.2.2	Provendo vídeo através da rede	112
A.2.3	Colocando distribuidores próximos aos usuários	113
A.3	Provimento de vídeo como um grafo	115
A.3.1	Grafo de distribuição em redes sem fio	117
A.3.2	Componentes da distribuição de mídia contínua	118
A.3.3	Cadeia de distribuição	119
A.4	Qualidade de Serviço	120
A.4.1	Outros fatores que influenciam QoS	122
A.4.2	Tempo disponível para recuperação de falhas	123
A.4.3	Migração de streams	124

Lista de Figuras

2.1	<i>Content Distribution Network</i>	7
2.2	Transmissão em árvore, sem tolerância a falhas	9
2.3	Transmissão em árvore, com tolerância a falhas	11
2.4	Transmissão em árvore, apresentando uma falha	11
2.5	Algoritmo simples para balanceamento de carga entre distribuidores	12
2.6	Transmissão em árvore, tolerante a falhas, com possibilidade de balanceamento de carga	13
2.7	Transmissão em redes WiFi	14
2.8	Transmissão em uma rede de celulares	15
2.9	Transmissão utilizando CINAPS	17
2.10	Transmissão <i>peer-to-peer</i>	18
2.11	Consolidação dos cenários	20
2.12	Grafo de distribuição de conteúdo	23
3.1	Cenário de uso WAN e LAN	35
4.1	Estrutura geral do GloVE-LAN	39
4.2	Estrutura geral do GloVE-WAN	39
4.3	Estrutura geral do GloVE-MIX	41
4.4	Diagrama simplificado do armazenamento do grafo no Supervisor	43
4.5	Estrutura de classes para políticas de provimento	44
4.6	Arquitetura do Cliente GloVE	46
4.7	<i>GloVE Client Engine</i>	47
5.1	Estrutura dos Experimentos	55
5.2	GloVE-MIX - Variação da quantidade de clientes/máquina - 2 clientes	64
5.3	GloVE-WAN - Variação da quantidade de clientes/máquina - 2 clientes	64
5.4	GloVE-MIX - Variação da quantidade de clientes/máquina - 5 clientes	65

5.5	GloVE-WAN - Variação da quantidade de clientes/máquina - 5 clientes . . .	65
5.6	GloVE-MIX - Variação da quantidade de clientes/máquina - 7 clientes . . .	66
5.7	GloVE-WAN - Variação da quantidade de clientes/máquina - 7 clientes . . .	66
5.8	GloVE-MIX - Variação da quantidade de clientes/máquina - 9 clientes . . .	67
5.9	GloVE-WAN - Variação da quantidade de clientes/máquina - 9 clientes . . .	67
5.10	GloVE-MIX - Variação da quantidade de vídeos - 1 vídeo	69
5.11	GloVE-WAN - Variação da quantidade de vídeos - 1 vídeo	69
5.12	GloVE-MIX - Variação da quantidade de vídeos - 8 vídeos	70
5.13	GloVE-WAN - Variação da quantidade de vídeos - 8 vídeos	70
5.14	GloVE-MIX - Variação da quantidade de vídeos - 16 vídeos	71
5.15	GloVE-WAN - Variação da quantidade de vídeos - 16 vídeos	71
5.16	GloVE-MIX - Variação da taxa de chegada - 1 cliente/min	73
5.17	GloVE-WAN - Variação da taxa de chegada - 1 cliente/min	73
5.18	GloVE-MIX - Variação da taxa de chegada - 6 clientes/min	74
5.19	GloVE-WAN - Variação da taxa de chegada - 6 clientes/min	74
5.20	GloVE-MIX - Variação da taxa de chegada - 60 clientes/min	75
5.21	GloVE-WAN - Variação da taxa de chegada - 60 clientes/min	75
5.22	GloVE-MIX - Variação do tamanho do <i>buffer</i> - 8 MB	77
5.23	GloVE-WAN - Variação do tamanho do <i>buffer</i> - 8 MB	77
5.24	GloVE-MIX - Variação do tamanho do <i>buffer</i> - 16 MB	78
5.25	GloVE-WAN - Variação do tamanho do <i>buffer</i> - 16 MB	78
5.26	GloVE-MIX - Variação do tamanho do <i>buffer</i> - 32 MB	79
5.27	GloVE-WAN - Variação do tamanho do <i>buffer</i> - 32 MB	79
5.28	GloVE-MIX - Custo da comunicação	82
A.1	Rede celular	109
A.2	Um usuário isolado	111
A.3	Provendo o vídeo através da rede	112
A.4	Provendo o vídeo para vários usuários através da rede	113
A.5	Provendo o vídeo para vários usuários com mais de um servidor	114
A.6	Distribuidores para auxiliar no provimento	114
A.7	Árvore simples	115
A.8	Árvore	115
A.9	Um grafo complexo de provimento	116

A.10 Somando as vazões disponíveis	117
A.11 Grafo de distribuição sem fio, considerando os APs	117
A.12 Grafo de distribuição sem fio, sem APs	118
A.13 Grafo de distribuição sem fio, com os distribuidores nas estações base . .	118
A.14 Distribuição como grafo linear	119
A.15 Utilização do <i>buffer</i> em uma perda de conexão	121

Lista de Tabelas

5.1	Carga de Trabalho	60
A.1	Progressão em $O(n)$ da vazão necessária	108
A.2	Um exemplo de condições de transmissão	123

Capítulo 1

Introdução

1.1 Introdução

A convergência do conteúdo de mídia contínua clássica — tais como músicas, filmes, programas de TV e similares — com as tecnologias de redes de computadores tem aberto campo de pesquisa tanto para a comunidade acadêmica quanto para companhias envolvidas na distribuição deste conteúdo. A evolução da vazão de rede que pode estar disponível em residências tem favorecido esta convergência, uma vez que conteúdos de mídia contínua, em particular vídeo, possuem requisitos elevados em termos de transmissão e exibição. Um filme armazenado em um DVD, codificado em MPEG-2 (o que é usual), requer taxas de transferência de disco a picos de 8 Mbps.

A evolução dos aparelhos domésticos tem levado a uma oferta crescente de poder computacional, não somente em computadores “de mesa” e *notebooks*, mas também, ou especialmente, em *handhelds*, *smartphones*, consoles de jogos, MP3/MP4 *players*, etc. Estes aparelhos vêm acrescentando recursos de exibição 3D, recursos gráficos e decodificadores de vídeo e áudio cada vez mais poderosos, aumentando a gama de opções para entrega de conteúdo de mídia contínua (entre outros) para os usuários.

Para que os serviços possam ser utilizados em larga escala, é necessário equilibrar alguns fatores. O **custo** da solução, tanto para o usuário quanto para o(s) provedor(es) do serviço não pode ser elevado, pois seria um componente restritivo da solução, impedindo o seu uso em larga escala. Por outro lado, a própria utilização em larga escala contribui para a redução do custo, como pode ser observado, entre outros exemplos, com a oferta de vazão de rede nas residências.

A capacidade de crescimento da solução é fundamental, pois atender a um grande número de usuários traz desafios diferentes do atendimento a um número restrito de usuários, ou oferecer um leque pequeno de opções. A transmissão de TV atinge a um

grande número de usuários consumindo menos recursos que um serviço de mídia contínua sob demanda, especialmente se considerarmos a possibilidade de interatividade (avanços, pausas, recuos, etc.). Isso ocorre pois em TV há uma seleção de canais transmitidos de acordo com a programação da operadora/emissora de TV. Desta forma, o leque de opções é restrito e a transmissão se dá por difusão (*broadcast*), utilizando uma quantidade menor de recursos.

Uma aplicação de mídia contínua **sob demanda** precisa oferecer um leque de opções maior que o de uma transmissão de TV. A quantidade de conteúdo até pode ser a mesma porém o **momento** em que os usuários começam a assistir ao conteúdo caracteriza, a grosso modo, uma variação neste conteúdo. Um usuário pode estar assistindo a um filme no início enquanto outro assiste ao mesmo filme uma hora à frente. Outro aspecto complicador é que em uma transmissão de TV, os programas são oferecidos como um único conteúdo, pois a programação do canal transmite um programa após o outro. Transmitir dois ou mais eventos simultâneos, como no caso de ter-se várias modalidades em um grande evento esportivo, requer que a emissora/operadora prepare dois ou mais canais de transmissão. Conforme o número de usuários cresce — e também a oferta de conteúdo simultâneo disponível —, a utilização de recursos pode se tornar proibitiva, inviabilizando a solução de mídia contínua sob demanda.

Assim, a **escalabilidade** da solução é especialmente importante para o uso em larga escala. Ou seja, a capacidade que a solução tem de atender a um número maior de usuários com a adição de mais recursos. Quanto melhor for a escalabilidade, menos recursos são necessários para aumentar a capacidade de atendimento. A curva que separa, distingue, um sistema escalável de um não escalável é a linear com derivada igual a 1, ou seja, para dobrar o número de usuário é preciso dobrar o número de recursos. Se um sistema tem curva acima desta linear, ele é tido como não escalável, e escalável do contrário.

Esta definição de escalabilidade **não** está relacionada à **economia de escala**, onde dobrar a quantidade de recursos não deve implicar em dobrar os custos da solução. Observando sobre o aspecto econômico, a solução é escalável se os custos da solução crescem em um ritmo menor que o aumento da sua capacidade.

A faceta mais comum das soluções de distribuição de mídia contínua sob demanda — em particular, Vídeo sob Demanda (VoD - *Video on Demand*) — é ter foco no alto consumo de vazão de rede requerida por estes serviços. A utilização de disco – seja em relação ao espaço ou à utilização de vazão de disco – é outro aspecto bastante atacado.

Como consequência da grande utilização destes recursos — além do uso de memória

—, aumentar a escalabilidade é uma das motivações na pesquisa de soluções de VoD. Grupos pequenos de usuários e/ou vídeos de baixa resolução/qualidade podem ser atendidos por soluções mais simples. Contudo, conforme a quantidade de usuários cresce, a escalabilidade da solução torna-se cada vez mais importante. Neste cenário, o projeto GloVE e suas soluções relacionadas [1, 2, 3, 4, 5, 6] oferecem uma solução de distribuição de VoD com bom custo/benefício e escalável para grandes volumes. Um ponto forte do projeto GloVE é utilizar a memória dos componentes de distribuição de forma eficiente.

Porém, conforme se avança da construção de protótipos para a transformação destes em produtos, outras questões tornam-se tão prioritárias quanto o consumo de recursos e a escalabilidade. Garantir Qualidade de Serviço (QoS - *Quality of Service*) ao usuário do serviço de VoD é um ponto importante, pois com o uso contínuo os defeitos da solução causam maior impacto que suas virtudes.

Qualidade de Serviço pode significar diversas garantias que a solução deve prover ao longo do uso do serviço. Pode significar características relativamente subjetivas, como qualidade da imagem e do som. Também pode implicar em garantias mais objetivas como vazão máxima de rede/disco que pode ser utilizada, máximo de memória utilizada nos componentes do cliente ou nos componentes de distribuição.

Uma garantia a ser oferecida ao usuário – em acréscimo às garantias inerentes a uma solução de VoD, como baixo tempo de início da transmissão, e da já mencionada escalabilidade – é a de que o vídeo poderá ser assistido até o final, sem interrupções.

Esta garantia de QoS relativamente simples gera diversas consequências na solução de VoD. Uma primeira preocupação é como garantir a transmissão no caso de falha de um componente. Tolerância a falhas é um assunto estudado por anos, tanto por soluções acadêmicas quanto por produtos corporativos. Porém, as características de aplicação de *soft real-time* presentes em uma solução de VoD – aliadas à garantia de exibição ininterrupta do vídeo – tornam escasso o tempo disponível para a recuperação de uma falha.

De forma semelhante, a utilização cada vez mais freqüente de redes sem fio traz desafios para a garantia de exibição ininterrupta. Ainda que a mobilidade do usuário na rede possa ser tratada apenas nas camadas inferiores da rede, em algumas situações a participação de VoD em nível de aplicação pode melhorar a eficiência na distribuição de VoD. Se os pontos de acesso (AP - *Access Points*) forem integrados aos distribuidores de vídeo em um único dispositivo, trocar de AP pode implicar em uma troca equivalente de distribuidor.

Quando a troca de AP implica em se perder a conectividade com o componente de

distribuição, a mobilidade dos usuários em uma rede sem fio se assemelha a um cenário de falha. Desta forma, atender a cenários de mobilidade em redes sem fio tem algumas características de atendimento a cenários de tolerância a falhas.

Uma forma de se atacar o problema da garantia de exibição ininterrupta é reservar recursos nos componentes de distribuição para atender à migração do fluxo de distribuição de um componente para outro. Esta migração poderia ser efetuada sem a reserva de recursos – algo como uma simples reconexão ao serviço de VoD – porém isso implicaria em possíveis dispêndios de tempo de *prefetch* (ou seja, uma espera inicial para preenchimento de *buffer*) nos componentes de distribuição. Com a reserva de recursos, e a consequente diferenciação de uma migração de componente e uma nova conexão, o tempo de *prefetch* pode ser reduzido.

Mesmo em situações em que se pode receber vídeo de um distribuidor ainda ativo enquanto se recebe de um novo distribuidor, a ferramenta de migração de fluxos de vídeo mostra-se útil. Quando um distribuidor sofre uma manutenção programada – ou quando se adiciona um novo distribuidor –, os clientes deste distribuidor (que sai/entra) precisam ser migrados para/de um outro distribuidor. Desta forma, a reserva planejada de recursos contribui para o balanceamento de carga da solução de VoD.

Nesta dissertação apresenta-se a integração das versões do projeto GloVE, demonstrando os benefícios da utilização das técnicas CVC e C-CVC em conjunto na redução da necessidade de recursos de um sistema de VoD. Analisa-se também a melhora em termos de confiabilidade e escalabilidade oferecidas com um gerenciamento integrado dos componentes de distribuição de vídeo. O restante do texto apresenta a seguinte estrutura. No Capítulo 2 serão descritos os cenários previstos de utilização da solução de VoD e as possibilidades de migração de fluxos. No Capítulo 3 são apresentadas as versões isoladas do projeto GloVE, com seus cenários favoráveis e desfavoráveis. No Capítulo 4 descreve-se, brevemente, a situação encontrada antes da integração. São analisadas as alterações efetuadas para integrar as versões do GloVE e quais os benefícios que a versão integrada oferece. No Capítulo 5 é efetuada a análise experimental da solução proposta, avaliando-se os ganhos de escala e os custos – em termos de uso de recursos – adicionais do gerenciamento integrado. No Capítulo 6 são descritas algumas situações de garantia de Qualidade de Serviço e os pontos que podem ser agregados a este trabalho. No Capítulo 7 são elencados trabalhos relacionados à dissertação, seja no aspecto da confiabilidade seja em relação à distribuição de mídia contínua sob demanda. No Capítulo 8 são apresentadas conclusões sobre os resultados desta dissertação, assim como indicações de trabalhos

futuros. No Apêndice A são apresentados alguns conceitos básicos sobre distribuição de média contínua sob demanda.

Capítulo 2

Redes de Distribuição de Mídia Contínua

2.1 Introdução

Distribuição de conteúdo de mídia contínua é uma aplicação cada vez mais utilizada em redes de computadores. Seu advento, porém, é anterior ao uso de redes de computadores em larga escala. Assistir a um programa de TV ou ouvir uma música no rádio são exemplos de distribuição de conteúdo sem o uso de rede de computadores. Ao buscarmos soluções para distribuição de conteúdo de mídia contínua, especialmente vídeo sob demanda, podemos nos beneficiar de conceitos mais gerais de distribuição de conteúdo.

Em particular, neste capítulo, serão analisados alguns cenários de distribuição de mídia contínua, buscando identificar similaridades entre eles. Serão descritos alguns problemas recorrentes na preparação de uma **Rede de Distribuição de Conteúdo** (CDN - *Content Distribution Network*). Em particular, serão discutidas situações de falhas na rede de distribuição e algumas formas de se contornar o problema.

2.2 Rede de Distribuição de Conteúdo

A Figura 2.1 mostra uma Rede de Distribuição de Conteúdo. O conteúdo de mídia contínua, armazenado no **repositório**, é enviado através da CDN até o **exibidor**, responsável por apresentá-lo ao usuário. Esta abstração pode ser utilizada como base para os modelos de distribuição de conteúdo de mídia contínua.

As opções a serem empregadas em cada componente são as mais diversas. No contexto de transmissão de TV por VHF/UHF, o exibidor no cliente é um aparelho comum de TV analógica. A ligação do usuário com a CDN pode ser considerada como as ondas VHF/UHF transmitidas pelo ar e mais a antena/fio ligados à TV. A CDN é com-

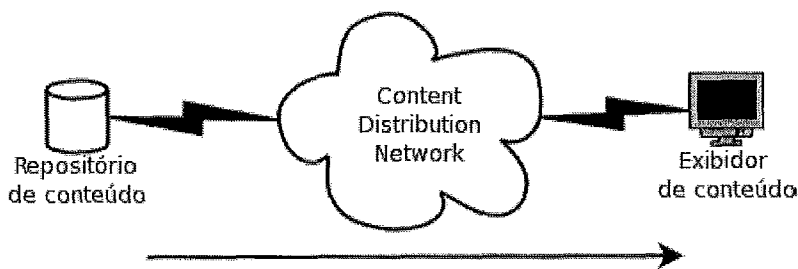


Figura 2.1: *Content Distribution Network* - Rede de distribuição de conteúdo.

posta pelos diversos equipamentos de transmissão entre a casa do usuário e a emissora, tais como antenas repetidoras e antenas transmissoras. O repositório de conteúdo está localizado na emissora de TV, contendo o programa gravado para o horário em questão. Em caso de programa ao vivo, podemos considerar que o conteúdo está sendo gerado, colocado no repositório e transmitido ao usuário simultaneamente.

Em uma transmissão de TV a cabo, o processo é parecido com a TV VHF/UHF, com a diferença de termos o exibidor no usuário ligado à CDN através de um cabo coaxial/fibra ótica.

Mesmo neste cenário simplificado, já surgem algumas preocupações de QoS (*Quality of Service*). O repositório de conteúdo precisa ter vazão suficiente para atender todos os clientes simultâneos. O mesmo ocorre com a ligação entre o repositório e a CDN e entre a CDN e o exibidor. Nos exemplos de transmissão de TV, a ligação entre a CDN e o exibidor (TV) é efetuada utilizando-se **difusão** (ou *broadcasting*). Este mecanismo faz com que o uso de um meio compartilhado (como o ar ou o cabo) seja mais eficiente. Há apenas um **fluxo** de transmissão entre a CDN e os exibidores. Uma única frequência de VHF/UHF é utilizada para transmitir o conteúdo, simultaneamente, para todos os clientes ao alcance da antena transmissora/repetidora.

Esta adoção é possível pois o conteúdo obtido pelos usuários é o mesmo, variando apenas de acordo com o canal (frequência) sintonizado. Porém, não é possível a interação do usuário com o conteúdo, como recuos e avanços¹. Também o conteúdo não pode ser obtido **sob demanda**, pois a transmissão precisa ser igual para todos os usuários. A transmissão de conteúdo inicia e termina de acordo com a programação da emissora.

Ao analisarmos cenários específicos de distribuição de conteúdo, com a utilização

¹Os televisores que agregam esta possibilidade, na verdade, adotam um modelo híbrido. O conteúdo é armazenado em disco/memória. Quando o usuário pausa, o conteúdo continua a ser recebido e armazenado. Depois é exibido do disco/memória local. A recepção **direta** do conteúdo não permite recuos e avanços.

de tecnologias de redes de computadores, podemos traçar um paralelo com este cenário simplificado. Assim é possível comparar as necessidades de QoS, especialmente aquelas relacionadas com **tolerância a falhas** e **balanceamento de carga**, e também os requisitos de **escalabilidade** da solução.

Em alguns casos, serão discutidas variações do cenário com as respectivas implicações. O objetivo desta discussão é identificar os **requisitos** que cada cenário demanda para a solução de distribuição de mídia contínua.

Os cenários abordados neste texto estão relacionados à distribuição de **mídia contínua sob demanda**. Em particular, serão tratados cenários de vídeo sob demanda, mas outras formas de mídia contínua (como música, por exemplo) poderiam ser utilizadas.

Ao final do capítulo, teremos um cenário que consolida as necessidades dos cenários citados. Estes requisitos serão levados em consideração para a solução proposta.

2.3 Transmissão em árvore

Uma implementação básica, ainda que eficiente, para entrega de vídeo sob demanda é o modelo de **transmissão em árvore** (Fig. 2.2), adotado, entre outros, pelo GloVE[7, 4]. Para reduzir os requisitos de vazão de rede no servidor — o repositório de conteúdo — são utilizados distribuidores, que atuam como *proxies* dinâmicos para o fluxo de vídeo. Ou seja, cada distribuidor age como um *proxy*, recebendo o vídeo do servidor, agindo como se fosse o cliente, e o fornecendo ao cliente, agindo como se fosse o servidor. É um *proxy* dinâmico pois o conteúdo que está disponível é definido de acordo com as requisições que chegam dos clientes, mudando dinamicamente.

A utilização dos distribuidores pode aumentar a espera do usuário pelo início do vídeo, uma vez que este será solicitado ao distribuidor, que solicitará ao servidor. O *buffer* do distribuidor será preenchido e depois o vídeo transmitido ao cliente. Contudo, o processo utilizado pelo distribuidor [4] tenta evitar este atraso adicional. Inicialmente é efetuada uma carga do conteúdo dos vídeos. Assim, quando o usuário solicita um vídeo, este já está disponível no distribuidor.

Se cada vídeo for solicitado uma única vez, o distribuidor reduz a espera inicial pelo vídeo, pois já pode provê-lo imediatamente. Solicitações posteriores podem utilizar o conteúdo armazenado no *buffer* do distribuidor, reduzindo a demanda pelo servidor.

O espaço disponível para o *buffer* do distribuidor é limitado, seja pela quantidade de memória ou, eventualmente, pelo espaço em disco disponível. Se os vídeos forem

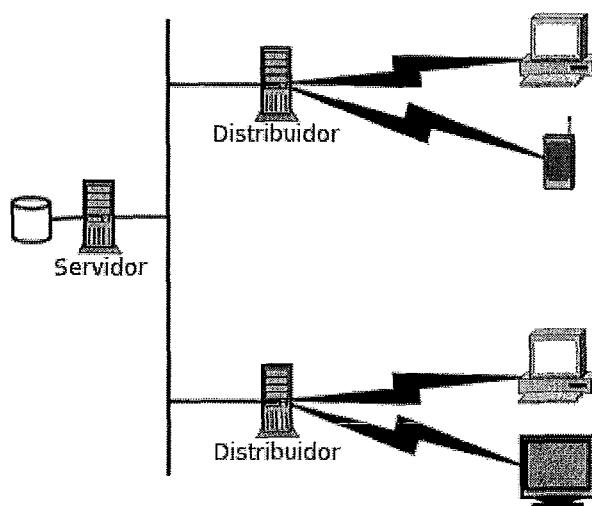


Figura 2.2: Transmissão em árvore, sem tolerância a falhas - A conexão do Distribuidor com os clientes descrita nesta figura é lógica e não física. A ligação não é necessariamente direta entre os Distribuidores e seus clientes

igualmente solicitados, ou seja, tiverem a mesma **popularidade**, este espaço será utilizado de forma equânime. Contudo, num cenário real, onde alguns vídeos são mais populares que outros, é mais adequado que os conteúdos mais utilizados utilizem mais espaço que os conteúdos menos utilizados. O projeto do distribuidor prevê esta adequação **dinâmica** às necessidades do sistema.

Os vídeos populares, que são os mais requisitados, geram grande demanda aos servidores. Em particular, **horários populares** também pressionam a demanda aos servidores, uma vez que vários usuários tendem a utilizar o sistema em horários próximos, acessando um mesmo conteúdo. O anúncio do lançamento de um novo sucesso do cinema pode fazer com que o sistema parasse de responder, dada a demanda por este conteúdo nos minutos seguintes ao anúncio. Ou no “horário nobre”, quando boa parte dos usuários está em casa após a jornada de trabalho.

Pelo modelo adotado pelo distribuidor, vídeos populares em horários populares **augmentam** a eficiência do sistema, uma vez que concentram usuários em pontos próximos de um mesmo conteúdo. Assim, a probabilidade do conteúdo já estar disponível em um *buffer* do distribuidor aumenta, levando a uma redução de demanda pelos servidores.

O problema transfere-se para os recursos (CPU, memória, vazão de rede) necessários ao distribuidor. Contudo, como não precisamos de equipamentos especiais — com discos de alta vazão e de grande capacidade, por exemplo — para os distribuidores, **escalar** o sistema pode ser uma questão de adicionar novas máquinas como distribuidores.

Se os distribuidores forem modificados para podermos utilizar, **quando aplicável**, um mecanismo de difusão (*broadcast*) ou difusão seletiva (*multicast*), poderemos reduzir a vazão de rede necessária entre a CDN e os exibidores. Por exemplo, se um distribuidor atendendo a uma rede local tem vários exibidores assistindo, em salas distintas, à transmissão de uma conferência, poderíamos utilizar *multicast* para a entrega do conteúdo a esses clientes.

Neste cenário, o QoS está limitado à exibição do vídeo para o usuário. Em princípio, cada distribuidor atende a um grupo de usuários. Se um dos componentes encerra abruptamente, o serviço é comprometido. Se o servidor falha, os distribuidores podem atender aos clientes para alguns trechos de alguns conteúdos, aqueles presentes em seu *buffer*. Contudo, como não armazenam o vídeo inteiro, não há garantia de continuidade do serviço. Se um distribuidor falha, os clientes atendidos por ele não terão acesso aos conteúdos.

2.3.1 Introduzindo tratamento de falhas

Para aumentar a tolerância a falhas nesses casos, podemos tratar os diversos distribuidores como possíveis candidatos a atender a qualquer usuário. Quando o exibidor precisa de um provedor para o conteúdo, o distribuidor mais próximo é escolhido. Se esse distribuidor falhar, o exibidor solicita um outro provedor. Como o distribuidor escolhido não estará disponível, é alocado o distribuidor seguinte mais próximo. Com os distribuidores conectados através de uma rede de distribuição (conforme visto na Fig. 2.2), a penalidade de se utilizar um outro distribuidor é pequena. Como a conectividade nesse cenário é a mesma entre os componentes, a transmissão a partir de um outro distribuidor é imediata, desde que este tenha o conteúdo em seu *buffer*.

Se considerarmos que a rede não falha — ou que o tratamento de uma falha de rede está fora do escopo deste trabalho — para se recuperar da falha do servidor, a configuração apresentada não nos oferece solução. É necessário ter, no mínimo, mais um servidor no sistema. Assim, caso o servidor que atende a um distribuidor falhe, a solicitação pode ser redirecionada para o outro servidor ainda operante.

Desta forma, para se ter algum tratamento de falhas, a configuração mínima sugerida é a mostrada na Figura 2.3. Quando um distribuidor falha (Fig. 2.4), o distribuidor alocado a um outro grupo de clientes assume a transmissão do restante do conteúdo.

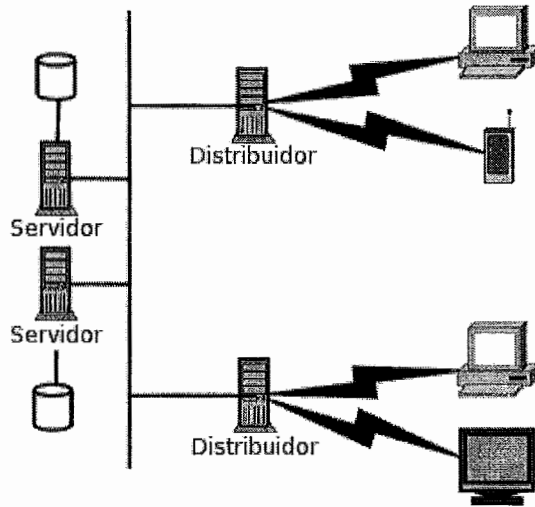


Figura 2.3: Transmissão em árvore, com tolerância a falhas

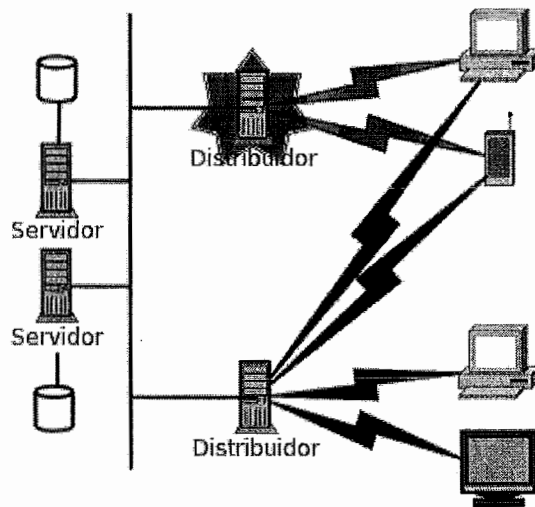


Figura 2.4: Transmissão em árvore, apresentando uma falha - A ligação representada entre um Distribuidor e seus clientes é lógica e não física. Assim, a falha de um Distribuidor não implica na reconexão dos clientes à rede mas somente na associação com outro Distribuidor

```

obterDistribuidor(idConteudo, posicaoConteudo) {
  se (há distribuidor provendo o vídeo idConteudo na posicao posicaoConteudo) {
    retornar distribuidor;
  } senão {
    retornar distribuidor com menor carga;
  }
}

```

Figura 2.5: Algoritmo simples para balanceamento de carga entre distribuidores

2.3.2 Distribuição em árvore com possibilidade de balanceamento

Para generalizar a estrutura de tolerância a falhas, podemos considerar os distribuidores alocados segundo outros fatores que não de proximidade geográfica (ou na rede). Se os distribuidores estiverem em uma LAN, atendendo a mais de um grupo de usuários, estes podem ter a sua carga **balanceada** pelo conteúdo que já possuem. Na Figura 2.5, temos um algoritmo simples de balanceamento de carga, que considera se o distribuidor já possui o conteúdo, no ponto em que o usuário está interessado. A carga pode ficar desbalanceada, se muitos usuários estiverem assistindo ao mesmo conteúdo no mesmo ponto de exibição. Porém, o distribuidor funciona melhor[4] quando há concentração de usuários em um mesmo ponto do mesmo conteúdo, uma vez que a quantidade de fluxos oriundos do servidor pode ser reduzida. Isso, é claro, se houver recursos disponíveis no distribuidor em questão para atender ao novo usuário (vazão de rede, CPU, memória, etc.).

Na Figura 2.6 temos uma estrutura do sistema que permite a tolerância a falhas e o balanceamento de carga.

2.3.3 Requisitos do cenário

Para atender ao cenário descrito nesta seção, a solução proposta deve oferecer as seguintes possibilidades:

- Escolher qual o distribuidor mais adequado a transmitir o vídeo, dado um conteúdo e uma posição inicial de exibição;
- Permitir que o usuário receba conteúdo de um distribuidor diferente, em caso de falha;
- Detectar a falha em um componente e delegar as atividades deste para outro(s) componente(s) equivalente(s);

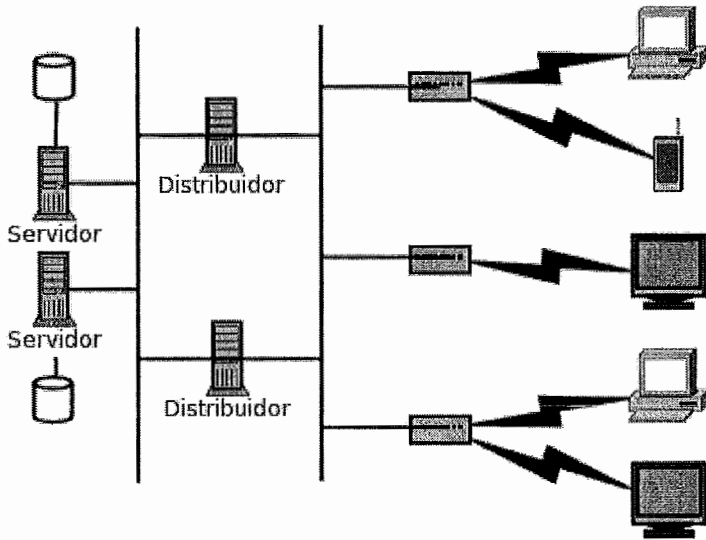


Figura 2.6: Transmissão em árvore, tolerante a falhas, com possibilidade de balanceamento de carga

- Balancear a carga entre distribuidores/servidores, considerando a vazão de rede e o uso de outros recursos (memória, disco, etc.);
- Otimizar o uso de recursos pelos distribuidores. Em alguns casos, será mais benéfico não balancear a carga nos distribuidores para reduzir o número de fluxos oriundos do servidor. Em particular, quando um usuário solicita um conteúdo que um distribuidor já possui em cache, é melhor alocar o novo fluxo do cliente neste distribuidor — que poderá servi-lo imediatamente — do que alocá-lo em outro — que necessitará receber o conteúdo do servidor primeiro, para depois atender ao usuário;
- Minimizar, ou mesmo eliminar, as situações em que o usuário terá a exibição do conteúdo interrompida, especialmente quando esta interrupção for ocasionada por uma falha de um componente do sistema.

2.4 Situações de mobilidade

O usuário pode empregar equipamentos sem fio para a exibição do conteúdo. Uma das opções mais comuns é o uso de redes WiFi (Fig. 2.7). Neste caso, o exibidor está tipicamente utilizando um dispositivo móvel (notebook, PDA, iPod[®]), trazendo algumas considerações adicionais ao problema. O usuário pode se mover, indo do raio de alcance de um

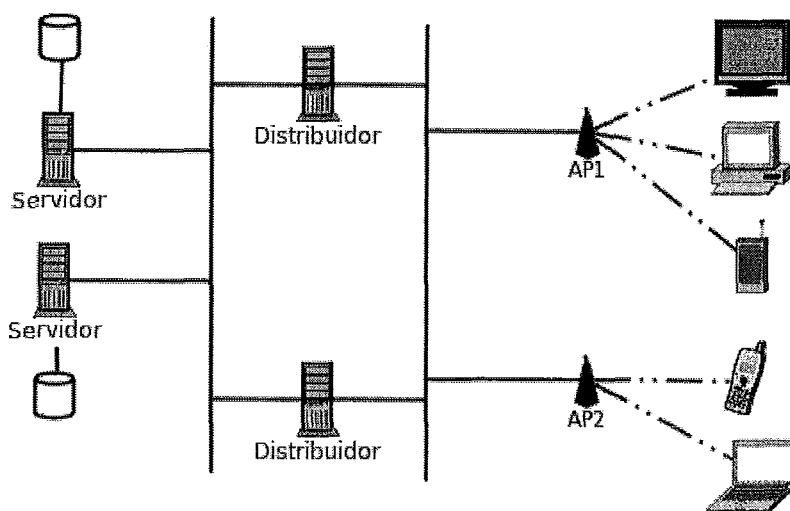


Figura 2.7: Transmissão em redes WiFi

Ponto de Acesso (AP - *Access Point*) para o de outro. Esta ação pode causar alterações na CDN ou não. Se a modificação ocorre apenas nos níveis mais baixos do protocolo de comunicação (abaixo da camada de rede), o sistema de distribuição de conteúdo de mídia contínua pode ignorar a mudança, tratando, quando for o caso, apenas os pacotes perdidos durante esta **migração** (ou *hand-off*). Se o endereço do exibidor é alterado (troca de endereço IP, por exemplo), a aplicação pode ter que executar alguma reconfiguração ou realocação de distribuidores. Usualmente, esta realocação não é necessária, bastando informar ao distribuidor que o destinatário do conteúdo foi modificado, uma vez que a conectividade não foi alterada.

2.4.1 Distribuidores integrados aos APs

A configuração do sistema pode ser diferente, forçando uma migração do provedor de conteúdo dentro da CDN. Em redes internas esta situação é mais difícil de ser verificada uma vez que a ligação entre os distribuidores e os pontos de acesso se faz via rede local. Deste modo, para o distribuidor, a transmissão através do ponto de acesso A ou B é indiferente. Por outro lado, em redes sem fio de escala mais ampla, podemos ter configurações diferentes. Em uma rede de celulares, a configuração do sistema poderia ser igual a de uma rede sem fio com pontos de acesso (Fig. 2.7). Porém, como a quantidade de usuários em uma célula pode ser maior que em uma rede interna, uma opção é alocar um distribuidor diretamente a cada Estação Rádio Base - **ERB** (Fig. 2.8). Desta forma, a vazão de rede entre o distribuidor e os clientes seria limitada somente pela vazão de rede disponível

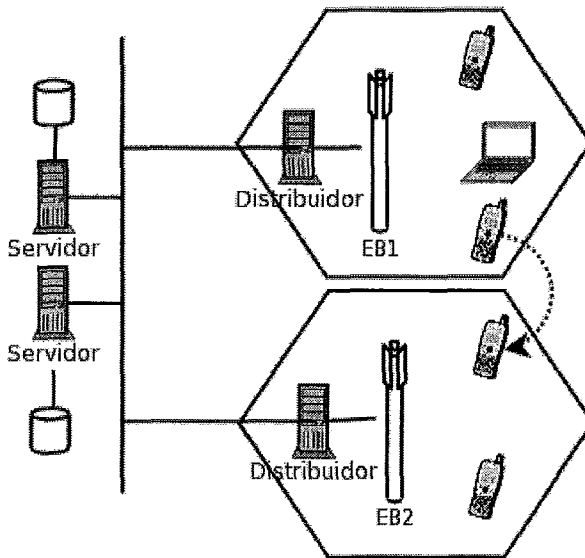


Figura 2.8: Transmissão em uma rede de celulares

entre os aparelhos celulares e a ERB. Este cenário só faz sentido se a utilização prevista considerar que é alta a probabilidade de vários usuários de uma mesma célula estarem assistindo a conteúdo de mídia contínua e, em especial, que haverá concentração em alguns itens de conteúdo. Podemos citar, como exemplo, a transmissão de um importante evento esportivo.

Quando um usuário migra de uma célula para outra, é mais eficiente trocá-lo, ou melhor, **migrá-lo** de um distribuidor para outro. Assim, continuaríamos utilizando o distribuidor conectado diretamente à ERB que atende à célula em que o usuário está.

Colocar o distribuidor associado à ERB ou ao ponto de acesso (AP) não é uma situação muito distante da realidade. Atualmente encontram-se alguns APs que integram algumas funções, dentre as quais podemos citar: *firewall*, roteador de rede fixa, servidor de impressão, servidor de arquivos. Esses aparelhos demonstram que agregar funcionalidade a um aparelho de uso geral seria factível. Agregando mais memória e o software do distribuidor, poderíamos ter uma solução alternativa para a criação de uma rede de acesso à conteúdo de mídia contínua. Nesta situação, a migração do usuário para a área de cobertura de outro AP — como seria o caso de um conjunto de APs em um saguão de aeroporto — nos levaria a desejar a migração de distribuidor. Caso contrário, em lugar de utilizar a transferência via barramento interno (ou memória) do equipamento, teríamos que receber conteúdo de um outro AP, através de uma rede cabeada (ou, o que seria pior, através da rede sem fio).

Na Figura 2.8, consideremos as ERBs EB1 e EB2 e seus respectivos distribuidores como “pontos de acesso” inteligentes para mídia contínua — formando, cada par, um único equipamento. Consideremos ainda a migração do usuário da “célula” atendida por EB1 para a “célula” atendida por EB2, sem a respectiva migração do fluxo para o distribuidor correspondente. Neste caso, estaríamos utilizando a ligação LAN entre EB2 e EB1, além de alocar recursos (possivelmente escassos) em EB1, em lugar de utilizar a conexão de memória em EB2 para servir o conteúdo para o usuário na área de cobertura de EB2, liberando os recursos em EB1 para atender clientes na área de cobertura de EB1.

Em suma, a mudança de ponto de acesso (AP), ou ERB, pelo usuário precisa ser informada ao sistema de distribuição de mídia contínua para que este possa tomar as ações necessárias à utilização eficiente do serviço. Em alguns casos, conforme visto anteriormente, a troca de ponto de acesso não afetará a distribuição do conteúdo de mídia contínua. Assim, apenas possíveis pacotes perdidos precisarão ser tratados, se estes forem imprescindíveis para a exibição do conteúdo. Em outros casos, a mudança de AP (ou ERB) implicará em uma realocação de distribuidores, em prol de uma melhor eficiência na transmissão.

2.4.2 Uma variação na conexão sem fio

Uma opção para o tratamento da conexão sem fio é utilizar o projeto CINAPS [8] (*Cluster of INexpensive Access PointS*) para provimento de uma vazão de rede com mecanismos de garantia de largura de banda mínima gerenciada a nível de aplicação. Para a transmissão de conteúdo de mídia contínua, o CINAPS poderia ser considerado como **ligação do cliente com a CDN**, não afetando diretamente a configuração da distribuição do conteúdo em si. O código do exibidor de conteúdo precisaria estar preparado (através da utilização de uma API, por exemplo) para tratar as possíveis mudanças de associação de AP impostas pelo CINAPS. Contudo, a transmissão da mídia contínua seria potencialmente mais estável, devido à garantia de largura de banda mínima oferecida pelo sistema.

A Figura 2.9 mostra como os clientes poderiam estar conectados com a CDN utilizando o CINAPS. Na migração entre APs dentro do mesmo *cluster*, por exemplo C1, não teríamos impacto na configuração da transmissão uma vez que o endereço do cliente não seria alterado. Entretanto, ao migrarmos do *cluster* C1 para o C2 teríamos uma situação em que o endereço de rede seria modificado, nos levando a uma troca na configuração. Com os distribuidores externos ao *cluster*, não há necessidade de troca de distribuidor. Se optarmos por integrar o distribuidor com o equipamento do CINAPS, de forma parecida

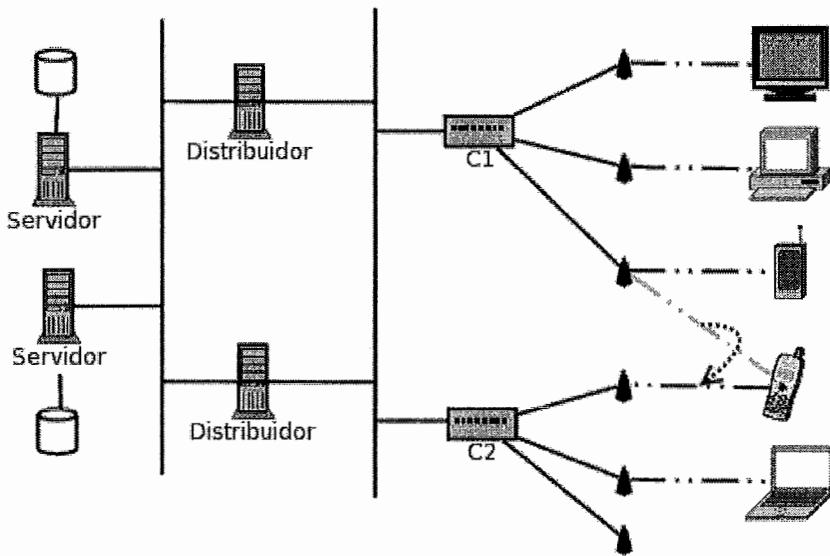


Figura 2.9: Transmissão utilizando CINAPS

com o descrito na seção 2.4.1, uma troca **intra-cluster** não afetaria a distribuição. Mas uma troca **extra-cluster** — de C1 para C2, por exemplo — pode nos forçar a também migrar de distribuidor.

2.4.3 Requisitos do cenário

Para que possamos atender ao cenário descrito nesta seção, a solução proposta deve oferecer, em adição ao descrito na seção 2.3.3, estas possibilidades:

- Identificar modificações na associação à rede sem fio, indicando quando ocorrer migrações (*hand-off*) de APs ou ERBs;
- Identificar a topologia dos equipamentos utilizados na CDN (distribuidores, servidores, APs), de forma a permitir realocações visando o uso eficiente dos recursos;
- Prever possíveis migrações de usuário, ao menos limitando quais os distribuidores (ou outros componentes) seriam candidatos para atender à migração. Idealmente, precisaríamos prever para qual componente migrar e em que momento;
- Dissociar a identificação dos clientes de seus endereços IP. O usuário pode mudar de endereço/porta e o sistema precisa identificar que se trata do mesmo usuário.

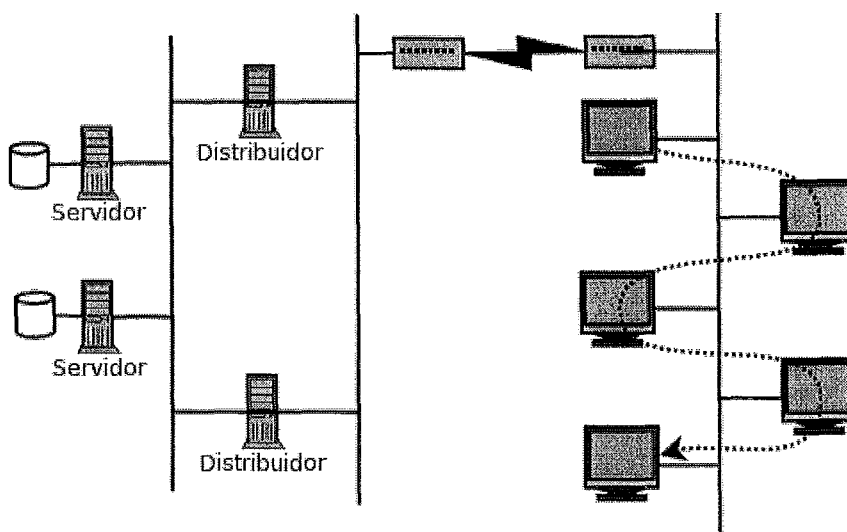


Figura 2.10: Transmissão *peer-to-peer*

2.5 Transmissão *peer-to-peer*

Para vídeos populares — ou em situações que nos levam a ter várias pessoas assistindo a um mesmo vídeo em um mesmo ponto, como a transmissão de um evento — podemos ter vários exibidores acessando ao mesmo vídeo em pontos idênticos ou próximos. Nestas situações, podemos utilizar a **transmissão *peer-to-peer*** (Fig. 2.10), onde o conteúdo do *buffer* de um exibidor é utilizado para prover vídeo para outro exibidor [9]. Em particular, esta técnica é aplicável quando há vazão de rede disponível, tanto para recepção quanto para transmissão do vídeo. Por exemplo, quando os exibidores estão ligados através de uma rede local.

Poderíamos ter um distribuidor alocado a esta porção da rede e, potencialmente, servindo aos clientes por meio de **difusão seletiva** (*multicast*). Contudo, esta alternativa nos leva a ter uma máquina dedicada para ser distribuidor. E implica em ter a rede suportando *multicast*. A utilização da transmissão *peer-to-peer* pode ser mais atrativa, especialmente por não precisar ser planejada antecipadamente. O fato dos clientes estarem recebendo o mesmo vídeo em pontos próximos pode ser identificado dinamicamente e esta otimização ser utilizada sem uma configuração prévia.

Se os conteúdos forem diferentes, ou estiverem em pontos distantes de forma a não estarem presentes no *buffer* de um outro cliente, um fluxo direto do distribuidor será utilizado, consumindo vazão de rede entre a rede de exibidores e a CDN.

Quando provendo para um outro cliente, o exibidor também faz parte da CDN. O *buf-*

fer no cliente, utilizado para tratar pacotes fora de ordem e *jitter* na rede, estará provendo conteúdo a um outro usuário. A transmissão a partir de um outro usuário na mesma rede seria preferencial à transmissão a partir de um distribuidor, por reduzir o consumo de recursos neste último e, em caso do distribuidor estar em outra rede, reduzir a demanda por largura de banda entre redes.

Este cenário, contudo, apresenta uma restrição. Como estamos prevendo interação do usuário com a exibição, permitindo recuos, avanços, pausas e etc, o ponto do vídeo disponível no *buffer* do exibidor pode ser modificado. Nestes casos, o exibidor que recebe o vídeo precisa identificar esta alteração — ao parar de receber vídeo, por exemplo — e solicitar um novo provedor. Este pode ser um outro exibidor na mesma rede ou um distribuidor.

Esta situação é parecida com o caso de falha de um componente, pois implica em uma migração de provimento de um componente para outro.

2.5.1 Requisitos do cenário

Os requisitos do cenário são semelhantes aos descritos na seção 2.3.3.

2.6 Consolidando os cenários de transmissão

Para que possamos identificar como os diversos requisitos apresentados nos cenários anteriores podem ser atendidos por uma solução de distribuição de mídia contínua sob demanda, a Figura 2.11 consolida os exemplos apresentados.

O exemplo consolidado está separado em regiões, para simplificar a associação com os cenários anteriores. A **região R1** apresenta uma célula em uma rede de celulares, com uma ERB sem modificações. Um usuário que deseje receber um conteúdo de mídia contínua precisa se conectar a um distribuidor conhecido na rede (ex. D1) para que possa receber o conteúdo. Cada usuário recebe um fluxo direto de um distribuidor, independente de estar assistindo ao mesmo conteúdo em ponto próximo ou não. No exemplo, o mesmo distribuidor foi alocado para que o fluxo oriundo do servidor S1 pudesse ser utilizado por dois clientes atendidos por D1, otimizando o uso de recursos pelo sistema.

A **região R2** mostra uma célula em que a estação rádio base foi integrada a um distribuidor (D4). Isto ocorre pois, em princípio, o custo de comunicação de D4 com sua respectiva estação rádio base é consideravelmente menor que o de qualquer outro distribuidor. Dependendo das circunstâncias, a concentração de um conteúdo na célula aten-

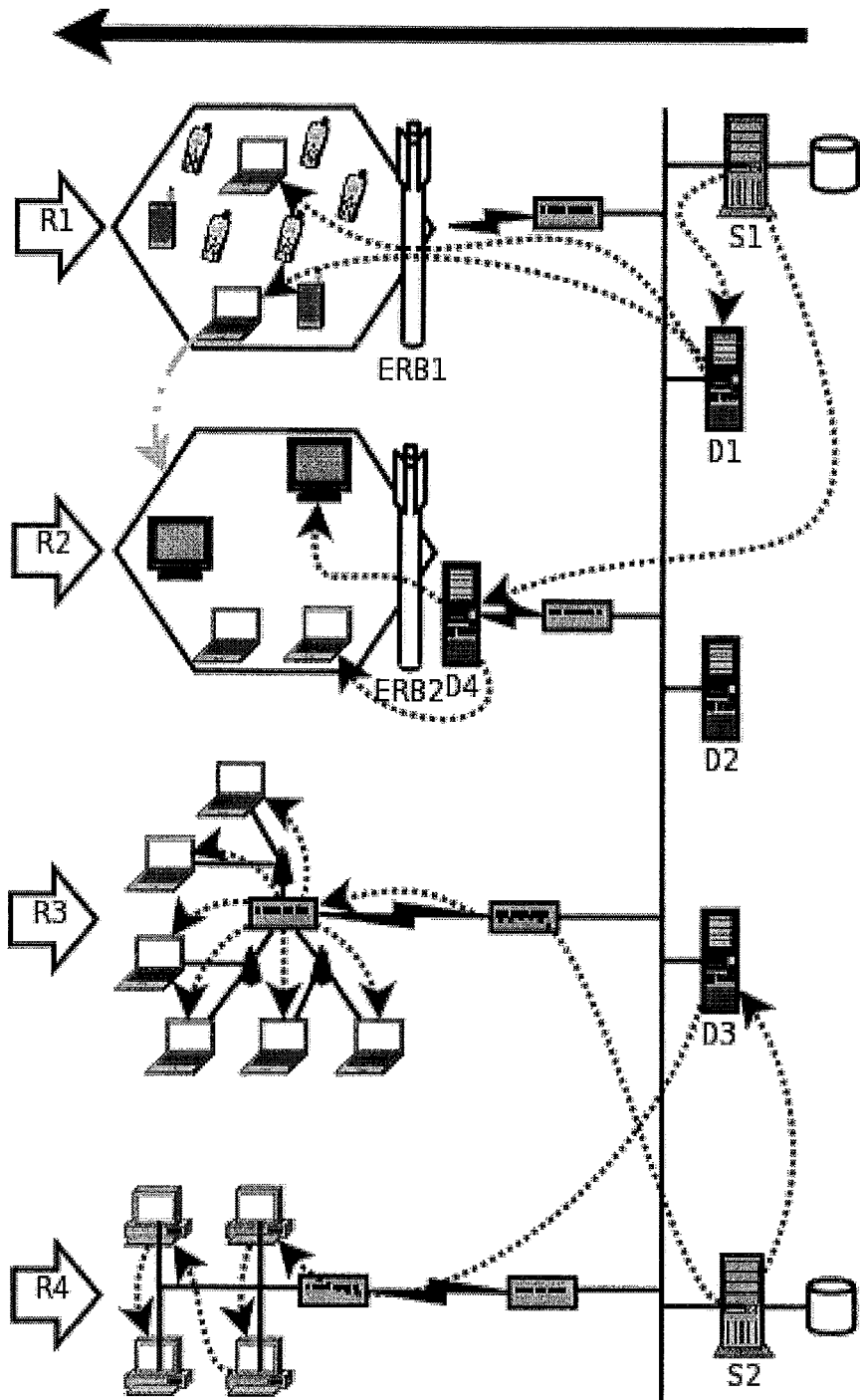


Figura 2.11: Consolidação dos cenários. R1 - Estação rádio base; R2 - Estação rádio base com distribuidor integrado; R3 - CINAPS com distribuidor integrado; R4 - Rede local com distribuição *peer-to-peer*

didada por D4 pode ser maior que em outra célula. Assim, podemos ter vários usuários assistindo ao mesmo conteúdo, mesmo em pontos esparsos. Com a concentração destes usuários em D4, teremos mais recursos disponíveis para este conteúdo, potencializando a redução dos fluxos necessários a partir de um servidor. Usuários atendendo a uma conferência poderiam assistir às palestras em seus notebooks, quando a sala da palestra em questão estiver lotada. Espectadores de um evento esportivo podem assistir as provas mesmo fora do estádio. Em todos esses casos, os usuários nesta célula serão atendidos por D4, exceto em caso de não haver recursos disponíveis em D4 para o atendimento, ou quando este falhar.

Na **região R3** temos um *cluster* de pontos de acesso utilizando o CINAPS. No exemplo, o *cluster* foi integrado a um distribuidor, cabendo os comentários apresentados para região R2. Se optarmos por **não** integrar o distribuidor, teríamos um caso parecido com a região R1.

Para a **região R4**, consideramos uma rede local com possibilidade de transmissão *peer-to-peer*. Considerando o mesmo conteúdo em ponto próximo, podemos ter um único fluxo do distribuidor (no caso D3), e a transmissão em cadeia (neste exemplo) para os demais clientes. Assim, a utilização de vazão de rede através de WAN pode ser reduzida. Se um determinado usuário efetuar uma iteração no conteúdo (ex. recuo), teríamos potencialmente um novo fluxo oriundo do mesmo distribuidor (D3). Por questões de balanceamento de carga, poderíamos alocar D2 para o provimento deste novo fluxo. D4 ou o distribuidor integrado na região R3 não deveriam ser escolhidos, pois eles **devem** ser reservados para atender às suas regiões de ação (R2 e R3, respectivamente). D1 ou D3 poderiam ser escolhidos em detrimento à D2, se a parte necessária do conteúdo em questão já estivesse disponível em seus *buffers*. Assim, apesar de ficar com a carga desbalanceada, tendo D2 completamente ocioso na Figura 2.11, a utilização do sistema é mais eficiente pois temos menos fluxos oriundos de servidores. Por outro lado, precisamos considerar se esta potencial otimização não irá exaurir os recursos em D3 ou D1. Se for o caso, realmente a melhor escolha seria D2.

Nesta dissertação não é considerado o caso em que a CDN possui mais de 2 níveis ou, particularmente, quando o distribuidor recebe de outro distribuidor ou é o próprio repositório de conteúdo. Estes casos podem ser considerados extensões das situações aqui listadas. De qualquer forma, devem ser fruto de estudo para trabalhos posteriores.

2.6.1 Simplificando a representação

Para simplificar a representação dos componentes envolvidos na transmissão do conteúdo de mídia contínua, passemos a representá-los como **vértices** de um **grafo**. Em particular, adotaremos o nome de **grafo de distribuição de conteúdo** para o grafo $G = (V_G, E_G)$ onde V_G , **vértices** do grafo, é o conjunto de componentes de transmissão (distribuidores, servidores, exibidores) e E_G , **arestas** do grafo, é o conjunto das ligações existentes entre estes componentes.

A Figura 2.12 mostra a transformação da Figura 2.11 em um grafo. As arestas contínuas representam as ligações possíveis entre os componentes. Arestas tracejadas indicam provimento de conteúdo, conforme indicado na Figura 2.11. Para reduzir a poluição do grafo, arestas contínuas mais grossas foram utilizadas para representar um conjunto de arestas que ligaria um distribuidor a diversos exibidores em uma região. Por exemplo, a aresta contínua entre D1 e R1 substitui seis (6) arestas que estariam ligando D1 aos exibidores em R1 que não estão conectados por arestas tracejadas.

Apesar do modelo tratar apenas de um vídeo, a solução aplica-se para situações em que são oferecidas diferentes opções de conteúdo. Cada conteúdo oferecido terá um grafo de distribuição associado. Se um componente participa da distribuição de n conteúdos, ele é listado em n grafos de distribuição.

Algumas ligações foram omitidas da Figura 2.12. As arestas entre os distribuidores D1, D2 e D3 e as regiões R2 e R3 não foram representadas. Para atender R2 deve ser utilizado o distribuidor D4. Da mesma forma, D_{r3} atende R3. Porém, D1, D2 e D3 podem atender R2 se D4 não estiver disponível. O mesmo vale para R3 em relação a D_{r3}.

As ligações entre os exibidores da região R4 também estão ausentes. A ligação entre eles deve representar um **grafo completo**, uma vez que todos poderiam prover para todos visto que, na Figura 2.11, R4 é uma rede local. Contudo, se considerarmos que a aresta indica uma possibilidade de provimento, somente os exibidores que já possuem o conteúdo podem provê-lo a outro. Assim, se os exibidores entrarem em sequência no sistema, o grafo indicado para R4 na Figura 2.12 estaria estritamente correto.

O sub-grafo R1 também poderá ser um grafo completo se houver a possibilidade de provimento entre clientes ligados na rede sem fio.

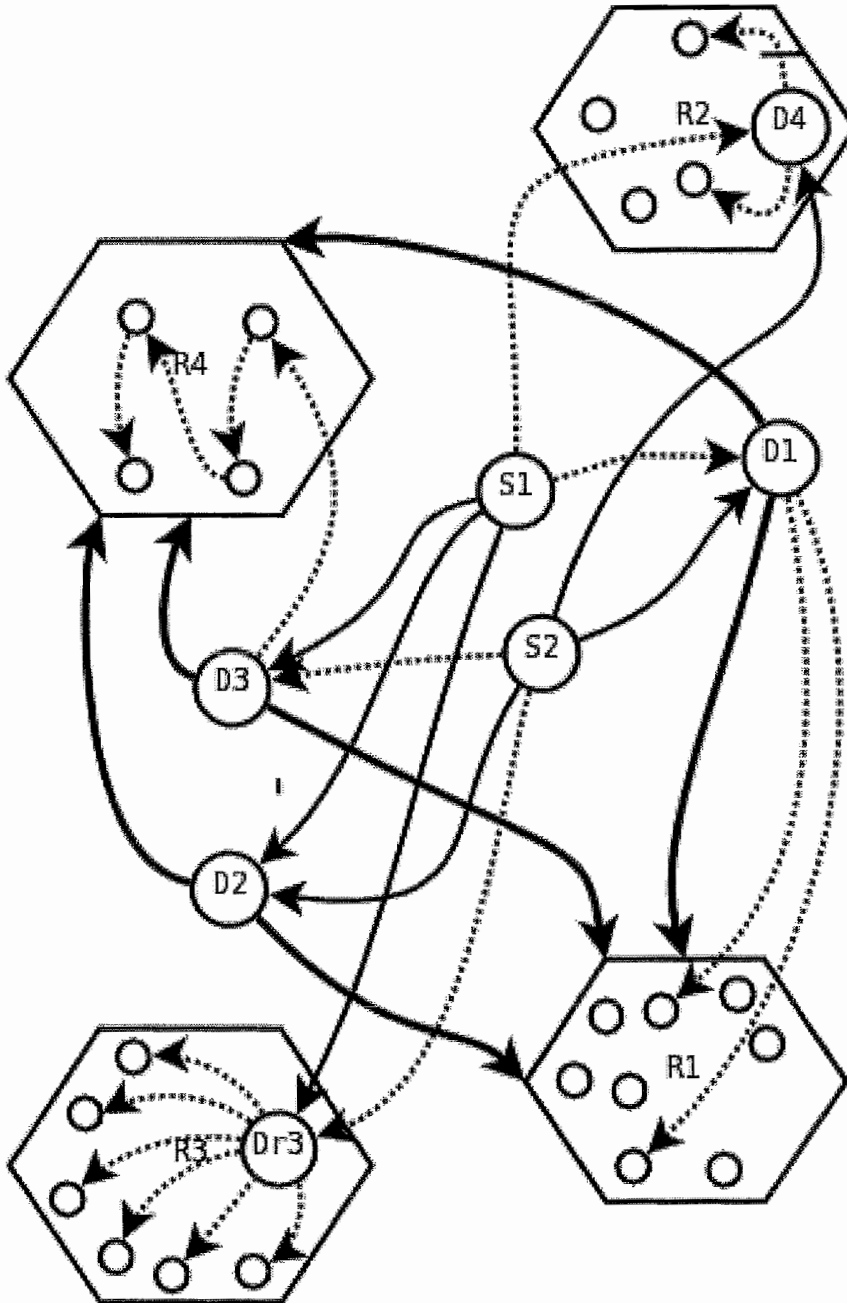


Figura 2.12: Grafo de distribuição de conteúdo. Arestas contínuas indicam ligações possíveis. Arestas tracejadas representam provimento de conteúdo em andamento. Arestas contínuas mais grossas (utilizadas para evitar uma maior poluição do grafo) indicam que um distribuidor está ligado à todos os elementos da região em questão.

2.7 Conclusões

Com base nos cenários apresentados neste capítulo, podemos listar alguns requisitos a serem atendidos pela solução proposta.

- Permitir entrada/saída de servidores no grafo;
- Permitir entrada/saída de distribuidores no grafo;
- Permitir entrada/saída de clientes no grafo;
- Ter um grafo de distribuição para cada vídeo oferecido;
- Identificar saída de Distribuidor;
- Identificar saída de Cliente;
- Identificar saída de Servidor;
- Permitir que um cliente receba de mais de um distribuidor/cliente;
- Permitir que um cliente solicite outro provedor (outro cliente/distribuidor);
- Permitir que um cliente receba solicitação de um componente de gerenciamento (ou outro componente) para receber vídeo de outro provedor
- Alterações no grafo de distribuição não devem interferir na exibição de vídeo para o usuário.

Os requisitos de tolerância a falhas e balanceamento de carga não são, diretamente, escopo deste trabalho. Contudo, os requisitos listados aqui são fundamentais para migrarmos o fluxo de vídeo de um provedor para outro. Esta migração é essencial para tolerância a falhas, pois com a quebra de um componente, digamos um distribuidor, podemos transferir os dados de um servidor para outro, sem a interrupção da transmissão para o cliente.

Para o balanceamento de carga, a entrada de um componente de transmissão (mais uma vez, um distribuidor, por exemplo) no grafo de distribuição levaria este componente a enviar vídeo para alguns clientes já associados ao grafo. Este envio ocorrerá com a migração dos clientes para o novo distribuidor. Esta migração não deve interferir na exibição do vídeo para o usuário.

Capítulo 3

Versões do Projeto GloVE

3.1 Introdução

A distribuição de conteúdo de mídia contínua, como filmes e músicas, através de redes de computadores sempre gerou grande interesse, tanto no meio acadêmico como em pesquisa corporativa. Entregar conteúdo com boa qualidade de imagem e som exige grande quantidade de recursos, tais como vazão de rede e de disco, capacidade de processamento na máquina exibidora, entre outros.

A melhoria constante nos formatos de transmissão de conteúdo têm exigido menor quantidade de recursos, em particular vazão de rede e de disco. Isso ocorre pois [10] os padrões de compressão atuais, notadamente baseados em esquemas assimétricos (maior complexidade no codificador do que no decodificador) de compressão com perdas (*lossy compression*) nos quais informações consideradas menos relevantes são removidas, estão atingindo maior compressão sem diminuir a qualidade das imagens percebida por quem as visualiza.

Em paralelo, a quantidade de recursos disponíveis é crescente, em todas as áreas. Atualmente é mais simples ter à disposição mais memória, mais vazão de rede, mais vazão de disco, mais capacidade de armazenamento, mais poder de processamento. Máquinas multiprocessadas, que eram consideradas privilégio de corporações ou centros de pesquisa, têm se tornado cada vez mais comuns. Da mesma forma, discos de grande capacidade (100 GB ou mais) são encontrados em um grande número de computadores domésticos. Assim como memórias superiores à 1 GB ou vazão de rede doméstica de 2 Mbps ou mais. E a expectativa é que esses números, representativos da realidade em 2007, sejam suplantados ano a ano.

Porém, com o mercado potencial de mídia contínua, a quantidade de usuários que um serviço deste tipo pode ter é muito grande. Assim, as soluções para distribuição de mídia

contínua precisam sempre tratar a questão da **escalabilidade**. Uma boa solução para uma quantidade pequena de usuários simultâneos pode falhar devido ao seu próprio sucesso, simplesmente o aumento no número de usuários simultâneos leva a solução ao colapso.

Soluções escaláveis para entrega de mídia contínua podem adotar modelos de distribuição de TV aberta em *broadcast*, por exemplo. Transmitir o conteúdo continuamente, com uma programação com horários estabelecidos, traz desafios para sua implantação, mas tem uma boa escalabilidade quando consideramos apenas o número de usuários. Afinal, é o mesmo conteúdo que precisa ser enviado para um número potencialmente crescente de usuários. Técnicas que utilizam *broadcasting* imitam de forma mais próxima esta transmissão de TV, atendendo a um número potencialmente ilimitado de usuários em uma determinada região geográfica¹. Porém, em redes públicas de computadores, como a Internet, este tipo de transmissão não é possível.

Os desafios da entrega de conteúdo de mídia contínua são aumentados quando se utiliza o modelo de mídia contínua sob demanda, onde o usuário escolhe o que quer assistir, no momento que mais lhe aprouver. Assim, além de haver preocupação com a entrega de conteúdo a um número crescente de usuários, também a quantidade de conteúdo disponível aumenta pois cada usuário pode estar assistindo a um mesmo conteúdo em pontos distintos. Acrescentar a possibilidade do usuário **escolher o ponto** em que quer assistir, permitindo saltos e pausas, adiciona novas considerações que uma solução de entrega de mídia contínua deve fazer.

O projeto GloVE, do Laboratório de Computação Paralela da COPPE/UFRJ, iniciado com as publicações de ISHIKAWA e AMORIM [11, 7, 3], propõe uma solução para a distribuição de **Vídeo sob Demanda** (*VoD - Video on Demand*) em redes de computadores. Em especial, esta solução se apresenta **escalável** e também custo-efetiva. Os dois modelos propostos por ISHIKAWA e AMORIM foram implementados por PINHO [12] – referente à CVC [11] – e por BRAGATO [4] – referente à C-CVC [7] –, em duas versões distintas denominadas (informalmente) de GloVE LAN e GloVE WAN, respectivamente.

O potencial de uso destas versões do GloVE é alto, mas cada uma tem o seu nicho específico. A versão LAN (ou *peer-to-peer*) é mais indicada para ambientes de redes locais, tais como dentro de uma escola ou empresa. A versão WAN (ou baseada em distribuidores de vídeo) é mais adequada para redes públicas, especialmente se estiver integrada aos equipamentos da provedora de rede (Operadora de telefonia ou TV a cabo,

¹A região é limitada pelo alcance do mecanismo de transmissão e pelo efeito que obstáculos exercem no sinal transmitido

por exemplo).

Nesta dissertação procurou-se agregar sinergia às duas versões, permitindo que a escolha do modelo de entrega de vídeo aconteça de forma automática, em uma solução integrada. Com isso, o potencial da versão integrada tende a ser maior que o das versões isoladas. Este capítulo apresenta os desafios encontrados na integração das versões e está estruturado conforme segue. A Seção 3.2 apresenta a versão GloVE LAN e alguns cenários de distribuição em redes com vazão de rede simétrica. A Seção 3.3 apresenta a versão WAN assim como cenários de distribuição de conteúdo em redes públicas assimétricas (tais como ADSL). A Seção 3.4 mostra as vantagens e desvantagens de cada versão. A Seção 3.5 descreve alguns cenários em que as versões não se adequam isoladamente.

3.2 Atendendo clientes em uma rede local - Opção *peer-to-peer*

O conteúdo desta seção é baseado em ISHIKAWA e AMORIM 2001 [11], PINHO 2002 [12] e na análise da implementação GloVE inicial, encontrada antes do início do trabalho de integração.

Uma solução de distribuição de mídia contínua encontra vários desafios a serem abordados. Um dos principais é a escalabilidade da solução, especialmente em relação à utilização de vazão de rede e disco. Esta vazão é particularmente crítica no Servidor de Conteúdo, pois este componente concentra o atendimento das requisições dos clientes.

Aumentar a quantidade de clientes atendidos pela solução envolve a gerência de uma complexa rede de distribuição de conteúdo. Contudo, em sistemas de Vídeo sob Demanda – em particular quando se prevê interatividade com o cliente (saltos, pausas, etc.) – as soluções convencionais utilizam um fluxo (*stream*) dedicado para cada cliente.

Esta abordagem faz com que a Complexidade (em termos de complexidade de algoritmos) do uso da vazão de rede seja $O(n)$, onde n é a quantidade de clientes. Assim, utilizando-se um vídeo à taxa de 2 Mbps, com 500 clientes atingimos uma vazão **efetiva** necessária de 1 Gbps. Vale ressaltar que uma rede Gibabit Ethernet oferece 1 Gbps de vazão de rede **nominal**, sendo que a vazão **efetiva** é menor.

Esta vazão de rede efetiva máxima pode não ser atingida pois há um outro gargalo a ser atravessado. A vazão de disco **nominal**, em um disco rígido topo de linha em 2007 [13], pode chegar à 125 MB/s ou 1 Gbps. Contudo, a proporção da vazão nominal que se transforma em vazão **efetiva** em um disco tende a ser menor que a vazão efetiva em uma

rede. No disco, a forma como os blocos estão armazenados e como são acessados afeta a vazão efetiva obtida, reduzindo ainda mais a proporção vazão efetiva/vazão nominal.

Adotar soluções convencionais de VoD, baseadas em servidores atendendo aos clientes da rede é possível mas, usualmente, não é viável. O custo de se ter alta vazão de rede e discos rápidos nos servidores é alto. Em acréscimo, gerenciar o conteúdo nos vários servidores e discos que uma solução para atender um número grande (acima de 1000) de usuários eleva os custos gerais da solução.

Com o artigo de ISHIKAWA e AMORIM [11] e a implementação efetiva da solução por PINHO [12], utiliza-se a abordagem *peer-to-peer* para a distribuição de Vídeo sob Demanda. A introdução da CVC (*Cooperative Video Cache*) permite que o número de clientes atendidos pela solução de VoD aumente com uma pressão menor sobre os recursos utilizados do servidor.

A CVC se baseia no princípio que os clientes necessitam ter um *buffer* para tratar o retardo médio na rede (*jitter*). Mesmo em uma rede sem *jitter*, os bytes de vídeo não poderiam ser entregues diretamente da rede para decodificação. Isso exigiria que a taxa de transmissão e de decodificação fossem **exatamente** iguais. Esta não é uma premissa razoável de se assumir.

Assim, como todos os clientes terão um *buffer* de vídeo, a CVC apresenta uma forma inovadora de gerenciar este *buffer*, de forma que um cliente possa prover a outro(s) cliente(s) sem a necessidade de um novo fluxo de vídeo oriundo do servidor. Utilizando técnicas de distribuição de vídeo tais como *Chaining* [14], a CVC utiliza em média 20% dos recursos necessários da solução convencional.

A implementação da CVC no GloVE original introduz a possibilidade de se usar servidores que não aproveitam a funcionalidade de IP *multicast* provida na camada de rede, quando disponível, mantendo a eficiência na distribuição de vídeo. O gerenciamento de cada *buffer* cliente é efetuado por um componente separado, o CVCM (*CVC Manager*). Este componente identifica qual cliente é mais adequado para prover a outro cliente, segundo heurísticas de maximização do uso dos *buffers* e a consequente redução no uso de fluxos oriundos de um servidor. PINHO [12] avalia o protótipo em uma rede local, com suporte à IP *multicast* com variações de taxa de chegada de clientes e de tamanho de *buffer* disponível.

O desempenho da solução GloVE *peer-to-peer* (ou LAN) é positivo em redes locais que permitem vazão de rede simétrica, ou seja, em que a vazão disponível para recepção e transmissão são equivalentes. Em redes com vazão de rede assimétrica (ex. ADSL),

ou onde um cliente não tem acesso a outro, a solução *peer-to-peer* proposta pelo GloVE degenera em uma solução convencional, com o atendimento efetuado pelo servidor.

3.3 Clientes em uma rede pública - Opção com distribuidores

O conteúdo desta seção é baseado em ISHIKAWA e AMORIM 2003 [7], BRAGATO 2006 [4] e na análise da implementação GloVE-WAN inicial, encontrada antes do início do trabalho de integração.

A solução *peer-to-peer* descrita na seção 3.2 é eficiente na redução dos recursos de rede necessários para a distribuição de vídeo em uma rede local. Porém, em redes públicas, algumas questões podem ser levantadas [7]:

- Aumentar o tráfego entre clientes em pontos opostos da rede pode congestionar o *backbone* da rede, impossibilitando o uso da CVC em ambientes WAN
- A CVC não é indicada para links assimétricos que são mais comuns no último trecho (*last mile*) da rede até o usuário
- A confiabilidade de um sistema de VoD baseado na CVC é baixa, devido às falhas silenciosas que podem ocorrer no cliente
- O uso da CVC pode afetar a privacidade do usuário

Com base nestas limitações, é proposta uma nova abordagem do uso da CVC, chamada C-CVC (*Collapsed Cooperative Video Cache*), em que os *buffers* de cooperação dos clientes são **colapsados** em um outro componente, chamado **C-CVC Proxy** [7] ou Distribuidor [4].

Este componente agrega (ou **colapsa**) os diversos *buffers* dos clientes em pontos de concentração de acesso na rede. Idealmente, os Distribuidores precisam estar em pontos de rede o mais próximo possível do cliente, mas não tão próximo que não possa atender outros clientes. A localização do Distribuidor deve equilibrar os objetivos de minimizar a distância – em saltos (*hops*) de rede – até os clientes e maximizar a quantidade de clientes que podem ser atendidos. Em redes ADSL, o Distribuidor deve ser instalado próximo ao DSLAM para um bom aproveitamento [4]. A grosso modo, o Distribuidor atua, para os clientes, como um servidor convencional. Porém, com algumas diferenças substanciais.

As latências e o retardo (*jitter*) agregados da rede de distribuição são minimizados pelo *buffer* no Distribuidor. O *buffer* no cliente pode ser menor, uma vez que só precisa lidar com o *jitter* entre o Distribuidor e o cliente.

O Distribuidor, por trabalhar primordialmente com conteúdo em memória, tem menos limitações de vazão que um servidor convencional. A vazão de disco necessária é menor, mesmo em se considerando (esta opção não foi implementada e avaliada) que o armazenamento de parte do *buffer* do Distribuidor em disco. Esta porção estaria relacionada com pontos do vídeo que não estão em uso pelos clientes, por estarem muito à frente ou em ponto anterior. Estes blocos podem, inclusive, ser gerenciados como um arquivo de paginação.

Outra característica distinta é que o distribuidor trabalha com configuração dinâmica, ou seja, os vídeos não são armazenados diretamente no Distribuidor. O conteúdo é obtido do Servidor e repassado aos clientes. Assim, adicionar um novo vídeo ao sistema não implica em configuração específica em cada máquina Distribuidora nova, ao contrário do que ocorre com os Servidores. Uma limitação da implementação atual é que a lista de vídeos é obtida na carga inicial, exigindo reiniciar o Distribuidor para adicionar um novo vídeo. Porém, esta não é uma limitação **estrutural**, diferentemente do que ocorre com um Servidor. Afinal, é necessário copiar o conteúdo do vídeo para cada novo Servidor, ao passo que o conteúdo no Distribuidor é obtido *on demand*.

Utilizou-se o modelo *push* na entrega de vídeo pelo Distribuidor. Assim, aproveita-se melhor a largura de banda de rede disponível, pois não é necessário arcar com o tempo de *round-trip* entre o cliente e o Distribuidor a cada bloco recebido pelo cliente. Utilizando o modelo *push*, o Distribuidor pode enviar os blocos na taxa necessária pelo cliente, que os recebe na mesma taxa (sem considerarmos o *jitter*). No modelo *pull* adotado pelo Servidor na versão GloVE LAN (ou *peer-to-peer*), cada bloco recebido é antes solicitado. O cliente precisa aguardar o tráfego da solicitação até o Servidor e o tráfego de volta do bloco de vídeo. A vazão de rede necessária é maior que no modelo adotado pelo Distribuidor. O mesmo modelo *push* é adotado pelo GloVE LAN quando um cliente provê para outro.

Por ter uma capacidade de *buffer* maior que o necessário a um cliente, e por estar (usualmente) em um ponto da rede de maior visibilidade para outros clientes, o Distribuidor tem potencial maior para reuso do fluxo oriundo do Servidor. E é mais simples de se gerenciar que um Servidor convencional. Instalar um novo Distribuidor em um ponto da rede é tão simples quanto colocar uma máquina comum (COTS - *Commercial off-the-shelf*), com significativa quantidade de memória (4 GB ou mais, em 2007). Prover

conectividade do Distribuidor com o Servidor e dos Clientes com o Distribuidor completa a configuração necessária. Um novo Servidor, por outro lado, exigiria a cópia do conteúdo dos vídeos e a manutenção dos mesmos em caso de mudança no acervo disponível.

Porém, se o Distribuidor estiver atendendo a diversos clientes, com alguns deles em uma rede local, a funcionalidade oferecida pela versão LAN pode ser atrativa. Na Seção 3.4 são abordadas algumas vantagens e desvantagens das versões LAN e WAN.

3.4 Vantagens e desvantagens das versões isoladas

A CVC (*Cooperative Video Cache*) introduziu uma solução escalável e custo-efetiva para distribuição de vídeo em redes locais. Seu primeiro protótipo, o GloVE – mencionado nesta dissertação como GloVE LAN ou versão *peer-to-peer* – demonstra que a solução é viável para uma implementação real de um sistema de VoD utilizando a CVC. Para utilização em redes públicas e/ou com vazão de rede assimétrica, introduziu-se a C-CVC (*Collapsed CVC*). A implementação desta C-CVC deu origem a uma variação do GloVE original, denominado GloVE-WAN. Este protótipo foi avaliado inclusive em uma operadora de rede pública ADSL, demonstrando que a solução é viável e ainda com custo competitivo.

Porém, a evolução do GloVE para o GloVE-WAN trouxe benefícios, mas descartou algumas de suas vantagens. Nesta seção serão apresentadas as vantagens e desvantagens no uso de cada versão.

3.4.1 Vantagens

Cada uma das versões possui cenários em que sua forma de operar é mais adequada. Estas situações são descritas nesta Seção.

3.4.1.1 GloVE-LAN

Por utilizar o conceito de *peer-to-peer*, a versão GloVE-LAN tem a característica de ter seus maiores recursos à disposição conforme o número de clientes conectados aumenta. A principal característica do serviço de VoD que é explorada pelo projeto GloVE (quaisquer versões) está relacionada com a popularidade dos vídeos. Quanto mais popular um conteúdo, mais clientes estarão assistindo ao mesmo conteúdo. Com isso, maior a probabilidade do conteúdo necessário a um cliente estar disponível no *buffer* de outro. Ou, na versão GloVE-WAN, de já estar disponível no *buffer* do Distribuidor.

Na versão GloVE-LAN a disponibilidade **geral** de recursos cresce com a entrada de mais clientes. Não somente em relação ao espaço em *buffer* mas também em relação à vazão de rede agregada. A utilização de *switches* em redes locais, especialmente com a popularização de redes locais Gigabit Ethernet, permite que a vazão **nominal** entre cada cliente em uma mesma sub-rede (ligado ao mesmo *switch*, por exemplo) seja uniforme. Assim, a cada cliente que entra a quantidade de vazão **agregada** disponível aumenta na mesma proporção. O melhor cenário para a versão GloVE-LAN é ter os clientes em uma rede interligada por um *switch* – que permite agregação na vazão de rede – assistindo a um mesmo vídeo e acessando o sistema de VoD em sequência. Assim, um dos clientes recebe do Servidor e forma-se um encadeamento no provimento de vídeo, aproveitando-se ao máximo a largura de banda de rede disponível internamente no *switch*.

3.4.1.2 GloVE-WAN

Projetado de acordo com uma modelagem voltada para redes com largura de banda assimétricas, a versão GloVE-WAN adequa-se muito bem a redes públicas com ADSL. Instalar distribuidores no último trecho de rede antes da linha do usuário (*last mile*) permite a redução de uso de recursos no Servidor de conteúdo. Como o ambiente do Distribuidor até o cliente é controlado – afinal, é necessário acordo com a operadora de rede e esta tem controle de sua rede até a casa do usuário – as necessidades de vazão de rede podem ser dimensionadas de forma mais acurada. Também se pode dimensionar a quantidade máxima de usuários simultâneos que o Distribuidor deverá atender, visto que este é limitado pelo número de usuários que estará conectado ao componente de rede (por exemplo, DSLAM).

Por adotar o modelo *push* (assim como o provimento entre clientes no GloVE-LAN), o Distribuidor utiliza melhor a largura de banda disponível. Como a ligação entre o Distribuidor e o Servidor estará usualmente em uma rede de alta velocidade (próximo ao *backbone*, por exemplo), utilizar o modelo *pull* no Servidor não é tão proibitivo e simplifica a implementação do Servidor.

A agregação dos *buffers* de CVC no Distribuidor permite que o Distribuidor possa gerenciar melhor os blocos que devem ser descartados e os que devem ser mantidos em *buffer*. Se dois clientes estão assistindo a um mesmo vídeo em pontos distintos, em princípio o Distribuidor utiliza dois fluxos do Servidor para prover estes clientes. Porém, se o Distribuidor tem recursos (memória) disponíveis, ele pode estabelecer *link slots* entre estes trechos de vídeo, fazendo com que o conteúdo localizado mais à frente não seja des-

cartado. Assim, quando o cliente que estava assistindo o vídeo mais para o início atinge o bloco que já está armazenado, o Distribuidor pode interromper um dos fluxos do Servidor, reduzindo o tráfego na rede.

3.4.2 Desvantagens

Algumas características tanto da versão do GloVE para LAN como para WAN não são tão interessantes em cenários adversos. Estas situações são descritas nesta Seção.

3.4.2.1 GloVE-LAN

Um dos fatores que motivaram a criação da versão GloVE-WAN foi a inadequação da versão GloVE-LAN a redes com vazão de rede assimétrica. Em uma rede ADSL, por exemplo, pode-se ter uma vazão nominal máxima de rede chegando ao cliente de 8192 Kbps associada a uma vazão nominal máxima de rede saindo do cliente de 500 Kbps. Neste cenário, o cliente pode receber vídeo de alta-qualidade mas não conseguirá prover um vídeo de qualidade relativamente baixa.

Também não é indicado provimento através de um servidor NAT (*Network Address Translation*), associado ou não a um *firewall*. Para que o desempenho de uma solução *peer-to-peer* seja adequado, cada um dos clientes precisa ter acesso direto ao outro. Porém, os clientes atrás do NAT usualmente aparecem para outras máquinas externas ao NAT com o mesmo endereço. E, associado a um *firewall*, usualmente só é permitido estabelecimento de conexão para máquinas específicas na rede interna.

A privacidade dos usuários pode ser comprometida, uma vez que os pacotes de vídeo serão transmitidos de um cliente para outro diretamente. O uso de um programa de monitoração identifica para qual endereço IP o vídeo está sendo enviado, permitindo que os clientes em relação de provimento/recepção possam conhecer seus endereços IP. Em uma rede local este problema não é tão grave, mas em uma rede pública esta característica pode não ser tolerada.

Como o cliente recebe vídeo de outro cliente, esta transmissão está condicionada à disponibilidade do cliente provedor. Se o cliente provedor encerra a exibição do vídeo abruptamente, o cliente receptor precisa de outra fonte de provimento. Como o cliente não está comprometido com a transmissão para outro cliente (e nem deve estar), este problema é frequente.

A situação ideal para o GloVE-LAN é ter os clientes admitidos no sistema em sequência. Se a admissão é simultânea, a probabilidade de se ter todos (ou boa parte) os

clientes atendidos pelo Servidor é maior. Este cenário pode ocorrer quando, apesar de ser um sistema de VoD, o GloVE-LAN é utilizado com características de transmissão em *broadcasting*, ou seja, combina-se um horário em que todos assistirão a um determinado conteúdo, como um vídeo de treinamento. Este cenário pode ocorrer de outra forma, com o anúncio de um novo lançamento de um filme de grande apelo de público. Ato contínuo deste anúncio, um grande número de usuário pode solicitar a transmissão, sendo admitidos de forma praticamente simultânea no sistema.

3.4.2.2 GloVE-WAN

O GloVE-WAN é, em princípio, uma evolução do GloVE-LAN. Isso indica que as situações em que o GloVE-LAN não é adequado pode ser o nicho de operação do GloVE-WAN. Efetivamente, o GloVE-WAN opera melhor em situações de rede assimétrica e rede pública que o GloVE-LAN.

Contudo, o GloVE-WAN não se aproveita da agregação de recursos oferecida pela entrada de novos clientes na rede. E como os clientes recebem vídeo do Distribuidor, a vazão de rede necessária no Distribuidor cresce linearmente ($O(n)$, onde n é o número de clientes) com a quantidade de clientes atendidos. Assim, com vídeo à taxa de 2 Mbps, um Distribuidor ligado a uma máquina com interface Gigabit Ethernet está limitado a atender 500 clientes. Na verdade, o limite é inferior pois a vazão de rede de 1 Gbps é nominal enquanto a taxa de 2 Mbps do vídeo precisa ser efetiva para não comprometer a exibição. A forma mais indicada de se contornar este problema é instalar um outro Distribuidor.

3.5 Cenários insatisfatórios para as versões isoladas

Para demonstrar a motivação para a integração das versões LAN e WAN, nesta Seção serão descritos alguns cenários em que a utilização isolada de uma das versões se mostra inadequada. Na verdade, para simplificar a discussão, será apresentado **um único** cenário base e algumas variantes neste cenário.

A Figura 3.1 mostra uma situação que pode ocorrer em uma instalação do GloVE-WAN em uma rede pública. Um condomínio de apartamentos contrata um serviço de rede em banda larga e coloca seus condôminos interligados através de uma rede local utilizando tecnologia *Fast Ethernet* (ou seja, à 100 Mbps) com um *switch*.

O serviço de VoD contratado e disponível pela operadora de telefonia prevê um Distribuidor instalado próximo da ligação entre a rede da operadora e a chegada ao condomínio.

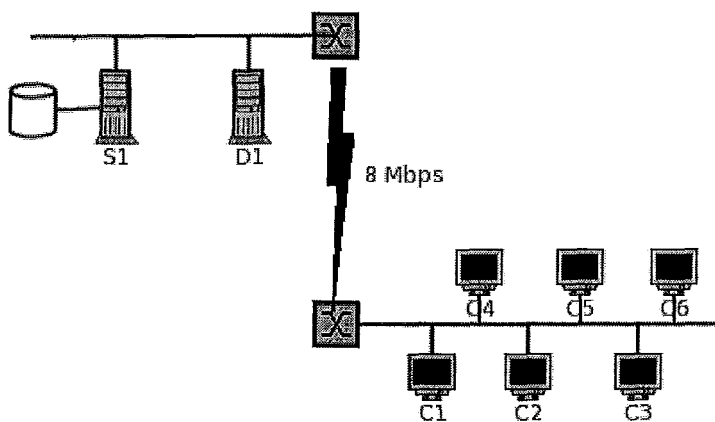


Figura 3.1: Cenário de uso WAN e LAN - O distribuidor é representado por D1, o Servidor por S1 e os clientes por C1 a C 6. Os clientes estão ligados por uma rede local. A ligação entre a operadora e a rede do local é de 8 Mbps

A vazão de rede contratada para o *link* até o condomínio é de 8 Mbps.

Com a divulgação de um novo vídeo disponível na rede da operadora de telefonia, os clientes do condomínio solicitam a transmissão do conteúdo. A utilização do *link* WAN consegue atender, considerando que a banda oferecida é efetiva e não nominal, um máximo de 4 clientes à taxa de transmissão de 2 Mbps. E não haverá mais vazão de rede disponível para qualquer outro serviço no condomínio.

É claro que, com vídeos distintos, este cenário não teria muitas opções de solução. Vídeos distintos exigem fluxos distintos. Possivelmente a mesma quantidade de fluxos será iniciada pelo Distribuidor no Servidor. Contudo, a premissa do projeto GloVE (e deste cenário em particular) é a demanda por um conteúdo comum por parte dos usuários, em um tempo próximo. Neste cenário, ainda que o Distribuidor consiga reduzir a quantidade de fluxos oriundos do Servidor, o consumo de vazão de rede entre o Distribuidor e os clientes segue um padrão de Servidor convencional, com um crescimento linear.

A utilização da versão *peer-to-peer* (ou LAN) isoladamente não é adequada, pelos motivos que levaram à criação da versão WAN. Utilizar o GloVE-LAN implicaria em se permitir que os clientes C1 a C2 recebessem vídeo do Servidor S1. Esta configuração pode degradar o desempenho de S1 para os demais clientes. A alternativa de se colocar um outro Servidor dedicado aos clientes do condomínio também não é adequada, aumentando a complexidade de gerenciamento da solução. Instalar um servidor dentro do condomínio é uma opção ainda mais inadequada, pois coloca o conteúdo do serviço de VoD fora da rede da operadora, comprometendo a segurança do conteúdo e facilitando possíveis

quebras de direito de propriedade intelectual.

O Capítulo 4 apresenta a solução que integra as versões WAN e LAN, permitindo que as vantagens de cada versão sejam aproveitadas, quando possível, de forma automática.

Capítulo 4

Integrando as versões LAN+WAN - GloVE MIX

4.1 Introdução

O projeto GloVE tem como base oferecer uma solução escalável e custo-efetiva para um sistema de VoD. A introdução da CVC (*Cooperative Video Cache*) e da C-CVC (*Collapsed CVC*), em conjunto com as implementações do GloVE-LAN e GloVE-WAN, oferecem uma forma eficiente de se distribuir VoD sem a necessidade de máquinas especialmente construídas para isso. Utilizando-se máquinas COTS (*Commercial off-the-shelf*) consegue-se obter um desempenho satisfatório, mesmo com uma grande quantidade de clientes.

Dadas as vantagens oferecidas pelo projeto GloVE em relação a soluções convencionais de VoD, o caminho natural é integrar as versões do GloVE em uma única versão auto-gerenciável. Assim, em lugar de se escolher qual versão será utilizada em uma instalação, pode-se escolher (ou, se possível, deixar o sistema escolher) a melhor forma de entregar conteúdo de vídeo em um determinado trecho da rede.

Integrar as versões do GloVE, e apresentá-las em uma roupagem “profissional”, apresentou alguns desafios. Ao longo deste Capítulo serão descritos os caminhos escolhidos nesta integração.

4.2 Situação encontrada antes da integração

No início do projeto de integração haviam duas versões distintas do projeto GloVE. Não apenas em termos de conceito de distribuição de vídeo (CVC x C-CVC), mas efetivamente em árvores de código distintas. A forma do cliente operar era diferente, os componentes envolvidos na distribuição e gerenciamento de vídeo eram exclusivos de cada

versão. Apenas o Servidor de Vídeo poderia ser considerado comum, porém isso já era esperado uma vez que o Servidor faz parte da distribuição de VoD mas também pode ser considerado um componente **externo** à solução GloVE.

Outra característica do projeto antes da integração era a possibilidade de se executar o cliente GloVE em algumas plataformas distintas. Havia algum código comum às versões Windows+MFC e Linux+Qt mas esta abordagem não era sistemática. Apesar de alguns componentes (como o Distribuidor) terem alguma previsão de execução em várias plataformas (Windows e Linux, por exemplo), apenas a versão Linux era efetiva.

O GloVE necessita de código próprio executando na máquina do cliente. Um dos motivos é a utilização de um protocolo proprietário para distribuição do vídeo. Contudo, especialmente para a versão *peer-to-peer* (ou LAN) há um motivo mais nobre. O GloVE precisa ter gerência sobre o *buffer* no cliente para que possa utilizá-lo no provimento para outros clientes. Na versão WAN esta necessidade não é tão forte e um *player* de vídeo de terceiros (Windows Media Player, MPlayer, Xine, VideoLAN, etc.) poderia servir. De qualquer forma, o cliente GloVE, independente de a qual versão estava associado, utilizava um cliente dedicado.

Por ter foco na distribuição do conteúdo de VoD e não especificamente na decodificação do vídeo, os componentes cliente do GloVE não efetuam a decodificação de vídeo em si. Para este fim, utiliza-se um decodificador externo. Nas versões encontradas utilizava-se o MPlayer [15], mas este não é um requisito forte. Em princípio, qualquer decodificador de vídeo pode ser utilizado.

A implementação do Gerente do GloVE-LAN, responsável por gerenciar o provimento entre clientes e a posição do *buffer* em cada cliente, utilizava estruturas dedicadas em memória para este fim. Ainda que a implementação fosse eficiente, a utilização da estrutura em memória leva a algumas considerações sobre escalabilidade e tolerância a falhas. Quanto maior o número de clientes, maior a quantidade de memória utilizada. E, em caso de queda no Gerente, todas as informações da rede GloVE estariam perdidas.

A conectividade entre os componentes era definida em arquivos de configuração estáticos, dificultando a gerência dos componentes. Cada Distribuidor estava associado a um único servidor e alterar esta associação implicava em se alterar arquivos de configuração na máquina do Distribuidor.

Nas seções seguintes serão descritas as decisões de projeto adotadas para a Integração das versões GloVE.

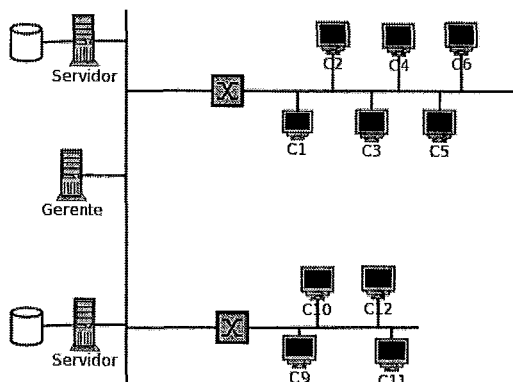


Figura 4.1: Estrutura geral do GloVE-LAN

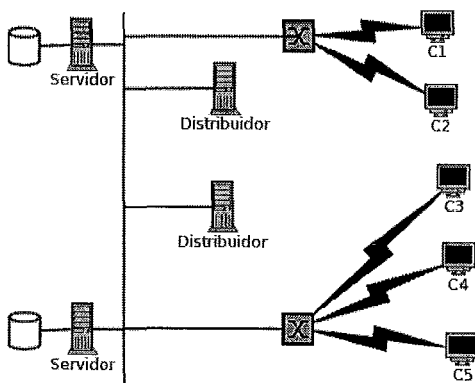


Figura 4.2: Estrutura geral do GloVE-WAN

4.3 Estrutura do GloVE

As Figura 4.1 e 4.2 mostram a estrutura geral do GloVE-LAN e GloVE-WAN respectivamente. A grosso modo, a estrutura das duas versões é completamente diferente.

A versão LAN utiliza um Gerente (CVCM) para o *buffer* dos clientes, mas os próprios clientes precisam estar configurados para identificar o Servidor de vídeo caso o Gerente indique que não há outro cliente para provê-lo. Os clientes podem ser providos diretamente pelo Servidor, utilizando-se o modelo *pull* de atendimento. Assim sendo, a vazão de rede não é utilizada da forma mais eficiente, arcando-se com o RTT (*round-trip time*) da rede a cada solicitação de bloco de vídeo.

O provimento de vídeo de um cliente para outro é governado pela taxa de consumo de vídeo no cliente. A cada bloco (ou conjunto de blocos) consumido pelo *player* externo, um conjunto equivalente é enviado para o outro cliente que está sendo provido. Esta abordagem simplifica a forma de gerenciar o envio do vídeo mas tem uma limitação.

Assim que o usuário pausa a exibição do vídeo, o outro cliente pára de receber vídeo. Isso ocorre mesmo que haja vídeo em *buffer* suficiente para o provimento.

A versão WAN não tem um Gerente de *buffer*, uma vez que o único *buffer* gerenciado é o próprio *buffer* do Distribuidor. Os clientes conectam-se ao Distribuidor, através de configuração específica na máquina do cliente. O Distribuidor tem configurado qual o Servidor a que deve solicitar vídeo, utilizando o modelo *pull*. Como o Distribuidor e o Servidor tendem a estar em uma mesma sub-rede ou, ao menos, estão localizados próximos ao *backbone* da rede, esta aparente ineficiência não é tão grave. Esta escolha é a mais indicada para que possamos ter independência do Servidor e permitir que o Servidor tenha uma implementação mais simples.

O GloVE-WAN possuía alguns componente acessórios para facilitar seu gerenciamento. Um destes componentes, o SGG (Sistema de Gerenciamento GloVE) — de autoria de Lauro Whately e Leonardo Pinho, com participação de Bernardo Costa — possui algumas funções de Autenticação e Autorização. Este componente permitia que usuários fossem cadastrados no sistema, tendo indicada a quais vídeos estavam autorizados a assistir. O SGG também possuía funções de tarifação, indicando quais vídeos haviam sido assistidos e quais opções de interatividade foram utilizadas pelo usuário (salto, pausa, etc.). Este componente (SGG) foi concebido como um adendo ao GloVE-WAN, para facilitar o seu gerenciamento e implantação em cenários reais.

4.3.1 Estrutura integrada

As alterações na estrutura do projeto GloVE foram adotadas buscando-se utilizar o melhor de cada versão para agregar valor ao projeto como um todo. A Figura 4.3 mostra a estrutura geral do GloVE-MIX, que integra as versões GloVE LAN e GloVE WAN.

Em linhas gerais, o Supervisor do GloVE-MIX substitui o papel do Gerente no GloVE-LAN. Contudo, além das funções do Gerente, ele agrega funções de Autenticação e Autorização projetadas para o SGG. De fato, com a utilização do GloVE-MIX no projeto Travis-QoS [16], o Supervisor passou a ter a capacidade de participar (embriõnariamente) de um processo de *single sign-on*. Integrando-se à estrutura de autenticação do AirStrike [17], o Supervisor permite que um usuário autenticado em uma rede segura possa utilizar o GloVE (desde que esteja autorizado para isso, é claro) sem a necessidade de nova autenticação. Maiores informações sobre as modificações no Supervisor podem ser encontradas na Seção 4.4.

O Supervisor passa a gerenciar todo o **grafo** de distribuição e não somente o *buffer*

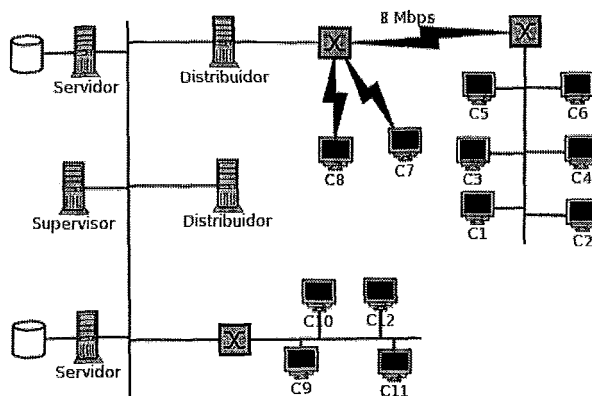


Figura 4.3: Estrutura geral do GloVE-MIX

nos clientes. A associação entre Distribuidores e Servidores também é gerenciada pelo Supervisor. O Distribuidor foi modificado para se conectar ao Supervisor e solicitar um Servidor. As modificações no Distribuidor são detalhadas na Seção 4.5.

Também o Servidor — ainda que seja um componente que possa ser considerado “externo” ao GloVE — foi modificado para indicar ao Supervisor quais vídeos está provendo. Assim, o Supervisor também tem conhecimento dos Servidores disponíveis no sistema. As alterações no Servidor estão detalhadas na Seção 4.6.

O cliente foi o componente com as maiores modificações, mesmo se considerarmos as alterações no Supervisor. As modificações no Cliente são detalhadas na Seção 4.7.

4.4 Supervisor

Esta Seção descreve as modificações efetuadas no Supervisor da versão integrada.

4.4.1 Atendimento das requisições

As alterações no Supervisor iniciaram na estrutura de atendimento das requisições. Enquanto o Gerente utilizava um protocolo baseado em UDP para comunicação com os clientes, o Supervisor (assim como o SGG) utiliza TCP como transporte para seu protocolo entre os demais componentes do sistema. Esta decisão tem a desvantagem de manter uma conexão TCP com cada componente ligado ao Supervisor. Porém, esta abordagem foi adotada visando a simplificação da implementação do Supervisor. Utilizando-se conexões TCP, não é necessário tratar quebras e/ou entrega fora de ordem nos pacotes do Supervisor. Também é mais simples tratar quando um componente falha, pois a conexão

TCP é interrompida, indicando ao Supervisor que o componente não está mais disponível na Rede GloVE.

Como a interrupção da conexão é efetuada por *timeout*, entre os demais componentes e o Supervisor é enviada, periodicamente, uma mensagem de *heartbeat*. Nesta mensagem de atualização, os componentes informam qual o estado de seu *buffer* local. Na implementação atual, apenas os clientes informam este estado completo ao Supervisor. O Distribuidor não informa o estado de seu *buffer*. Da forma que o Servidor é utilizado na Rede GloVE, não há o que ser informado.

O Supervisor tem um único processo com uma única *thread* tratando as requisições que são enviadas pelos demais componentes. O Supervisor monitora as conexões ativas e cada requisição enviada é tratada e respondida.

Esta forma de atendimento, embora simplifique a implementação, deve ser modificada no futuro. Atender em uma única *thread* cria um gargalo no atendimento das requisições. Se muitas requisições chegarem elas serão enfileiradas e tratadas uma a uma. A opção pelo atendimento em um único processo/*thread* deve-se a necessidade simplificar a forma de controlar a concorrência nas estruturas do Supervisor.

Conforme os componentes se conectam ao Supervisor e solicitam provedores e atualizam o Supervisor com o estado de seu *buffer*, o Supervisor precisa manter o estado do grafo de distribuição coerente. Para a utilização de uma estrutura mais adequada de atendimento de requisições, recomenda-se a execução do Supervisor em um Servidor de Aplicações. Este trabalho envolverá modificações no Supervisor e na forma de comunicação com os demais componentes.

Visando esta migração futura, e facilitar a implementação de diferentes políticas de escolha de provedor de vídeo, a estrutura do Supervisor passou a ser armazenada de forma diferente, conforme descrito na Subseção 4.4.2.

4.4.2 Armazenamento das informações

O Supervisor precisa armazenar o Grafo de Distribuição de Vídeo, também chamado de Estado da Rede GloVE. Este grafo contém os componentes de distribuição de vídeo e suas relações de provimento, modelado da seguinte forma:

Seja o grafo $G = (V_G, E_G)$ onde V_G , **vértices** do grafo, é o conjunto de componentes de transmissão (distribuidores, servidores, clientes) e E_G , **arestas** do grafo, é o conjunto das relações de provimento existentes entre estes componentes.

Em lugar de se armazenar os vértices e arestas diretamente em memória, utiliza-se um

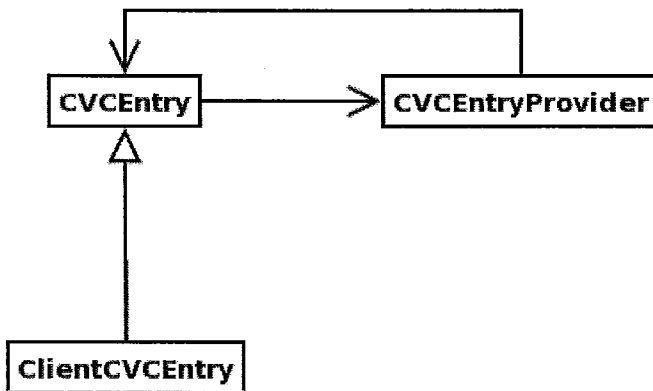


Figura 4.4: Diagrama simplificado do armazenamento do grafo no Supervisor

SGBD (Sistema de Gerenciamento de Banco de Dados) para tal fim. Na implementação da integração efetuada nesta dissertação, as arestas e vértices estão colocadas em tabelas no MySQL [18]. De modo simplificado, a estrutura é a representada na Figura 4.4. Cada vértice do grafo (servidor, distribuidor ou cliente) é uma linha na tabela `CVCEntry`. Como informações detalhadas são armazenadas, na implementação atual, somente para os clientes, há também uma linha extra por cliente na tabela `CVCEntryClient`. A relação de provimento é armazenada na tabela `CVCEntryProvider`.

Cada vídeo possui um grafo de distribuição distinto. Assim, a cada vídeo que um servidor oferece é criada uma linha em `CVCEntry`. Desta forma, cada linha em `CVCEntry` é, na verdade, a indicação que um componente do GloVE está associado a um vídeo. Na implementação atual também há tabelas acessórias para os vídeos cadastrados, os usuários autorizados e suas senhas, quais dos vídeos cadastrados estão disponíveis para provimento, etc.

Armazenar o Grafo de Distribuição de Vídeo em um SGBD tem uma vantagem em particular. Pode-se implementar políticas de escolha de provedor de forma mais simples, através de consultas SQL. Adicionalmente, a Figura 4.5 apresenta a estrutura de classes (o Supervisor foi implementado em C++) que o Supervisor possui relacionada às políticas de escolha de provedor.

4.5 Distribuidor

O Distribuidor é o principal componente de distribuição de vídeo na versão integrada. Enquanto na versão GloVE-LAN (ou *peer-to-peer*) o cliente recebe vídeo de um Servidor,

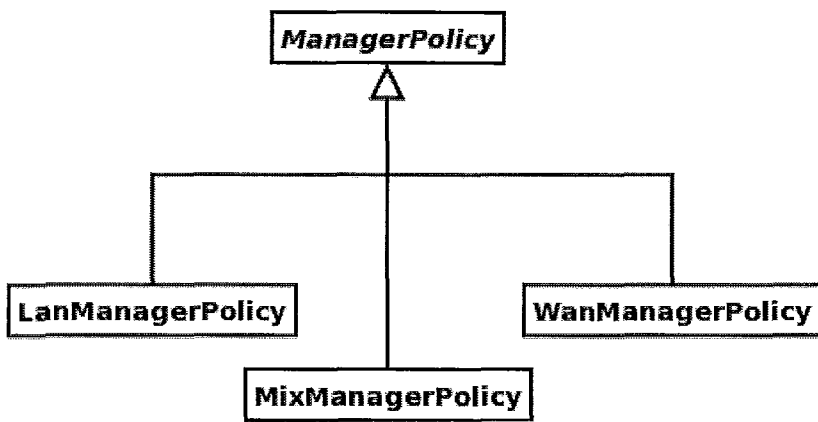


Figura 4.5: Estrutura de classes para políticas de provimento

na versão integrada é sempre um Distribuidor (ou outro cliente) que provê vídeo. Esta aparente simplificação faz com que a quantidade de fluxos oriundos do Servidor possa ser melhor gerenciada, uma vez que o Distribuidor (por usar a técnica C-CVC) permite o reuso dos fluxos de forma mais eficiente.

Assim, como o cliente não tem mais a configuração estática da localização de qualquer componente que não o Supervisor (vide Seção 4.7), o Distribuidor precisa se conectar também ao Supervisor, informando os vídeos que está provendo. O Distribuidor não foi alterado para ter uma lista dinâmica de vídeos, configurando-se na carga inicial com os vídeos que o Supervisor indica que estão disponíveis.

Outra modificação foi obter qual(is) o(s) Servidor(es) que atenderá(ão) o Distribuidor. Para cada vídeo que o Distribuidor aloca *buffer*, é solicitado um endereço de Servidor para provimento. A política de alocação de Servidores a Distribuidores é, então, dada pelo Supervisor.

Apesar de não associada diretamente com a Integração, o Distribuidor foi ligeiramente modificado para poder executar em diferentes plataformas, utilizando a infra-estrutura montada para a construção do cliente GloVE (Seção 4.7).

4.6 Servidor

Para o projeto GloVE, o Servidor é um componente que, apesar de essencial, é tratado como um componente externo. Não há técnicas ou estratégias específicas para o Servidor. Assume-se que sua implementação pode ser a mais simples possível, carregando blocos de disco quando solicitado (modelo *pull*) e enviando-os para o componente solicitante.

Na versão integrada, apenas o Distribuidor utiliza os serviços do Servidor. Entretanto, esta não é uma política rígida. O cliente está preparado para receber vídeo utilizando o mesmo protocolo (proprietário) que o Distribuidor utiliza para se comunicar com o Servidor. Contudo, dadas as necessidades de recursos de um Servidor, especialmente em termos de vazão de disco, optou-se por utilizar sempre um Distribuidor como provedor dos clientes, caso outro cliente não possa prover.

A modificação que foi efetuada no Servidor — e que será uma adaptação necessária ao se utilizar um Servidor de terceiros — foi a integração com o Supervisor. O Servidor informa ao Supervisor quais vídeos é capaz de prover. Desta maneira, o Supervisor pode adicionar os Servidores ao Grafo de Distribuição de Conteúdo.

4.7 Cliente GloVE

O cliente GloVE tem um papel importante na utilização da versão GloVE LAN (ou *peer-to-peer*) pois o *buffer* no cliente pode ser utilizado para provimento a outro cliente. Porém outra característica foi fundamental para investir-se na reestruturação do cliente. Ele é o componente que está em contato direto com o usuário do serviço. Em uma solução utilizada em larga escala, é o componente que mais tende a sofrer modificações e personalizações de acordo com a preparação que será efetuada na solução para implantação em um sistema real.

Assim, o cliente GloVE foi completamente reestruturado para comportar as versões GloVE-LAN e GloVE-WAN. A Figura 4.6 mostra a arquitetura do cliente GloVE.

Duas diretrizes guiaram o projeto da nova arquitetura. Em primeiro lugar, deve ser possível **variar a interface com o usuário**. Uma vez que o cliente GloVE é o componente de contato com o usuário, torna-se o principal candidato a sofrer modificações para as possíveis implantações em um ambiente real. Pode-se ter variações na apresentação ao usuário, na plataforma, na biblioteca gráfica, etc. A segunda diretriz é a da **portabilidade**, permitindo que o usuário utilize a plataforma que mais atenda às suas necessidades.

Contudo, o cliente tem um papel importante na Rede GloVE, com o *buffer* de vídeo servindo como possível fonte de provimento para outro cliente. Assim, é necessário que o componente cliente tenha código originário do GloVE e que este código tenha um comportamento uniforme em todas as implementações.

Deste modo, há quatro grandes divisões no componente cliente do GloVE.

1. **Interface com o usuário** - Esta camada é responsável pela apresentação do GloVE

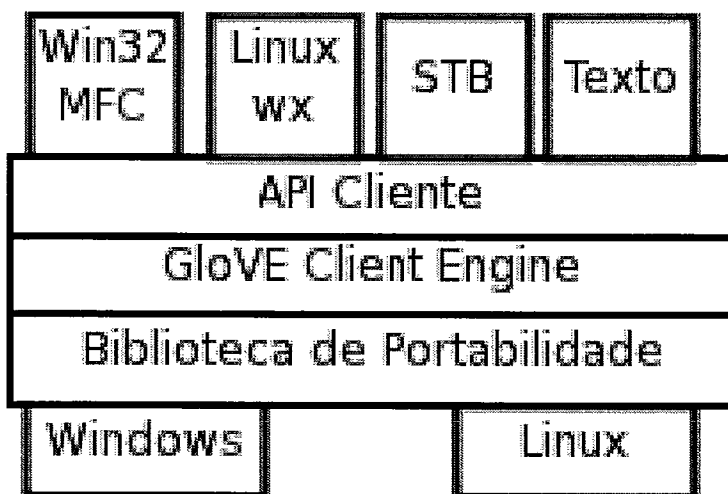


Figura 4.6: **Arquitetura do Cliente GloVE** - A intenção é permitir que haja opções de interface com o usuário - GUI (na figura, Win32/MFC, Linux/wxWidgets, STB-Set-top box e uma interface de Texto para testes). Também deve ser possível variar as plataformas em que o cliente está disponível (na figura, Windows e Linux, já implementadas, mas permitindo a implementação de outras). A API Cliente permite a variação de GUI e a Biblioteca de portabilidade permite a variação das plataformas. Assim, o núcleo do cliente GloVE (GloVE Client Engine) não é afetado.

ao usuário, permitindo sua interação com o sistema de VoD. Em princípio, esta camada **não** pertence ao código GloVE. Ou seja, a implementação desta porção do código cliente não exige conhecimentos do funcionamento do GloVE.

2. **API Cliente** - Oferece chamadas que implementam funções de interação com o vídeo, como *play*, *pause*, *next*, *previous*. Estas chamadas são mapeadas em operações do núcleo GloVE (*GloVE Client Engine*).
3. **GloVE Client Engine** - Inteligência do componente cliente, responsável pela gerência do *buffer*, interação com o Supervisor, recepção de vídeo, provimento para outro cliente, etc. Suas funções e arquitetura própria serão discutidas na Seção 4.7.1.
4. **Biblioteca de Portabilidade** - Para evitar o tratamento das idiossincrasias de cada plataforma de execução no núcleo do GloVE, implementou-se uma biblioteca de portabilidade. Assim, o que as funções do sistema que as bibliotecas padrão da linguagem de implementação (no caso, C/C++) não oferecem, esta camada se encarrega de oferecer. Isola-se, por exemplo, aspectos de gerenciamento de *threads*. Esta camada também foi utilizada nas alterações no Distribuidor, permitindo a sua

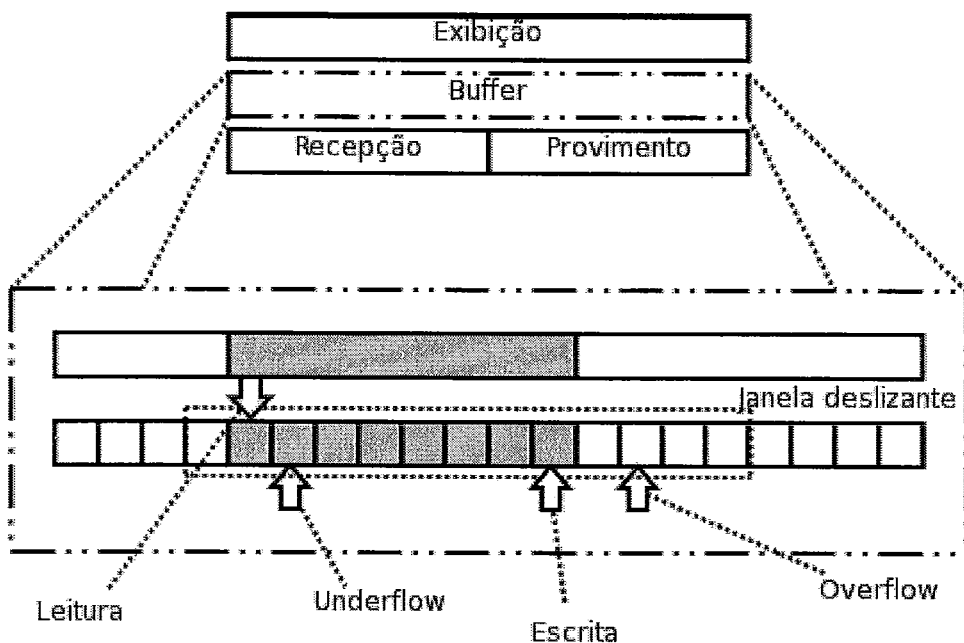


Figura 4.7: *GloVE Client Engine* - O controle de exibição gerencia o repasse de vídeo para o *player* externo, na implementação corrente utiliza-se o MPlayer. O controle de Recepção gerencia a interação com o provedor (Distribuidor ou outro cliente). O controle de Provimento envia o vídeo para outro cliente (pode-se ter mais de um cliente recebendo vídeo). O *buffer* contém o vídeo e tem sua estrutura expandida na figura. A estrutura do *buffer* é detalhada no texto.

execução tanto em Windows quanto em Linux. Outras opções poderiam ser implementadas.

4.7.1 *GloVE Client Engine*

O gerenciamento do *buffer* no cliente permite que o GloVE-MIX (assim como o GloVE-LAN) utilize, quando aplicável, o modelo *peer-to-peer* de distribuição de vídeo. O núcleo do módulo cliente — *GloVE Client Engine* - GCE — implementa esta inteligência, através da Arquitetura apresentada na Figura 4.7.

Há duas *threads* ativas na GCE. Uma delas controla a exibição do vídeo, contendo o módulo “Exibição” apresentado na Figura 4.7. Esta *thread* é responsável por repassar vídeo para o *player* externo. Na implementação desta Dissertação utilizou-se o MPlayer. Vale lembrar que não é foco desta Dissertação (ou do projeto GloVE em geral) lidar com a decodificação do vídeo. A exibição está em uma *thread* separada para que o processamento da decodificação do vídeo possa se sobrepor à recepção do vídeo, especialmente

em máquinas multiprocessadas.

A segunda *thread* é responsável por toda a gerência restante do *buffer*, aí inclusa a recepção do vídeo, a interação com o Supervisor e o provimento à outros clientes. A divisão desta *thread* não se mostrou atrativa, uma vez que o percentual de preenchimento do *buffer* está intimamente ligado com a recepção do vídeo. E o provimento a outros clientes depende do percentual de preenchimento ou, em outras palavras, da quantidade de vídeo disponível no *buffer* do cliente.

O *buffer* é implementado como um vetor circular (*ring buffer*), com tamanho fixo em memória. Contudo, nesta dissertação, algumas modificações periféricas foram introduzidas.

4.7.1.1 Visão de janela deslizante

O *buffer* é utilizado como se todo o vídeo estivesse em memória. Assim, se o vídeo tem 2 GB, os índices do *buffer* variam de 0 a 2097152 (em blocos de 1 KB). Contudo, apenas alguns destes índices estarão realmente em memória. Faz-se um mapeamento entre os índices do arquivo e os índices do *ring buffer* que armazena o vídeo. Conceitualmente, é como se houvesse uma **janela deslizante** que permite enxergar apenas uma parte do vídeo. Na Figura 4.7, esta janela é representada por uma linha pontilhada ao redor de alguns blocos do *buffer*. O tamanho da janela é dado pela quantidade de memória que se quer efetivamente utilizar para armazenamento do vídeo no cliente.

Por exemplo, se temos uma janela de 16384 blocos, temos acesso a 16 MB de vídeo na memória do cliente. Se a posição inicial da janela for zero (0), os blocos de 0 a 16383 estão visíveis. Se a posição inicial for 350, os blocos de 350 a 16733 estão visíveis.

O bloco estar visível não significa que está disponível. Blocos disponíveis são blocos que já foram recebidos de um provedor. Na Figura 4.7, os blocos escurecidos representam os blocos disponíveis. Obviamente, um bloco só pode estar disponível se estiver também visível.

Conforme visto na Figura 4.7, quatro índices controlam o gerenciamento do *buffer*:

- **Leitura** - Indica o ponto em que a *thread* de exibição está consumindo o vídeo. Este é gerenciado pela *thread* de exibição mas é importante para o controle da *thread* de recepção/provimento. Quaisquer movimentos na janela do *buffer* devem manter o índice de Leitura visível (e disponível).
- **Underflow** - Indica a iminência de uma pausa na exibição, por falta de vídeo dis-

ponível. Se tolera-se que a exibição seja interrompida, este índice é igual ao índice de Exibição. Caso contrário, deve-se prever uma folga entre o índice de *underflow* e o índice de leitura. Esta folga deve ser suficiente para a **migração** de um provedor para outro, evitando a interrupção da exibição.

- **Escrita** - Indica o próximo bloco que deverá ser recebido. No modelo *pull*, é utilizado para solicitar o bloco seguinte ao Servidor, no caso deste estar provendo o cliente¹. No modelo *push*, este índice é importante para se controlar blocos perdidos na recepção, ou blocos recebidos fora de ordem. Como o protocolo de vídeo utilizado nesta dissertação é baseado em UDP, estes eventos podem ocorrer.
- **Overflow** - Indica a iminência de ser necessário descartar vídeo recebido, por falta de espaço no *buffer* para armazená-lo.

A recepção do vídeo guia o movimento da janela deslizante. Conforme o índice de Escrita progride, a quantidade de blocos livres no final da janela diminui, aproximando os índices de Escrita e Overflow. Ao atingir o índice de Overflow, a janela é movida para direita.

Porém, o índice de Leitura precisa sempre estar visível e disponível. Se ao movermos a janela o índice de Leitura não estiver visível, o movimento da janela não é efetuado. Como o índice de Escrita atingiu o índice de *Overflow*, uma solicitação de pausa no provimento é efetuada, evitando que blocos enviados pelo provedor sejam descartados.

O índice de *Overflow* pode ser atingido se o usuário pausa o vídeo, por exemplo. Neste caso, a recepção do vídeo não é interrompida. Enquanto houver espaço em *buffer* para armazenar o vídeo, a recepção prossegue aumentando a margem de folga para exibição do vídeo e provimento para outros clientes.

Outra situação em que o índice de *Overflow* pode ser atingido é quando a taxa de recepção é maior que a taxa de exibição. Neste caso, a distância entre o índice de Leitura e de Escrita aumenta, aproximando o limite inferior da janela do índice de Leitura. Utiliza-se a taxa um pouco maior para que se possa aumentar a folga para tratamento de problemas, como uma possível troca (ou migração) de provedor. Esta forma de aumentar o preenchimento do *buffer* evita que seja necessário aumentar o tempo de espera inicial, para efeitos de *prefetch*, antes que o vídeo possa ser assistido.

¹A política empregada nesta dissertação não prevê o Servidor como provedor direto de um cliente. Contudo, esta opção pode ser facilmente introduzida.

O índice de *Underflow* atua na hipótese de uma degradação na recepção do vídeo. Isso pode ocorrer por congestionamento na rede, queda do provedor, interrupção do provimento por outro cliente (por salto no vídeo ou simplesmente fechamento pelo usuário), entre outros. O índice de *Underflow* garante uma quantidade mínima de vídeo disponível para uma eventual troca de provedor, nos casos em que não se deseja interrupção na exibição. Nos casos em que pausas na exibição são toleradas, o índice de *Underflow* é colocado em um valor mínimo para tratamento do *jitter* na rede. Após esse preenchimento mínimo, o índice pode ser colocado como sendo sempre igual ao de Leitura. Assim, só será efetuada troca de provedor se o conteúdo do *buffer* se esgotar.

A distância mínima entre os índices de Leitura e *Underflow*, quando se quer evitar pausas na exibição, é igual ao tempo necessário — idealmente, ao tempo **máximo** — para se efetuar uma troca (ou migração) de provedor. Quando o provedor é um Servidor, este tempo é dado pela latência da rede (em outras palavras, pelo *RTT* - *round-trip time*), considerando-se que o tempo de início de provimento no Servidor é desprezível. Para um Cliente provendo, o início também é dado pelo *RTT*, mas entre os clientes. Já o tempo de início de provimento pelo Distribuidor é, na média, metade do tempo de *slot* configurado no Distribuidor.

Assim, a distância entre o índice de Leitura e o de *Underflow* é dada, em linhas gerais, por:

$$distancia_{underflow} = (Escalonamento_{supervisor} \quad (4.1)$$

$$+ RTT_{cliente-supervisor} \quad (4.2)$$

$$+ RTT_{cliente-provedor} \quad (4.3)$$

$$+ Escalonamento_{provedor}) \quad (4.4)$$

$$\times taxa_{bps} \times \div 8 \div 1024 \quad (4.5)$$

Nesta dissertação utilizou-se $Escalonamento_{provedor} = 2 \times tempo_{slot}$, $RTT_{cliente-supervisor} = 0$, $RTT_{cliente-provedor} = 0$, $Escalonamento_{supervisor} = 0$ no cálculo da distância de *Underflow*. Para a distância de *Overflow* utilizou-se $tempo_{slot} \div 2$. Assim, permite-se que o Distribuidor pause o envio do vídeo em caso de *overflow*, evitando descarte de vídeo e consumo de vazão de rede desnecessariamente.

4.8 Adaptações para rede sem fio

A utilização de redes sem fio tem crescido, especialmente no último trecho da rede, exatamente onde são alocadas as máquinas dos usuários. Durante o Projeto Travis-QoS [16], algumas adaptações foram efetuadas, especialmente no Cliente GloVE.

A principal delas foi o tratamento de blocos perdidos. Na versão LAN, as perdas de blocos oriundos do Servidor eram tratadas solicitando-se novamente o bloco. No caso de perdas em bloco enviados por outro cliente, isso era considerado como falha no provimento, sendo solicitado outro provedor. A grosso modo, a versão LAN **não** considerava a situação de perda de blocos, uma opção factível em uma rede local, onde a taxa de perdas é baixa.

A versão WAN, por prever utilização em uma rede pública, considerava que um bloco poderia ser perdido. Dentro de alguns limites impostos pelo Distribuidor, os blocos perdidos eram solicitados novamente pelo Cliente, permitindo que a exibição prosseguisse sem maiores transtornos.

Ao se utilizar o GloVE com os clientes em uma rede sem fio, a taxa de perda de blocos é maior que a esperada em uma rede cabeada. Com o aumento no número de clientes de vídeo atendidos pelo mesmo Ponto de Acesso (*AP - Access Point*), a utilização de vazão de rede cresce, aumentando a quantidade de pacotes trafegando na rede sem fio. Consequentemente, a quantidade de colisões e perdas também cresce. Com a solicitação de reposições para blocos perdidos, o cenário de sobrecarga piora, aumentando a quantidade de blocos perdidos. Assim, o que foi previsto como uma solução passa a fazer parte do problema.

Para minimizar o impacto dos blocos perdidos, utilizou-se uma tolerância presente na exibição de vídeo. Em um vídeo com taxa de exibição de 2 Mbps, um segundo (1s) de vídeo equivale a aproximadamente 244 blocos de 1 KB – os blocos estão definidos neste tamanho para maximizar o uso da vazão de rede disponível, aproximando o tamanho dos blocos do MTU (*Maximum Transmit Unit*) da rede Ethernet. Saltar a exibição em 1s é incômodo para o usuário porém interromper a exibição causa desconforto maior. E nas situações em que a perda é menor, o formato de vídeo pode suportar o efeito de “quadricular” a exibição, ou seja, a percepção que o usuário tem da perda de alguns blocos é uma diferença muito pequena em alguns trechos da imagem, com o aparecimento de alguns “quadrículos” de cor diferente da esperada.

Assim, para evitar-se o agravamento da perda de blocos com a solicitação desnecessá-

ria dos blocos perdidos, o Cliente GloVE tolera uma perda (configurável) de até 2s (arbitrários) de vídeo. Se após o trecho perdido há vídeo que permita uma exibição contínua, o Cliente GloVE salta o trecho em questão. Quando o formato do vídeo permite “quadricular” a exibição, este pequeno salto pode não incomodar o usuário. Usualmente o formato do vídeo possui alguns quadros completos e vários quadros que aplicam diferenças. Assim, ao se perder blocos pertencentes a quadros de diferença, apenas alguns trechos da imagem serão afetados. Se pertencer a um quadro completo, a imagem pode ficar irreconhecível. De qualquer forma, este efeito será corrigido quando o próximo quadro completo for recebido.

Com esta modificação, a versão integrada consegue tratar perdas também quando clientes provêm. Não se implementou a solicitação de blocos perdidos quando um outro cliente é o provedor, evitando-se aumentar quantidade de processamento que o cliente provedor precisaria tratar para atender a outros clientes. Infelizmente, quando a quantidade de perdas é alta, este tratamento degrada a exibição do vídeo para o usuário. Em trabalho futuro, deve-se tratar a perda de blocos também na transmissão entre clientes.

O salto de blocos perdidos, quando possível, melhora a utilização do GloVE-MIX em redes sem fio e, como efeito colateral, melhora sua escalabilidade também em redes cabeadas. Afinal, nos dois casos, quando o problema da perda é localizado e passageiro, um pequeno impacto na exibição, que tem probabilidade alta de ser imperceptível ou desprezível para o usuário, é um preço menor a se pagar que a possibilidade de agravamento do congestionamento na rede, com o acréscimo da solicitação de blocos perdidos.

4.9 Conclusões

As versões do GloVE (LAN e WAN) oferecem soluções escaláveis atrativas para ambientes de redes locais com vazão simétrica e redes públicas com vazão assimétrica. Contudo, em algumas cenários, a implantação mais adequada é uma combinação das duas.

O trabalho de integração buscou aumentar a escalabilidade nas situações em que se pode utilizar da escalabilidade garantida pelo Distribuidor e agregar recursos dos clientes, quando vários usuários em uma rede local solicitam o mesmo vídeo.

A integração das versões permite futuras expansões nas formas de distribuição do vídeo, visto que o Supervisor passa a ter conhecimento do grafo de distribuição completo. Como as alocações de provimento passam a ser dinâmicas, políticas diferentes de aloca-

ção de provedor podem ser elaboradas, testadas e implantadas de forma mais simples.

Capítulo 5

Análise Experimental

5.1 Objetivo dos experimentos

Utilizar as técnicas CVC (*Cooperative Video Cache*) e C-CVC (*Collapsed CVC*) melhora a escalabilidade da solução de VoD, conforme demonstrado por ISHIKAWA e AMORIM [11, 7, 3], PINHO [12] e BRAGATO [4]. Nesta dissertação, as duas técnicas foram integradas, utilizando-se como base as implementações dos protótipos de CVC (GloVE-LAN [11, 12]) e C-CVC (GloVE-WAN [7, 4]). O objetivo da integração é aumentar a escalabilidade da C-CVC através da utilização da CVC nos trechos da rede em que a vazão de rede for simétrica (rede local, por exemplo) e houver concentração de usuários assistindo ao mesmo vídeo.

Assim, o objetivo dos experimentos é comparar o desempenho da versão GloVE-WAN com a versão integrada. A Seção 5.3, lista os elementos avaliados nesta dissertação. Idealmente, a versão integrada deve ser tão eficiente quanto a versão GloVE-WAN isoladamente. É mais eficiente nos cenários favoráveis à integração. Além do desempenho comparativo das versões do GloVE, a sobrecarga introduzida pelos controles necessários à versão integrada também será analisada.

O restante do Capítulo está estruturado conforme descrito a seguir. Na Seção 5.2 é descrito o ambiente experimental. Na Seção 5.3 são listadas as métricas de desempenho utilizadas. Na Seção 5.5 é avaliada a carga de trabalho que será medida. Na Seção 5.6 os resultados obtidos são listados e analisados. Na Seção 5.7 é efetuada uma discussão comparativa sobre os resultados obtidos e os esperados.

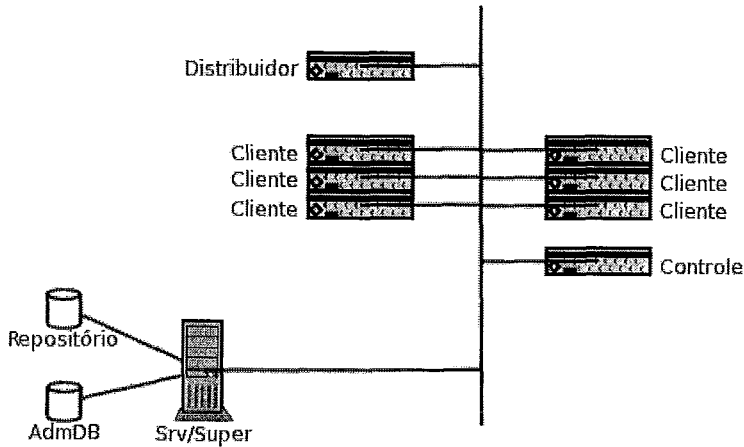


Figura 5.1: **Estrutura dos Experimentos** - A estrutura aqui apresentada é simplificada. As máquinas do cluster estão divididas em quatro (4) *cabinets*, todos interligados por *switch* gigabit Ethernet. As máquinas no cluster também estão interligadas por um *switch* gigabit Ethernet. **AdmDB** - Banco de dados de administração do grafo de distribuição. **Repositório** - Dados dos vídeos oferecidos. **Srv/Super** - Servidor de Vídeo + Supervisor GloVE. **Controle** - Máquina de controle do experimento.

5.2 Ambiente Experimental

Para avaliação dos resultados obtidos com as alterações efetuadas nesta dissertação, utilizou-se o ambiente computacional disponível no Laboratório de Computação Paralela (LCP) [19]. As configurações de ambiente de experimental são descritas a seguir. A Figura 5.1 mostra a estrutura das máquinas de forma simplificada.

5.2.1 Servidor e Supervisor

O Servidor e Supervisor estão configurados em uma mesma máquina, localizada nas instalações do LCP, a saber:

- Uma máquina Servidora/Supervisora com:
 - Intel Pentium D 2.8 GHz 2MB cache
 - 3 GB Memória
 - Disco Serial-ATA 160 GB
 - Linux kernel 2.6.9-42.0.8.EL i386
 - CentOS 4.1
 - MySQL Server 4.1

5.2.1.1 Clientes e Distribuidores

As máquinas clientes e distribuidoras foram alocadas no cluster do LCP. A alocação é aleatória, porém privilegiando o Distribuidor nas máquinas com mais recursos, e utiliza as seguintes máquinas.

- Cinco (5) máquinas
 - Intel(R) Pentium(R) 4 CPU 2.80GHz
 - 1 GB Memória
 - Disco IDE 40 GB
 - Linux kernel 2.6.9-34.ELsmp i386
 - Rocks 4.1

- Três (3) máquinas
 - Intel(R) Pentium(R) D CPU 3.00GHz
 - 2 GB Memória
 - Disco Serial-ATA 80 GB
 - Linux kernel 2.6.9-34.ELsmp i386
 - Rocks 4.1

5.3 Métricas de desempenho

A seguir, descreve-se as métricas utilizadas para avaliação dos resultados obtidos com a integração das técnicas de CVC e C-CVC..

5.3.1 Fluxos oriundos do Servidor

Como o Servidor de vídeo é um dos recursos mais caros na solução de VoD, a principal métrica é a quantidade de fluxos de vídeo oriundos do Servidor. Quanto menor for esta quantidade, mais clientes podem ser atendidos pela solução de VoD, sem a necessidade de se acrescentar mais recursos aos Servidores de Vídeo.

Esta mesma métrica foi utilizada pelos trabalhos anteriores, tanto na definição da CVC [11] ou C-CVC [7, 3], quanto nos protótipos que as implementaram, o GloVE-LAN [12] e o GloVE-WAN [4], respectivamente.

5.3.2 Fluxos oriundos dos Distribuidores

Com a integração das técnicas CVC e C-CVC adotada nesta dissertação, o Distribuidor assume o papel de “servidor de vídeo” para os clientes em uma rede LAN. Assim, a quantidade de fluxos oriundos dos Distribuidores também é avaliada, medindo-se o impacto da utilização da CVC na escalabilidade geral do sistema de VoD. Quando menos fluxos destinados a clientes forem originados de um Distribuidor, maior a quantidade de clientes que pode ser atendida pela solução de VoD.

5.3.3 Fluxos oriundos de outros Clientes

De forma complementar às métricas adotadas nas Seções 5.3.1 e 5.3.2, a quantidade de fluxos oriundos de outros clientes também é medida. De modo geral, busca-se (na solução integrada GloVE-MIX) maximizar a quantidade de fluxos oriundos de outros clientes, minimizando a quantidade de fluxos oriundos de distribuidores e servidores.

5.3.4 Custo de comunicação

De forma a medir o custo de comunicação introduzido pela comunicação entre os componentes e o Supervisor, é medida a quantidade de bytes extras introduzidos pelas mensagens de controle. Este (*overhead*) também pode ser considerado em quantidade de mensagens. Contudo, a dedução da quantidade de mensagens em relação à quantidade de bytes é relativamente direta, pois todas as mensagens são de 1KB.

5.3.5 Resumo das métricas

Aqui são resumidas as métricas que serão avaliadas. No lado esquerdo de cada métrica é colocada uma seta para cima (↑) indicando que a métrica deve ser maximizada. É colocada uma seta para baixo (↓) quando ela deve ser minimizada.

- (↓) Quanto menor a quantidade de fluxos oriundos de Servidor, melhor;
- (↓) Quanto menor a quantidade de fluxos oriundos de Distribuidor, melhor;
- (↑) Quanto maior a quantidade de fluxos oriundos de Cliente, melhor;

O Custo de comunicação não é avaliado de forma comparativa, somente como uma demonstração dos recursos que a solução demanda. Por não estar relacionado diretamente com as variações dos experimentos (conforme será detalhada mais adiante), o Custo de Comunicação é analisado à parte, na Seção 5.6.5.

5.4 Problemas encontrados durante os experimentos

A implementação utilizada para experimentação da integração é um protótipo com preocupações e recursos próximos de uma aplicação real de VoD. Isso implica suportar autenticação, ter funcionalidades (básicas) de proteção de direitos autorais, tratamento de blocos perdidos na rede, suporte à rede sem fio no cliente, suporte a execução em mais de uma plataforma, decodificador de vídeo (externo, MPlayer) completo, entre outros. Assim, os requisitos de memória e CPU necessários à sua execução são maiores. Alguns problemas foram encontrados durante a execução dos experimentos e são listados a seguir.

5.4.1 Quantidade de clientes por máquina

A quantidade de clientes por máquina é limitada, não somente pela quantidade de memória livre, mas também pela capacidade de processamento. Quando utilizados muitos clientes simultâneos em uma mesma máquina, utilizando-se o Kernel Linux 2.6, a aplicação apresentou degradação na execução após alguns (5 ou 7) minutos de execução. A taxa de recepção era reduzida, afetando também a taxa de envio. Desta forma, todos os clientes passavam a esgotar o seu *buffer* de vídeo, necessitando de troca de provedor.

Após alguns experimentos, chegou-se ao valor máximo de cinco (5) clientes por máquina para serem utilizados no Cluster do LCP. Experimentos com mais clientes foram executados para medir a degradação alcançada (maiores informações na Seção 5.6.1)

5.4.2 Estouro do *buffer*

Para reduzir o tempo inicial de *prefetch* necessário para se oferecer alguma margem de tolerância a falhas, optou-se por utilizar uma taxa de envio 10% superior à taxa de recepção. Assim, ao longo da exibição, o preenchimento do *buffer* aumenta, reduzindo a possibilidade de não haver vídeo disponível em *buffer* para tratar uma possível troca de provedor.

Porém, esta decisão causa um outro problema. Como o preenchimento do *buffer* é crescente mas o tamanho do *buffer* é constante, após algum tempo – que depende do tamanho do *buffer* e da taxa do vídeo – o *buffer* atinge 100% de preenchimento. Além deste ponto, os blocos recebidos devem ser descartados.

Para evitar que os blocos sejam descartados, o provimento é interrompido ou, em outras palavras, sofre uma pausa. Durante esta pausa no provimento, o usuário **continua**

a assistir ao vídeo – e um eventual outro cliente sendo provido continua a ser provido. O provimento precisa ser retomado quando o preenchimento do *buffer* cai, permitindo que mais blocos sejam recebidos.

A política de retomada do provimento varia de acordo com o tipo de provedor:

- **Distribuidor** - Quando o cliente recebe vídeo de um Distribuidor, a pausa é mantida pelo maior tempo possível. Ou seja, retoma-se a pausa quando o preenchimento do *buffer* atingir o ponto mínimo em que a exibição não será comprometida até que a pausa seja suspensa. O Distribuidor (na implementação utilizada) demora até $3 \times tempo_{slot}$ para retomar o provimento. Assim, quando este preenchimento mínimo é atingido, é solicitada a retomada do provimento ao Distribuidor;
- **Cliente** - Quando o cliente recebe vídeo de outro cliente, a pausa é mantida pelo **menor** tempo possível. Ou seja, suspende-se o provimento por algum tempo mas retoma-se logo que o *buffer* tenha espaço para receber mais vídeo. Utilizou-se o mesmo tempo que o definido para o *prefetch*.

As diferenças de estratégia devem-se ao comportamento do *buffer* no provedor.

O Distribuidor mantém os blocos que estão sendo providos armazenados. E um cliente em pausa de provimento tem seus recursos reservados, como se estivesse recebendo o vídeo. Retomar a pausa não é a mesma coisa que se conectar novamente. Além disso, o Distribuidor é um recurso que permite mais opções de provimento e deve ser preservado, da mesma forma que o Distribuidor é utilizado para preservar os recursos disponíveis nos Servidores de Vídeo.

O *buffer* do Cliente é limitado e tem seus blocos descartados conforme o usuário assiste ao vídeo. Manter a pausa de provimento por muito tempo aumenta a probabilidade dos blocos que estão sendo recebidos serem descartados, levando o cliente receptor a ter que trocar de provedor, podendo potencialmente passar a ser atendido por um Distribuidor.

5.5 Carga de Trabalho

Os valores utilizados para carga de trabalho são descritos na Tabela 5.1. O significado dos parâmetros é descrito a seguir.

A **Taxa de chegada** indica a quantidade de clientes que se conecta ao sistema a cada minuto. Como a quantidade de clientes que pôde ser medida foi pequena – por limitações

Tabela 5.1: Carga de Trabalho

Parâmetro	Valores	Valor Padrão
Taxa de chegada (cli/min)	1,6,60	6
Vídeos distintos	1,8,16	8
Distribuidores	1	1
Servidores	1	1
Tratamento de preenchimento de buffer	Sim, Não	Não
Taxa do vídeo	1,5 Mbps	1,5 Mbps
Taxa de recepção	1,65 Mbps	1,65 Mbps
Políticas de provimento	wan,mix	mix
Tempo de teste	30 min	30 min
Tempo do Vídeo	20 min	20 min
Tamanho de <i>buffer</i>	8MB, 16 MB, 32 MB	16 MB
<i>Prefetch</i>	5s (\approx 800KB)	5s
Tamanho de <i>Slot</i>	5s	5s
Máquinas utilizadas	8 = 6 + 1 + 1	8
Clientes por máquina	2,5,7,9	5

na quantidade de clientes por máquina, conforme descrito na Seção 5.4.1 – evitou-se utilizar um Processo de Poisson. Em lugar disso, utilizou-se um espaçamento regular na chegada dos clientes. Assim, à taxa de chegada de 6 clientes/minuto, por exemplo, utilizou-se um intervalo de 10 segundos entre cada cliente.

A quantidade de **vídeos distintos** indica quantos vídeos estão disponíveis para serem assistidos pelos usuários (na verdade, não há usuários no experimento, apenas os componentes clientes).

O número de **Distribuidores** foi colocado em um (1) para maximizar a quantidade de máquinas disponíveis para executar componentes cliente.

A quantidade de **Servidores** também poderia ter sido maior. Porém, como a instalação do Servidor exige a cópia dos vídeos, este procedimento demanda uma máquina dedicada para esta função. Isso impede (ou melhor, dificulta) que o Servidor seja alocado no Cluster durante a submissão do experimento. Para obter este efeito, seria necessário copiar os vídeos em todos os nós do Cluster ou ao menos copiar os vídeos no momento da submissão. Como o objetivo é minimizar o uso do Servidor, utilizou-se somente um Servidor. A mesma máquina serve de Supervisor da Rede de Distribuição, com o MySQL instalado e configurado nesta máquina.

O item **Tratamento de preenchimento de buffer** indica se o cliente gerencia o preenchimento do *buffer*, procurando se antecipar a uma necessidade de troca de provedor.

Quando o preenchimento do *buffer* começa a decrescer, independentemente se há vídeo suficiente para ser exibido, este tratamento especial força uma troca de provedor. Assim, aproveita-se a situação confortável de ter (potencialmente) o *buffer* com vídeo acima do *prefetch* para se efetuar a troca com alta probabilidade desta ser concluída antes do vídeo se esgotar no *buffer*. Sem este tratamento (introduzido no GloVE-MIX), a troca de provedor ocorre somente quando o nível de *prefetch* é atingido ou, no pior caso, quando o vídeo se esgota no *buffer*.

A **Taxa do vídeo** indica a que taxa (em Mbps) que o vídeo é exibido – em média – pelo decodificador. O vídeo é codificado a uma taxa quase CBR porém há uma pequena variação (10-15%) na taxa de exibição.

A **Taxa de recepção** é definida como sendo 110% da Taxa média do vídeo. Isso ocorre para que o preenchimento do *buffer* aumente com o tempo, reduzindo a probabilidade da exibição ser interrompida por falta de vídeo em *buffer*.

As **Políticas de provimento** indicam que tipo de alocação de provedor é utilizada. Na política *wan* adota-se o mesmo critério utilizado para o GloVE-WAN, ou seja, o Supervisor sempre aloca um Distribuidor para prover vídeo para um cliente. Na política *mix*, utiliza-se o critério introduzido pelo GloVE-MIX. Ou seja, busca-se primeiro um outro cliente que possa prover vídeo. Se não é encontrado, aloca-se um Distribuidor para esta tarefa.

O **Tempo de teste** indica quanto tempo o experimento permanece executando nas máquinas alocadas. Como o tempo de teste é maior que o tempo do vídeo, isso significa que o vídeo será exibido até o final – ao menos no primeiro cliente.

O **Tempo do Vídeo** indica qual a duração do vídeo que será “exibido” pelos componentes cliente. Este tempo foi escolhido de forma a ser maior que o tempo de estouro do *buffer*, para os parâmetros definidos para o experimento. Assim pode-se avaliar o comportamento da solução em uma situação de pausa de provimento.

O **Tamanho de *buffer*** indica quanto de espaço é reservado em memória para armazenamento de vídeo, nos clientes. Este tamanho foi definido de forma a ser o mesmo utilizado nos experimentos do GloVE-LAN [12] e do GloVE-WAN [4]. Além disso é um tamanho adequado para uso em diversos tipos de dispositivo de exibição de vídeo, seja um PC, um Set-top box ou um dispositivo móvel.

O **Prefetch** é o tempo em que o cliente receberá vídeo antes de exibí-lo. Este preenchimento inicial do *buffer* permite o tratamento de atrasos na entrega dos pacotes e da variação (*jitter*) da taxa de transferência introduzida pelo meio (rede). Também é utili-

zado para que a transmissão seja CBR com a decodificação VBR, ainda que a variação na taxa de exibição seja pequena.

O **Tamanho de Slot** é a unidade de alocação de recursos no Distribuidor. Este tempo é importante, no contexto do cliente, para se estimar o tempo que o Distribuidor levará para responder a uma requisição (o retorno da pausa de provimento, por exemplo).

O **Tempo até o estouro** (em segundos) do *buffer* foi medido utilizando-se a taxa de recepção (em bits por segundo), a taxa do vídeo (em bits por segundo) e o tamanho do *buffer* (em bytes), como descrito na Equação 5.1.

$$Tempo_{estouro} = \frac{Tamanho_{buffer} * 8}{Taxa_{recepcao} - Taxa_{video}} \quad (5.1)$$

A quantidade de **Máquinas utilizadas** influencia na quantidade de clientes que estão em uso no experimento. Foram utilizadas 8 máquinas no cluster. Seis (6) máquinas executando componentes cliente, uma (1) máquina executando Distribuidor e uma (1) máquina utilizada para controlar o experimento. A máquina do Servidor/Supervisor é contada à parte (ou seja, foram utilizadas 9 máquinas).

A quantidade de **Clientes por máquina** é o outro fator que determina a quantidade de clientes em uso no experimento. Todos os experimentos utilizaram cinco (5) clientes por máquina, num total de 30 clientes. Apenas nos testes de sobrecarga (veja na Seção 5.6.1) foram utilizados mais clientes por máquina para se avaliar o impacto da sobrecarga de clientes.

5.6 Análise dos Resultados

A seguir são apresentados os resultados dos experimentos e é feita uma análise sobre estes resultados.

5.6.1 Sobrecarga de clientes por máquina

Os experimentos de sobrecarga buscam demonstrar o limite de clientes por máquina, nas máquinas do cluster do LCP, de forma a não ocorrer degradação artificial do experimento. A degradação artificial, neste caso, é considerada como a degradação que ocorre com o excesso de clientes em uma mesma máquina mas que não ocorreria com um cliente por máquina. Afinal, o caso típico de uso do GloVE é com um componente cliente por máquina (PC, Set-top box, etc.) e não com vários clientes. A utilização de vários clientes

por máquina ocorre, principalmente, em experimentos visando reduzir a quantidade de hardware necessária aos experimentos.

As Figuras 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9, mostram os valores obtidos utilizando a política *mix*, ou seja, com uso do CVC em conjunto com a C-CVC, e também os valores obtidos utilizando a política *wan*, ou seja, com uso apenas da C-CVC.

Verifica-se que a quantidade de fluxos oriundos do Distribuidor cresce após alguns minutos de experimento e que esta variação é mais acentuada conforme se acrescenta clientes por máquina. Isso ocorre pois a utilização de memória e CPU torna-se cada vez mais intensa, degradando o desempenho. Como o cliente não consegue utilizar a quantidade de ciclos de CPU que precisa para receber, enviar e decodificar vídeo, o provimento é transferido para os Distribuidores. Mesmo assim, o tempo de experimento é mais longo (o vídeo não termina, na verdade) pois a taxa de exibição torna-se menor que a taxa média definida para o experimento.

5.6.2 Opções de vídeo disponíveis

Quando há apenas um único vídeo disponível, a probabilidade de concentração dos usuários em pontos próximos do mesmo conteúdo aumenta. Conforme aumentamos a quantidade de vídeos disponíveis, a tendência é que a solução não seja tão favorecida.

Contudo, esta é uma consequência óbvia e esperada, uma vez que os grafos de distribuição de cada vídeo são independentes uns dos outros. Logo, não há compartilhamento de recursos, seja no Servidor, no Distribuidor ou no Cliente. Mesmo no Supervisor, os grafos são mantidos de forma independente – apesar de estarem armazenados no mesmo Banco de Dados.

O que precisa ser avaliado, é qual a distância que a solução se coloca de uma solução convencional de VoD. Ou seja, o número de fluxos oriundos do Servidor deve ser próximo da quantidade de vídeos disponíveis. Afinal, a quantidade de fluxos oriundos do Servidor respeita a Inequação 5.2.

$$Videos_{empovimento} \leq Fluxos_{servidor} \leq Total_{clientes} \quad (5.2)$$

O elemento $Videos_{empovimento}$ indica quantos vídeos estão em real provimento, ou seja, sendo enviados para algum outro componente. Pode-se ter milhares de vídeos disponíveis para escolha mas termos apenas algumas dezenas sendo realmente utilizados. O elemento $Fluxos_{servidor}$ representa a quantidade de fluxos oriundos do Servidor. O elemento $Total_{clientes}$ representa a quantidade de clientes sendo atendidos pela solução.

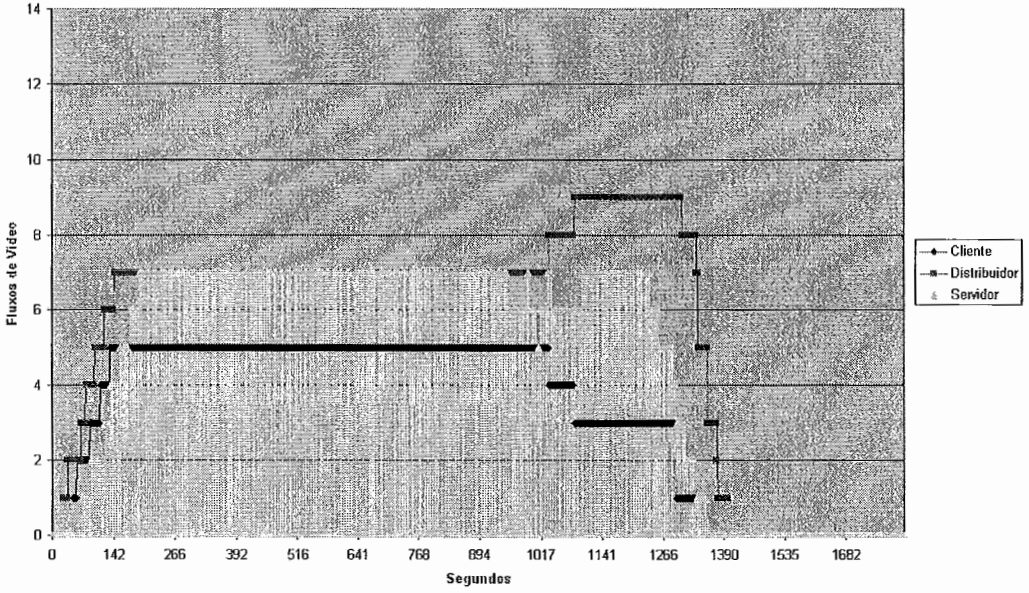


Figura 5.2: GloVE-MIX - Variação da quantidade de clientes/máquina - 2 clientes

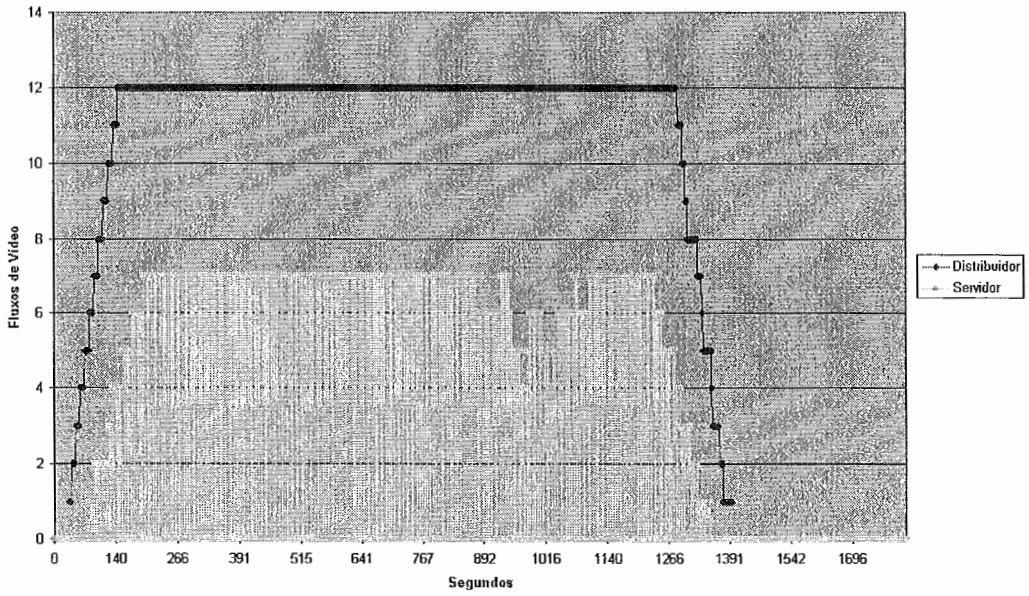


Figura 5.3: GloVE-WAN - Variação da quantidade de clientes/máquina - 2 clientes

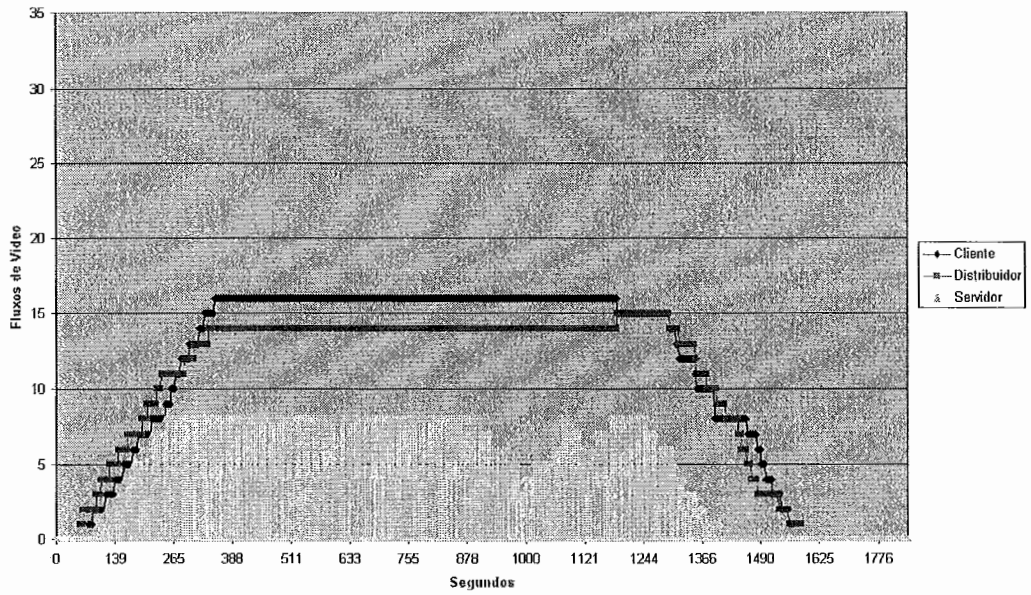


Figura 5.4: GloVE-MIX - Variação da quantidade de clientes/máquina - 5 clientes

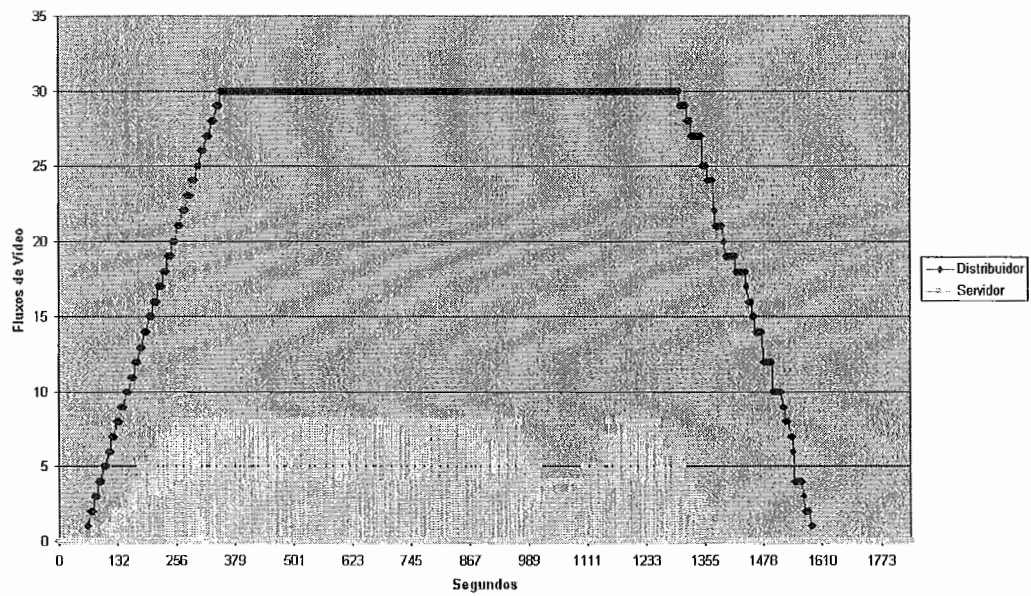


Figura 5.5: GloVE-WAN - Variação da quantidade de clientes/máquina - 5 clientes

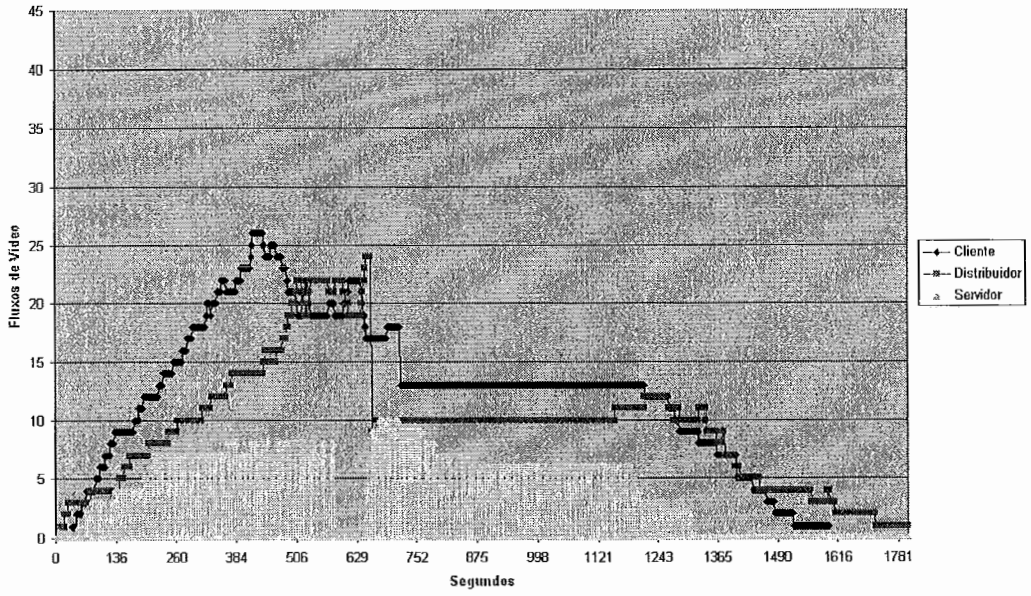


Figura 5.6: GloVE-MIX - Variação da quantidade de clientes/máquina - 7 clientes

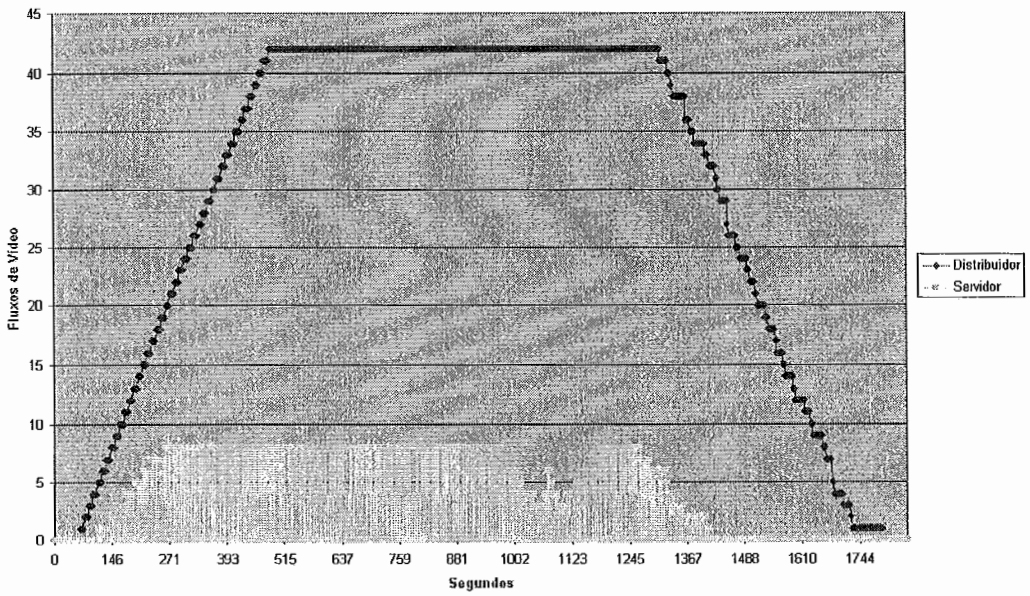


Figura 5.7: GloVE-WAN - Variação da quantidade de clientes/máquina - 7 clientes

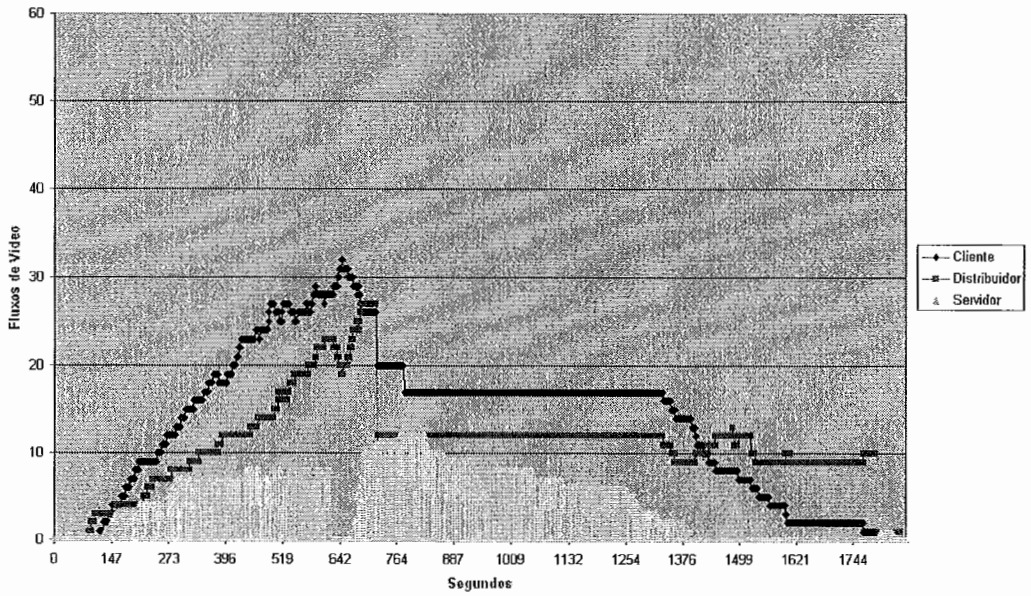


Figura 5.8: GloVE-MIX - Variação da quantidade de clientes/máquina - 9 clientes

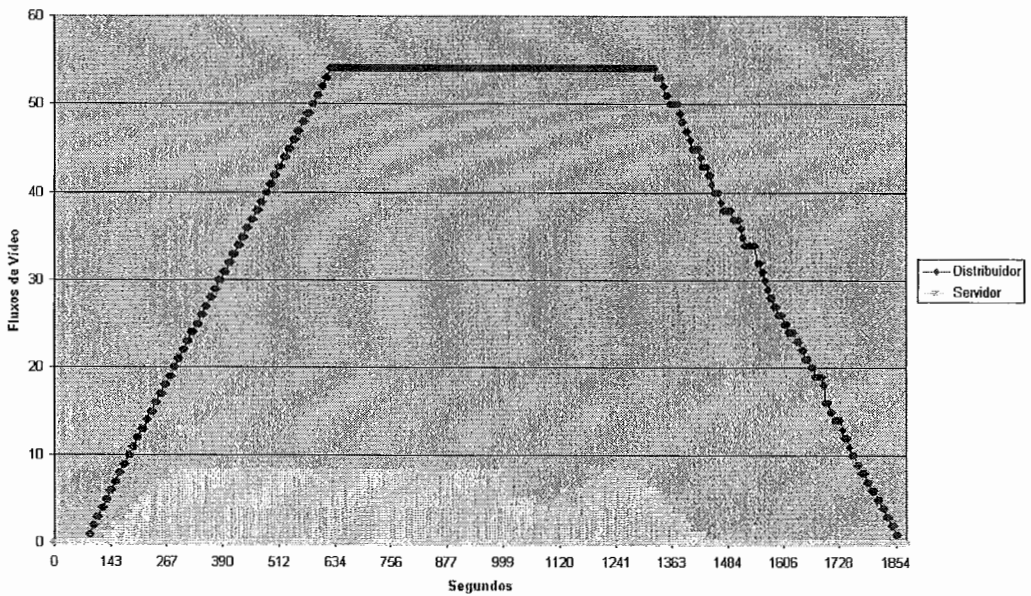


Figura 5.9: GloVE-WAN - Variação da quantidade de clientes/máquina - 9 clientes

Quanto maior a quantidade de vídeos oferecida, maior a quantidade **mínima** de fluxos oriundos do Servidor.

Além de aumentar a quantidade mínima de fluxos do Servidor, a maior quantidade de vídeos aumenta a probabilidade de blocos de vídeo serem descartados nos clientes antes que possam ser enviados para outros clientes. Por exemplo, se temos oito (8) vídeos disponíveis para provimento e se, no pior caso, eles são alocados em sequência para os clientes, que se associam ao sistema com taxa de chegada de 6 clientes/min (um cliente a cada 10 segundos), o segundo cliente para o primeiro vídeo seria alocado após 80 segundos. A uma taxa de 1,5 Mbps são necessários aproximadamente 14 MB de *buffer* para que o primeiro bloco de vídeo não seja descartado.

Como boa parte dos experimentos foi efetuada com 16 MB de *buffer*, a probabilidade de se ter provimento de um cliente para outro deveria ser satisfatória. Porém como a alocação dos vídeos utiliza Zipf de 0.7 – buscando reproduzir um comportamento que seria esperado em um cenário real [12] –, há uma certa concentração em alguns vídeos. Esta situação explica porque a quantidade de fluxos oriundos de Distribuidor é maior que a quantidade de vídeos (melhor caso) e menor que a quantidade de clientes (pior caso).

As Figuras 5.10, 5.11, 5.12, 5.13, 5.14 e 5.15 mostram os valores obtidos utilizando a política *mix*, ou seja, com uso do CVC em conjunto com a C-CVC, e também os valores obtidos utilizando a política *wan*, ou seja, com uso apenas da C-CVC.

Obviamente, a quantidade de fluxos oriundos de Distribuidor é maior na versão GloVE-WAN que na versão GloVE-MIX, uma vez que a versão GloVE-WAN *só* utiliza Distribuidores para o atendimento aos clientes. Assim, a métrica comparativa é a quantidade de fluxos oriundos de Servidor. Como a versão GloVE-MIX minimiza a quantidade de fluxos oriundos de Distribuidor, ela minimiza, conseqüentemente, a quantidade de fluxos oriundos de Servidor.

5.6.2.1 Análise das métricas

- **Fluxos oriundos do Servidor** - Em ambos os casos, a quantidade de fluxos oriundos do Servidor é pequena, graças ao trabalho que o Distribuidor faz para reduzi-la. Neste ponto a versão integrada (GloVE-MIX) parte do desempenho oferecido pela versão GloVE-WAN;
- **Fluxos oriundos de Distribuidor** - A versão GloVE-MIX reduz drasticamente a quantidade de fluxos oriundos de Distribuidor. Em boa parte do experimento o provimento é encadeado de um cliente para outro – isso é possível pois todos estão

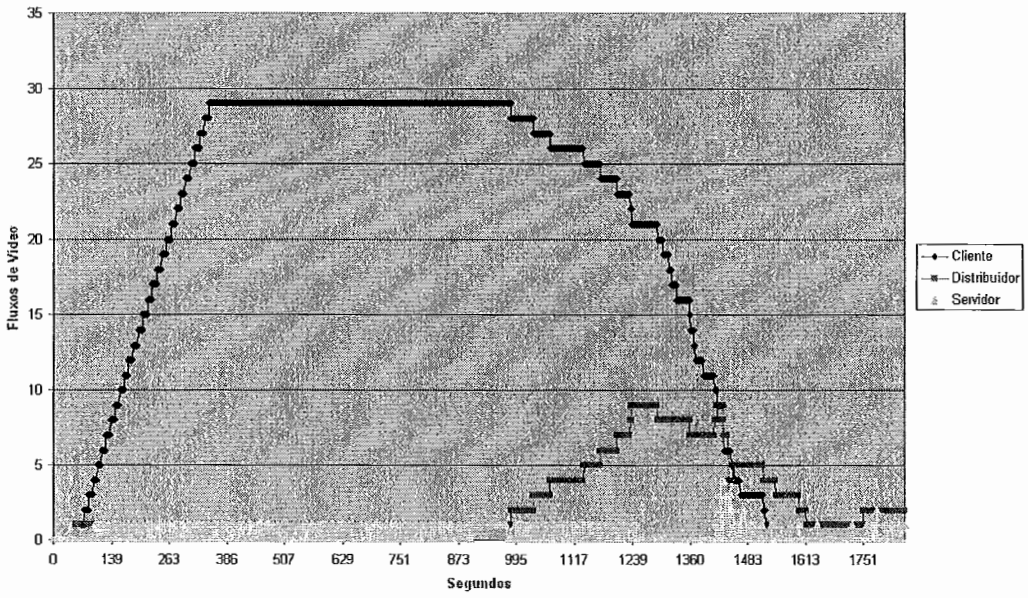


Figura 5.10: GloVE-MIX - Variação da quantidade de vídeos - 1 vídeo

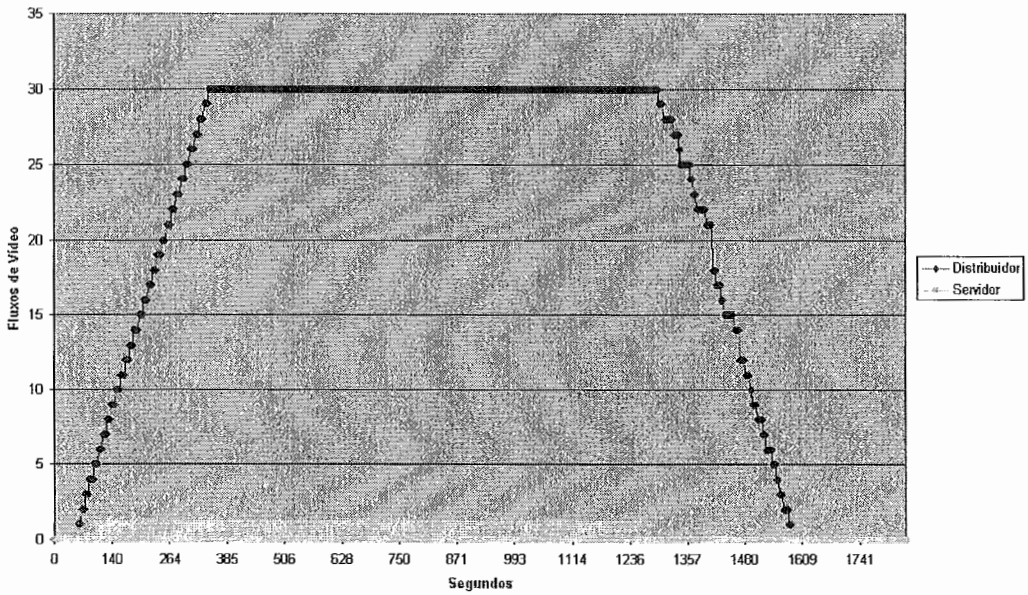


Figura 5.11: GloVE-WAN - Variação da quantidade de vídeos - 1 vídeo

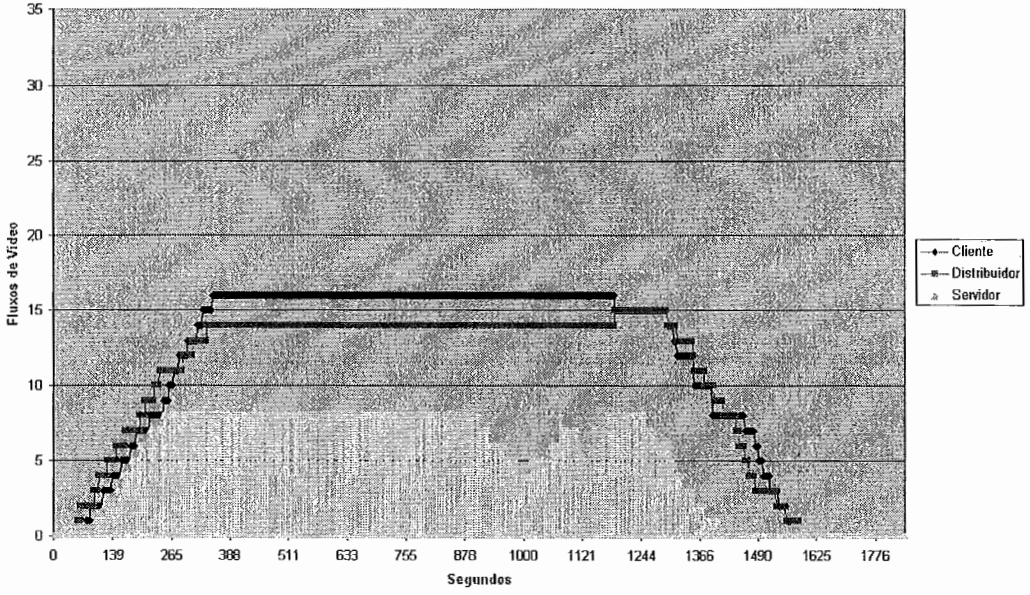


Figura 5.12: GloVE-MIX - Variação da quantidade de vídeos - 8 vídeos

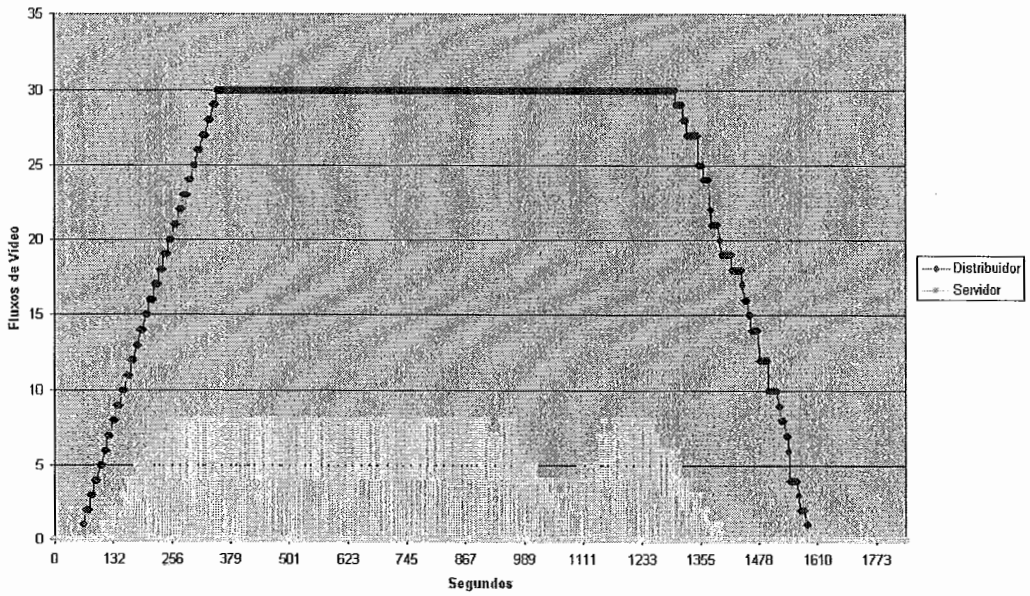


Figura 5.13: GloVE-WAN - Variação da quantidade de vídeos - 8 vídeos

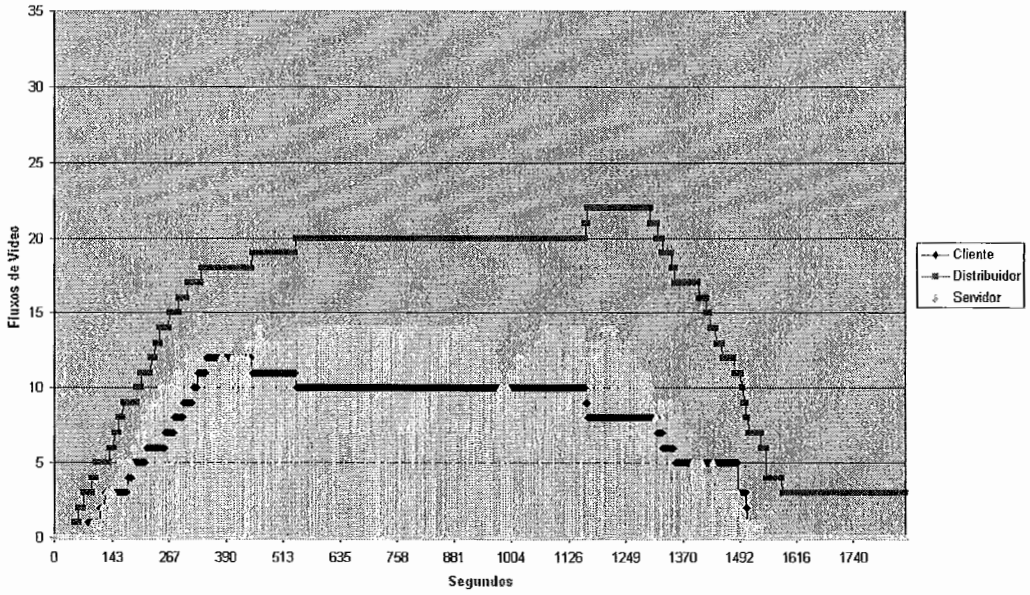


Figura 5.14: GloVE-MIX - Variação da quantidade de vídeos - 16 vídeos

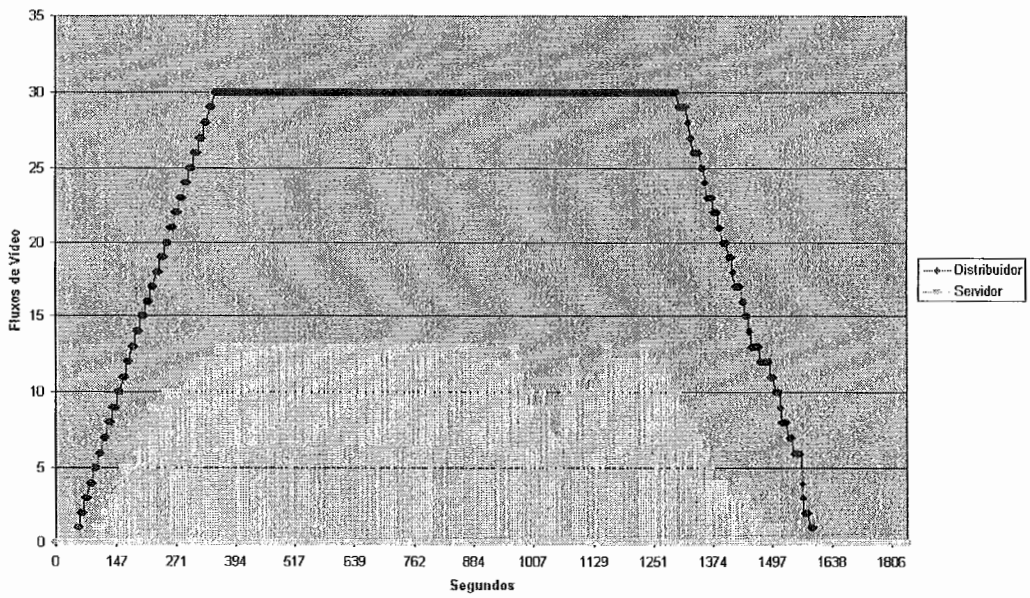


Figura 5.15: GloVE-WAN - Variação da quantidade de vídeos - 16 vídeos

na mesma rede, mas somente quando os clientes estão utilizando o mesmo vídeo – **reaproveitando** o fluxo de vídeo que o primeiro cliente recebe do Distribuidor. Ressalte-se que a quantidade mínima de fluxos é igual a quantidade de vídeos disponíveis, uma vez que o experimento utiliza todos os vídeos;

- **Fluxos oriundos de Cliente** - A quantidade de fluxos oriundos de Cliente é complementar a quantidade de fluxos oriundos de Distribuidor, pois a versão GloVE-MIX não utiliza o Servidor para atender clientes diretamente. A quantidade de fluxos oriundos de Cliente não é máxima devido à forma como a utilização dos vídeos está distribuída – conforme visto anteriormente no texto.

5.6.3 Taxa de chegada de clientes

A taxa de chegada de clientes influencia a probabilidade de se encontrar um outro cliente como provedor. Quanto menor a taxa de chegada, maior a chance dos blocos de vídeo serem descartados pelos clientes antes que estes módulos possam ser alocados como provedores.

Com 16 MB de *buffer*, e um vídeo com taxa de exibição de 1,5 Mbps, o *buffer* comporta aproximadamente 89s de vídeo. Assim, uma taxa de chegada de um (1) cliente/min (ou um cliente a cada 60s) permite que um cliente faça provimento para outro apenas quando entram no sistema um imediatamente após o outro. Se o cliente *a* solicita o vídeo *v1* e o cliente *b* solicita o vídeo *v2*, o cliente *c* só será provido por outro cliente se solicitar o vídeo *v2*. Se solicitar o vídeo *v1*, será atendido por um Distribuidor, pois o cliente *a* já terá descartado o início do vídeo.

Isso ocorre pois quando o cliente *b* entra no sistema, o cliente *a* já está exibindo vídeo há 60s – ignorando-se todo o tempo decorrido entre a solicitação do vídeo e o início da exibição. Quando o cliente *c* entra no sistema, o cliente *a* já exibe o vídeo há 120s. Como o *buffer* comporta aproximadamente 89s, necessariamente os primeiros blocos já foram descartados.

As Figuras 5.16, 5.17, 5.18, 5.19, 5.20 e 5.21 mostram os valores obtidos utilizando a política *mix*, ou seja, com uso do CVC em conjunto com a C-CVC, e também os valores obtidos utilizando a política *wan*, ou seja, com uso apenas da C-CVC.

Quanto menor a taxa de chegada, pior o aproveitamento das técnicas CVC e C-CVC. Contudo, até 6 clientes/min, o sistema se comporta de forma eficiente, mantendo uma quantidade pequena de fluxos oriundos do Servidor.

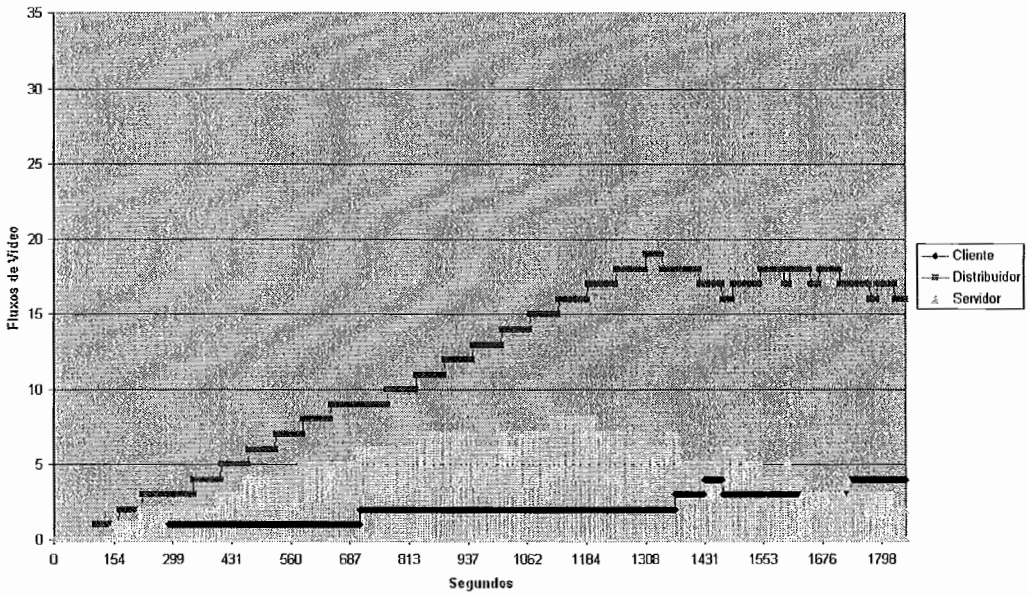


Figura 5.16: GloVE-MIX - Variação da taxa de chegada - 1 cliente/min

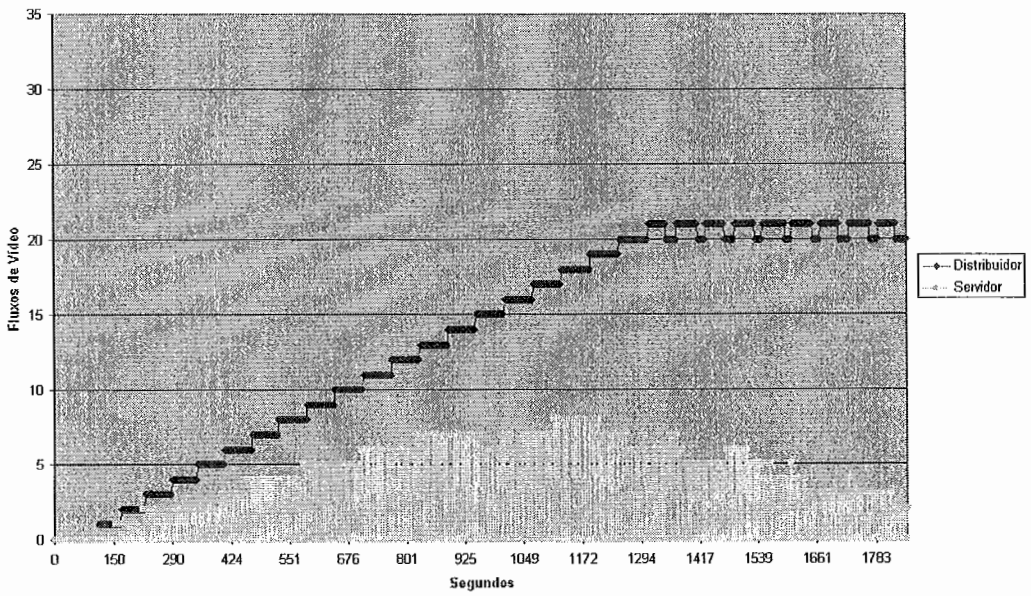


Figura 5.17: GloVE-WAN - Variação da taxa de chegada - 1 cliente/min

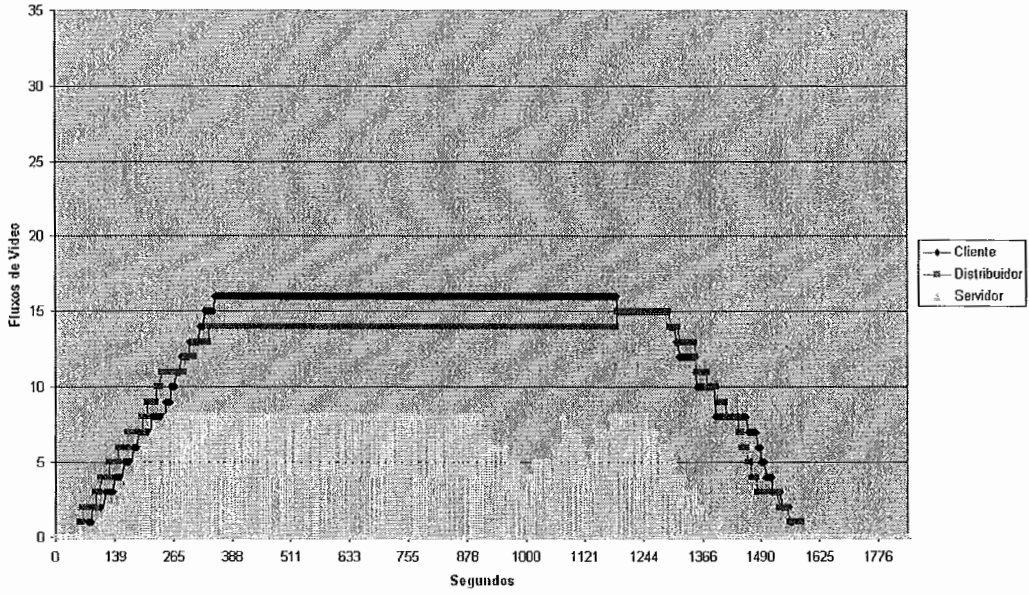


Figura 5.18: GloVE-MIX - Variação da taxa de chegada - 6 clientes/min

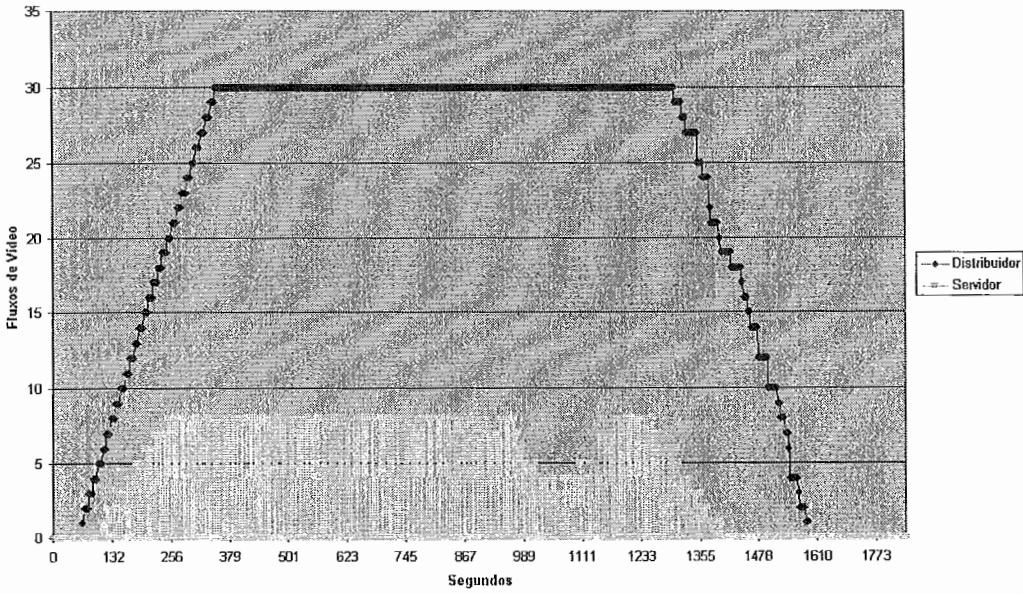


Figura 5.19: GloVE-WAN - Variação da taxa de chegada - 6 clientes/min

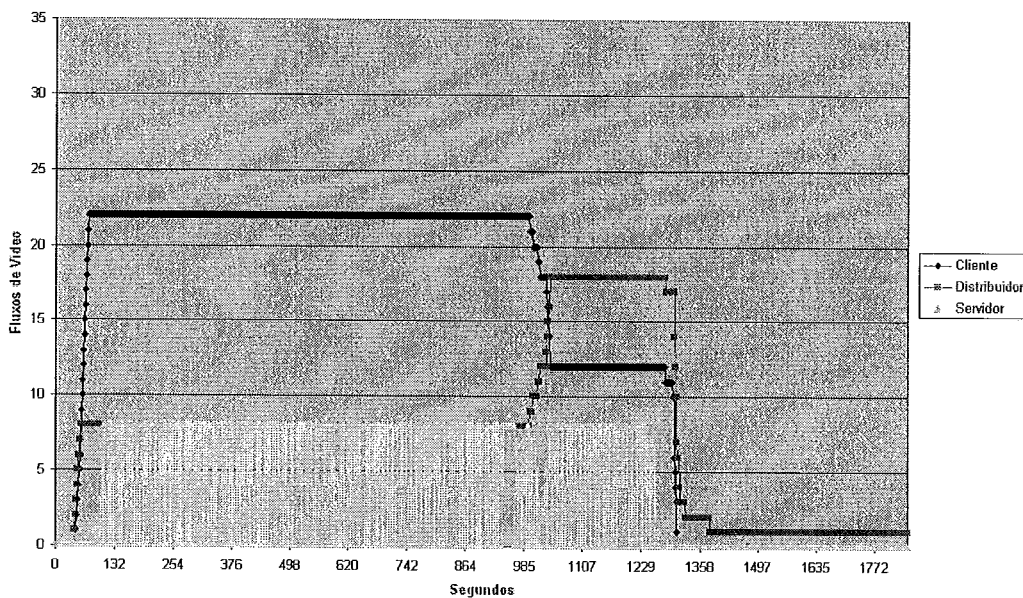


Figura 5.20: GloVE-MIX - Variação da taxa de chegada - 60 clientes/min

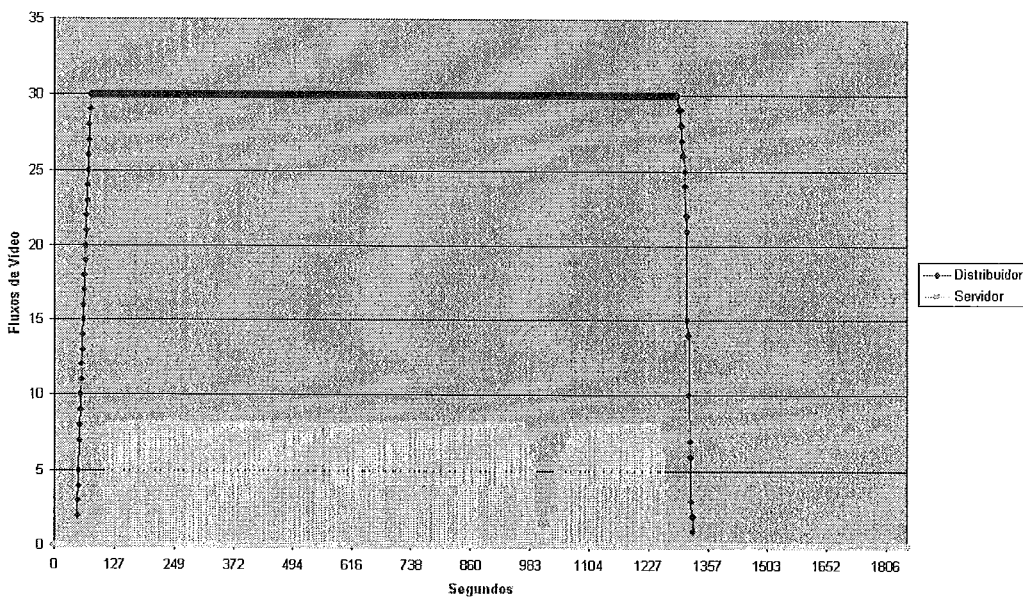


Figura 5.21: GloVE-WAN - Variação da taxa de chegada - 60 clientes/min

Quando se utiliza apenas um vídeo disponível, mesmo a uma taxa de 1 cliente/min, o provimento ainda pode ser efetuado de um cliente para outro. Porém, nos experimentos desta dissertação, utilizou-se como padrão oito (8) vídeos. Isso torna o cenário mais desfavorável para a versão integrada (GloVE-MIX), conforme pode ser observado nos gráficos da versão GloVE-MIX. Para a versão GloVE-WAN, a quantidade de fluxos oriundos do Servidor não cresce pois o *buffer* no Distribuidor é maior que um *buffer* de cliente, permitindo que a solução atenda a taxas de chegada menores.

Uma taxa de chegada de 1 cliente/min permite a ocorrência de outro efeito. Como temos 30 clientes (Cinco por máquina, Seis máquinas), são necessários 30 minutos para todos os clientes entrarem no sistema. Contudo, o vídeo tem a duração de 20 minutos. Assim, o primeiro cliente encerra o vídeo antes de todos os clientes estarem participando do experimento. Isso faz com que o total de clientes ativos se mantenha estável, mesmo com a entrada de novos clientes, causando a oscilação observada no trecho final do gráfico de “Fluxos oriundos de Cliente”.

Com uma taxa de chegada muito elevada (60 clientes/min), há alguns problemas quando os clientes começam a encerrar o vídeo. Como há vários vídeos disponíveis, o espaçamento entre clientes de um mesmo vídeo pode ser maior que a taxa de chegada. Quando o cliente *a* termina de exibir o vídeo *v1*, este encerra e **pára de prover** o cliente *d*, que também assiste a *v1*. Assim, *d* precisa trocar de provedor e, por falta de opção, este provedor é um Distribuidor.

5.6.4 Tamanho do *buffer*

Conforme já visto em seções anteriores, o tamanho do *buffer* tem influência sobre a probabilidade de se encontrar um outro cliente para provimento de vídeo, em lugar de se utilizar um Distribuidor (ou Servidor).

Com 16 MB de *buffer*, e um vídeo com taxa de exibição de 1,5 Mbps, o *buffer* comporta aproximadamente 89s de vídeo. Com 8 MB, temos 45s. Com 32 MB, temos 178s. Quanto maior a capacidade, mais tempo os blocos permanecem no *buffer* do cliente disponíveis para provimento.

As Figuras 5.22, 5.23, 5.24, 5.25, 5.26 e 5.27 mostram os valores obtidos utilizando a política *mix*, ou seja, com uso do CVC em conjunto com a C-CVC, e também os valores obtidos utilizando a política *wan*, ou seja, com uso apenas da C-CVC.

A melhora na probabilidade de provimento por um outro cliente pode ser percebida nos gráficos, pois conforme o tamanho do *buffer* aumenta, mais clientes são providos por

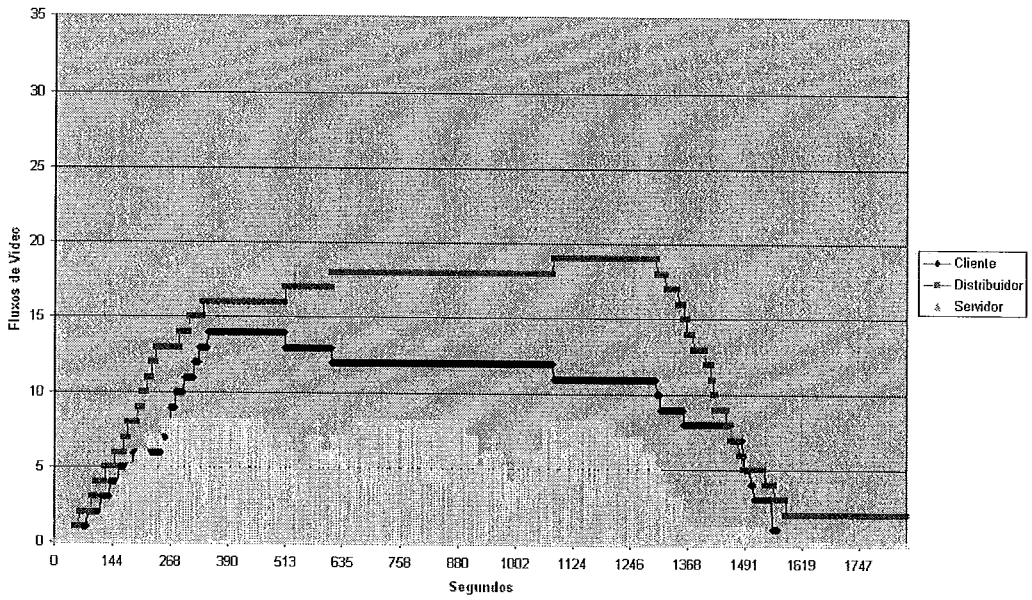


Figura 5.22: GloVE-MIX - Variação do tamanho do *buffer* - 8 MB

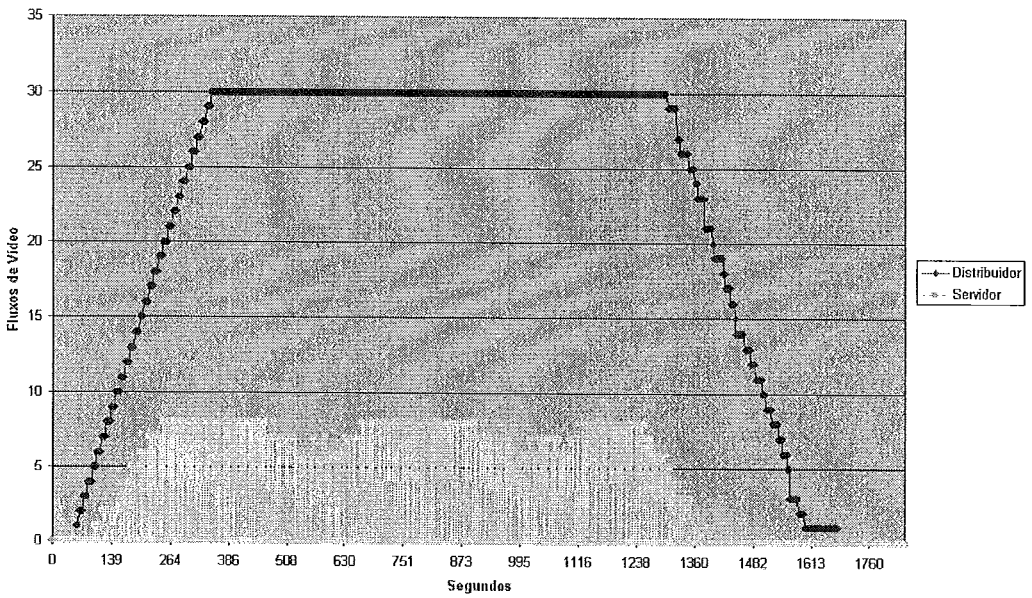


Figura 5.23: GloVE-WAN - Variação do tamanho do *buffer* - 8 MB

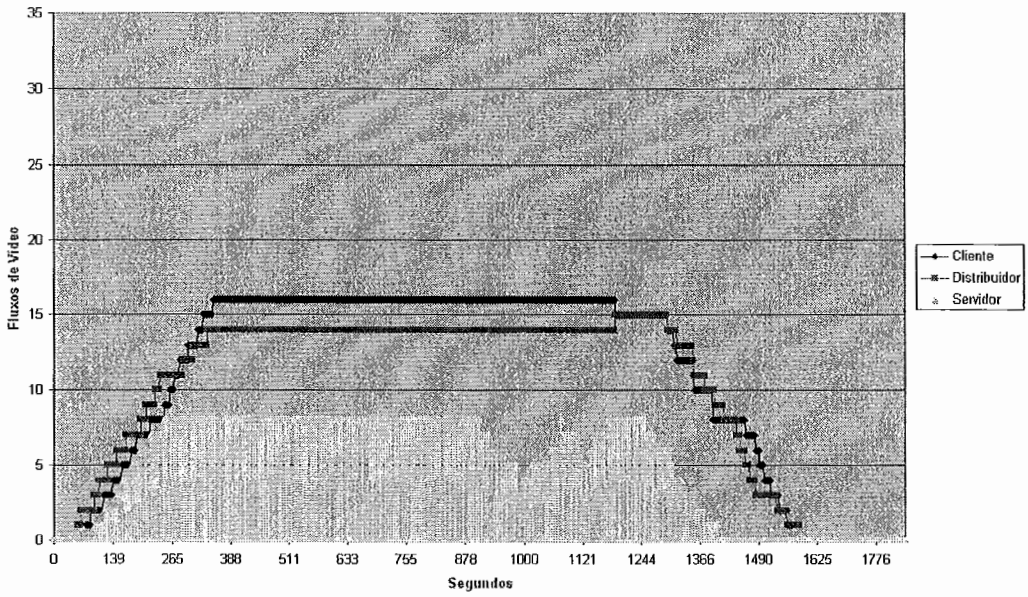


Figura 5.24: GloVE-MIX - Variação do tamanho do *buffer* - 16 MB

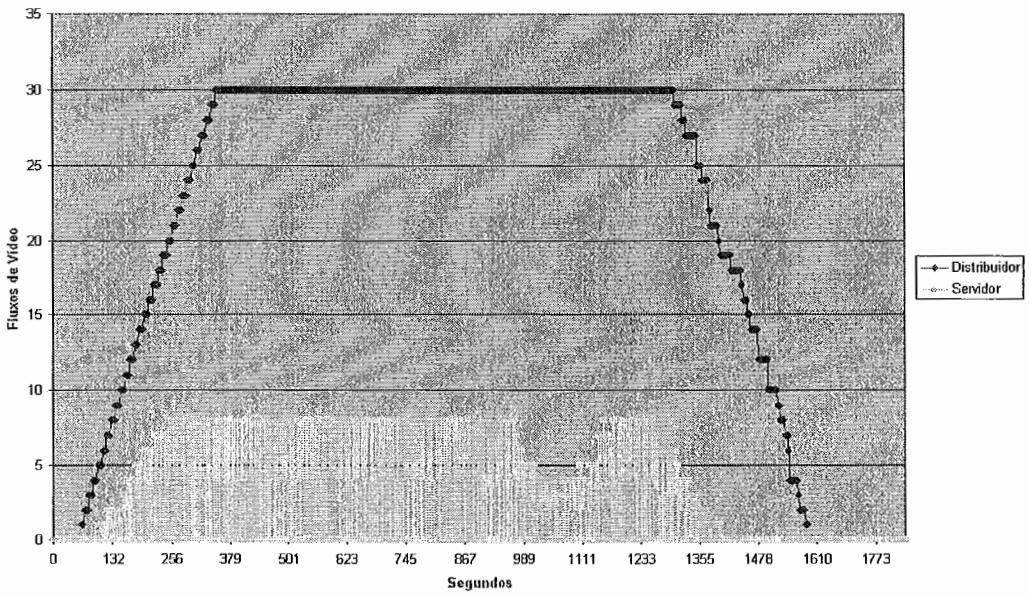


Figura 5.25: GloVE-WAN - Variação do tamanho do *buffer* - 16 MB

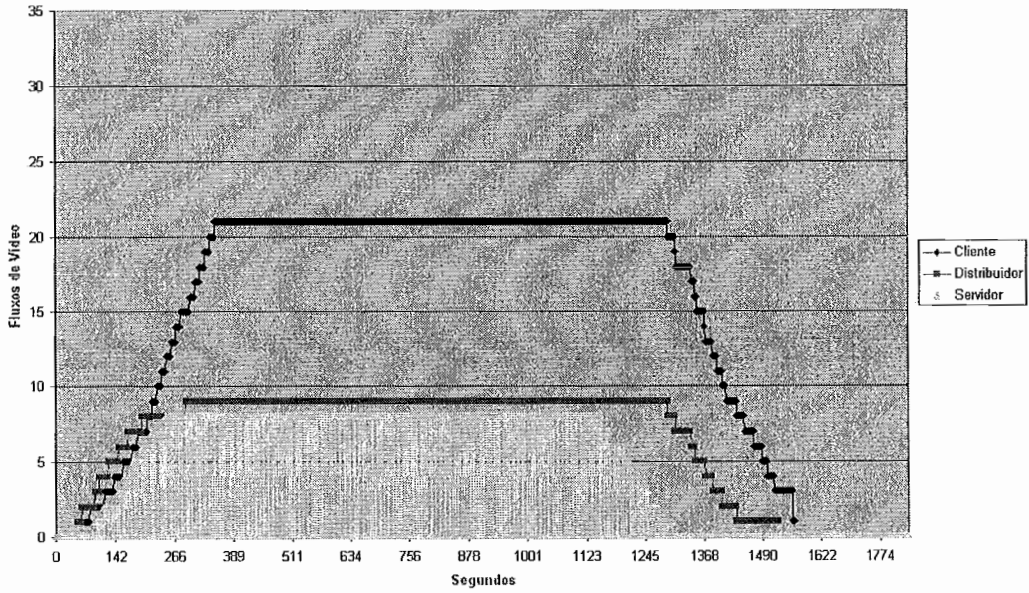


Figura 5.26: GloVE-MIX - Variação do tamanho do *buffer* - 32 MB

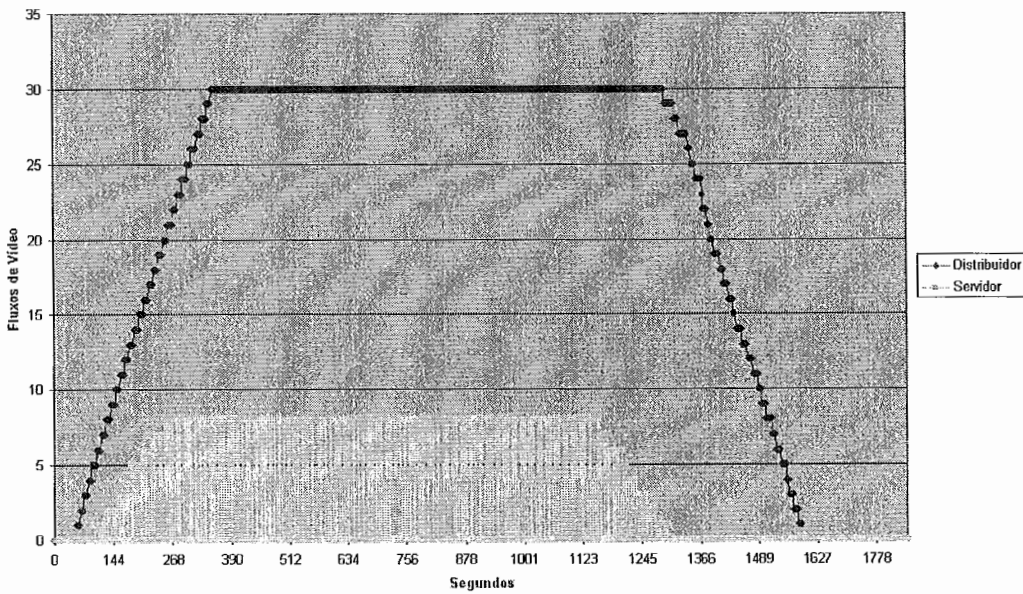


Figura 5.27: GloVE-WAN - Variação do tamanho do *buffer* - 32 MB

outros clientes.

Com 8 MB de *buffer* as limitações de tempo para tratar as atividades da recepção de vídeo ficam mais evidentes. Este experimento usou uma taxa de chegada de 6 clientes/min, logo são 10s entre um cliente e outro. Temos o *prefetch* de 5s, o tempo de *slot* no Distribuidor – que determina o tempo médio de resposta do Distribuidor para as solicitações, especialmente de inclusão, pausa e retomada de provimento – em 5s e um tempo de resposta do Distribuidor na média em 15s ($3 \times 5s$). Somando-se estes tempos temos 30s em um *buffer* de 45s (8 MB e um vídeo de taxa de exibição de 1,5 Mbps), o provimento de um cliente para outro ocorre nos limites do *buffer*, aumentando as chances do provimento ser interrompido.

Já com 16 MB de *buffer* (89s), as chances são maiores de haver um cliente disponível para provimento, especialmente se considerarmos que há 8 vídeos disponíveis, podendo haver um intervalo de 80s (ou mais) entre clientes de um mesmo vídeo, utilizando-se a taxa de chegada de 6 clientes/min. No gráfico, a relação entre provimento por Clientes e provimento por Distribuidores foi aproximadamente de 50% para cada um dos tipos.

Com um *buffer* de 32 MB (178s), as chances de o intervalo entre clientes de um mesmo vídeo ser maior que o *buffer* são pequenas. Isso pode ser percebido no gráfico, uma vez que há praticamente o número mínimo de Clientes atendidos por Distribuidores – que é igual à quantidade de vídeos em uso no experimento, de oito (8) vídeos.

Um outro aspecto de interesse é que as figuras apresentam características diferentes em relação à quantidade de Fluxos de Vídeo oriundos do Servidor. Em todos os gráficos com um *buffer* de 16 MB, há um “vale” na linha de Fluxos do Servidor. No gráfico de 8 MB há dois (2) “vales” e no de 32 MB não há “vales”. Isso acontece pois os “vales” ocorrem quando os clientes solicitam pausa de provimento, o que faz com que o *buffer* do Distribuidor possa também pausar a recepção de vídeo do Servidor, reduzindo a quantidade de Fluxos oriundos de Servidor. Com 8 MB de *buffer*, o **estouro** da capacidade do *buffer* ocorre mais vezes. Já com 32 MB, o experimento pode prosseguir até o final, sem haver estouro na capacidade do *buffer*, para um vídeo exibido à taxa de 1,5 Mbps e recebido à taxa de 1,65 Mbps, aproximadamente.

Quanto maior o *buffer*, maior a probabilidade de um outro cliente prover. Esta premissa, obviamente, não se aplica à versão GloVE-WAN, uma vez que esta **sempre** utiliza um Distribuidor para prover um cliente. Sendo assim, o tamanho do *buffer* não influencia a eficiência da distribuição. O único efeito de um *buffer* maior na versão GloVE-WAN é ter menos pausas de provimento por estouro do *buffer*.

5.6.5 Custo de comunicação

Para que a rede de distribuição possa ser monitorada, os componentes permanecem em constante comunicação com o Supervisor. Uma mensagem é enviada a cada segundo por cada um dos componentes. O tamanho da mensagem é fixo em ≈ 1 KB (1048 bytes). Desta forma, o impacto da comunicação ocorre com o crescimento da quantidade de componentes (clientes, distribuidores, servidores), ou seja, com o aumento do número de vértices no grafo de distribuição.

Nos clientes, este uso de vazão efetiva extra de 8 Kbps pode ser considerado desprezível. Nos servidores e distribuidores, esta vazão pode ser afetada pelo número de vídeos oferecidos pelo distribuidor/servidor. Em uma mensagem de 1 Kbps podem trafegar 256 identificadores de vértice na rede de distribuição, permitindo uma única mensagem de *heartbeat* para até 256 vídeos. Caso o componente sirva (ou distribua) mais vídeos, mensagens adicionais de *heartbeat* são necessárias. Uma otimização que pode ser efetuada, reduzindo este impacto, é identificar cada servidor/distribuidor unicamente, enviando apenas uma mensagem de *heartbeat*. Desta forma, o impacto seria $O(1)$ para cada componente, independentemente da quantidade de vídeos oferecidas. Assim, a vazão necessária nos Distribuidores e Servidores também pode ser considerada desprezível.

O maior custo da comunicação ocorre no Supervisor. A complexidade da comunicação é $O(n)$, onde n é o número de vértices do grafo de distribuição. As mensagens trafegam a 8 Kbps dos componentes para o Supervisor. Desta forma, o Supervisor precisa ter uma vazão **efetiva** de rede de $n \times 8$ Kbps. Logo, 10 componentes (distribuidores, servidores, clientes) demandam 80 Kbps de vazão efetiva no Supervisor. E 1000 componentes demandam 8 Mbps.

Esta vazão de rede necessária pode ser reduzida se a periodicidade do *heartbeat* for também reduzida. Aumentar o intervalo de 1 para 5 segundos reduz a vazão efetiva de rede necessária em 80%. Contudo, isso reduz também a precisão do monitoramento do Supervisor, especialmente sobre o *buffer* dos clientes. Esta redução na precisão piora a qualidade das decisões do Supervisor, no momento da escolha de um cliente para prover outro.

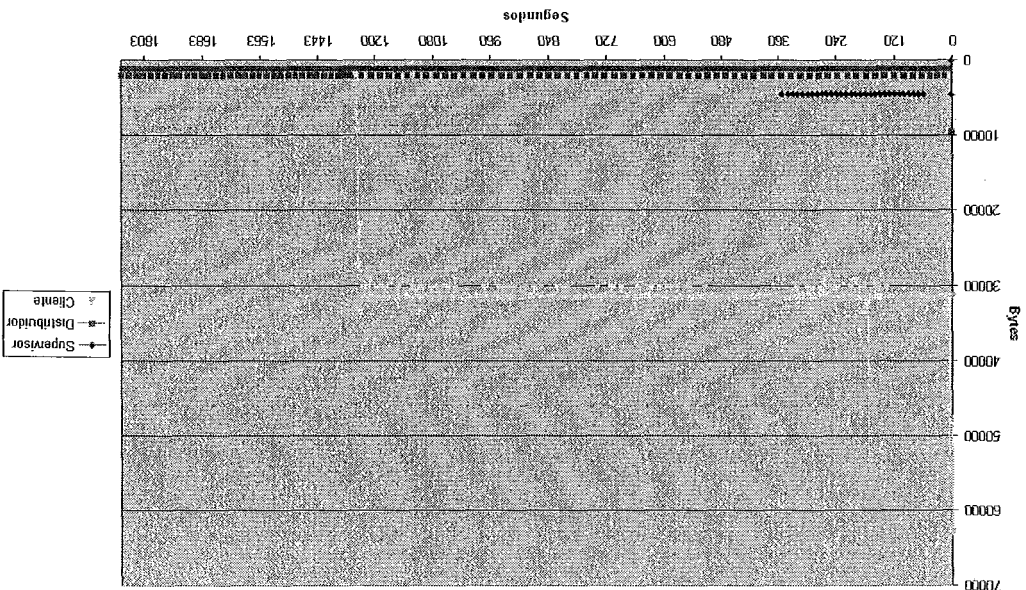
No intuito de melhorar esta precisão, podemos adotar um mecanismo de **previsão** do andamento do *buffer*. Ou seja, com base na taxa do vídeo e nas últimas informações enviadas pelo cliente, o Supervisor infere qual a posição corrente do vídeo decorridos x milissegundos desde a última atualização. Este mecanismo pode ser adotado independentemente

Os picos nas mensagens dos clientes são causados por particularidades na comunicação dos clientes/distribuidores com o Supervisor. O primeiro pico é causado pelas mensagens de inserção na Rede GioVE e pela solicitação dos vídeos disponíveis. O segundo pico é causado quando os clientes iniciam o aviso de que o seu *buffer* se deslocou até o ponto de iniciar o descarte de blocos – os que passam a estar fora da janela visível. O pico final ocorre com os clientes informando que estão encerrando o vídeo e pelos clientes que, pela saída de seu provedor, precisam de troca de provedor.

A Figura 5.28 mostra o resultado da aferição do custo de comunicação em um dos experimentos. A figura mostra o padrão relativamente constante do impacto da comunicação entre o Supervisor e os demais componentes. O maior impacto é na comunicação com os clientes. Isso acontece pois este é o tipo de componente que ocorre em maior número e, como visto antes, o volume de comunicação cresce com a quantidade de componentes. Praticamente não há variação entre as diversas configurações de experimento. Assim, optou-se por mostrar apenas um dos gráficos.

procedimento não foi adotado nesta dissertação para simplificação do Supervisor. Apesar de presente no Gerente da versão GioVE-LAN, este da granularidade do *heartbeat*, ou seja, mesmo com o envio de atualizações a cada 1s este cálculo pode ser adotado. Conforme mencionado no texto, este custo é relativamente constante ao longo do experimento e também entre as diversas configurações testadas.

Figura 5.28: GioVE-MIX - Custo da comunicação - Conforme mencionado no texto,



5.7 Discussão

Os resultados obtidos demonstram que a versão integrada (GloVE-MIX) beneficia-se da eficiência demonstrada pelas versões GloVE-LAN e GloVE-WAN. Quando o cenário é desfavorável, a forma eficaz de distribuição de vídeo adotada pela C-CVC permite que o custo da solução de VoD permaneça baixo, reduzindo a pressão por recursos no Servidor de Vídeo.

Quando o cenário é favorável, a versão GloVE-MIX, ao utilizar a técnica CVC em conjunto com a C-CVC, permite que a demanda por recursos nos Distribuidores também seja reduzida, melhorando a eficiência do sistema como um todo.

Entre os cenários favoráveis percebe-se que uma taxa de chegada relativamente alta (6 clientes/min) com uma alta concentração em alguns vídeos (algo como 1 cliente/min, se considerarmos apenas um vídeo) e um *buffer* no cliente grande (acima de 16 MB) permite que a técnica de CVC adicione eficiência ao sistema.

A Equação 5.3 mostra a relação entre as variantes dos experimentos. O elemento $\frac{1}{Taxa_{chegada}}$ indica a quantidade de segundos entre cada chegada de cliente – ou seja, 6 clientes/min = 6 clientes a cada 60s ou 10s entre cada cliente. A $Taxa_{video}$ é a taxa do vídeo em *bps* – ou seja, 1,5 Mbps = 1500000 *bps*. O $Tamanho_{buffer}$ é medido em *bytes* – ou seja, 16 MB = 16777216 *bytes*. Quando $Fator_{cvc} < 1$, a CVC poderá ser utilizada e um cliente poderá prover outro, reduzindo a demanda pelo Distribuidor e, conseqüentemente, pelo Servidor.

$$Fator_{cvc} = \frac{1}{Taxa_{chegada}} \times \frac{Taxa_{video}}{Tamanho_{buffer} \times 8} \quad (5.3)$$

É importante ressaltar que os grafos de distribuição de vídeo são **independentes** para cada vídeo considerado. Da mesma forma que $Taxa_{video}$ é relativa ao vídeo que se quer analisar, $Taxa_{chegada}$ também deve ser. Por exemplo, se a $Taxa_{chegada}$ é de 6 clientes/min, o cliente *a* solicita o vídeo *v1*, o cliente *b* solicita o vídeo *v2* e o cliente *c* solicita o vídeo *v1* novamente, a taxa de chegada (considerando somente estes casos) do vídeo *v1* é de 3 clientes/min (ou um cliente a cada 20s). Este é o valor que deve ser considerado na Equação 5.3.

Mesmo quando a CVC não pode ser utilizada, a versão GloVE-MIX oferece maior **flexibilidade** na distribuição de VoD. Como a decisão de usar um Distribuidor ou um outro Cliente é automática, o Administrador da solução de VoD não precisa configurar previamente os seus clientes para um ou outro caso.

Contudo, dado o comportamento dependente da utilização pelos usuários que a CVC possui, não se deve efetuar um planejamento de vazão de rede presumindo que será utilizado CVC em um trecho de rede. Por exemplo, se vários usuários recebem vídeo através de um canal ADSL de 8 Mbps, pode-se garantir que 8 (por segurança, 7) usuários simultâneos poderão receber vídeo de um Distribuidor localizado na rede da operadora de telefonia. Se o cenário for favorável à CVC, podemos ter mais usuários assistindo ao vídeo sem pressionar a demanda pelo canal de comunicação. Contudo, esta premissa não pode ser **garantida**.

Logo, a versão GloVE-MIX aumenta as opções de configuração da solução de VoD. A CVC não tem como garantir o provimento *peer-to-peer*, pois esta depende do comportamento do usuário. Mas ter a possibilidade de utilizá-la automaticamente pode aliviar a demanda por recursos na solução de VoD, **especialmente** nos momentos de pico.

Capítulo 6

Qualidade de Serviço

6.1 Introdução

Conforme a solução de VoD migra do âmbito da pesquisa para o provimento de serviço a usuários reais, algumas características importantes deixam de ser diretamente notadas. A grande utilização de recursos de rede, por exemplo, não é percebida exceto quando a falta da vazão de rede necessária impede a recepção do conteúdo. Se uma solução tem baixa escalabilidade, os primeiros usuários não perceberão este fato. De acordo com o crescimento da base de clientes é que esta deficiência da solução é percebida.

Assim, algumas qualidades perseguidas durante o desenvolvimento da solução, tais como baixo custo, redução no uso de recursos, escalabilidade não são diretamente percebidos pelos usuários. Essas características são fundamentais para que o produto seja viável, especialmente comercialmente. Contudo, a sua ausência muitas vezes é mascarada, seja reduzindo-se a quantidade de usuários que podem utilizar o serviço, seja comprometendo a qualidade do serviço (imagem, som, etc.) que é oferecido ou simplesmente fazendo-se uso de uma grande quantidade de recursos computacionais (discos, servidores, etc.) para que o serviço possa ser oferecido.

Outras características do serviço não podem ser disfarçadas e suas ausência é rapidamente percebida pelos usuários, afetando a opinião que estes têm do serviço prestado. Os primeiros aspectos que vêm à mente são a qualidade de imagem e som. Contudo, estes itens têm forte influência subjetiva e, sobretudo, são fruto de uma avaliação comparativa. Os usuários que nunca tiveram contato com uma imagem de alta definição e um som de alta qualidade podem considerar satisfatória uma transmissão de vídeo a uma baixa taxa, com baixa resolução e qualidade de som ruim. A própria qualificação de qualidade, muitas vezes, utiliza valores comparativos para melhor compreensão. Daí surgem as expressões “som de FM”, “som de CD”, “vídeo em qualidade de DVD”, etc.

No entanto, mesmo este QoS relativo tem alguns limites mínimos. Um som ininteligível ou com muitos chiados ou uma imagem com muita interferência não são facilmente aceitos por usuários de um serviço. Há exceções, quando o serviço é gratuito ou extremamente necessário. Vide o caso das péssimas transmissões de imagens de *webcams* por meio de enlaces com baixa largura de banda. Em alguns casos, pode ser o único meio de contato (ou ao menos, o mais econômico) com um parente ou amigo distante. Neste caso, tolera-se uma sequência de imagens com pausas e som entrecortado. Contudo, dificilmente este seria uma qualidade de serviço esperada de uma transmissão de um filme em *pay-per-view*, por exemplo.

6.2 Garantia de não interrupção

Uma garantia aparentemente simples que pode ser considerada a mais incômoda quando ausente — especialmente em conteúdo de longa duração e grande envolvimento do usuário, como um filme — é a **garantia de não interrupção**. Esta garantia é definida pela exibição ininterrupta do conteúdo de mídia contínua em sua totalidade. Com isso os usuários têm a segurança de que não assistirão apenas a um trecho do conteúdo.

Algumas soluções de distribuição de mídia contínua (TV aberta, por exemplo) têm por premissa interrupção programada da exibição do conteúdo, para apresentação de intervalos comerciais, por exemplo. Porém, esta interrupção é esperada e, mais importante, ela é **temporária**. Após o intervalo comercial o conteúdo/programa volta a ser exibido. Mesmo alguns cortes no conteúdo são tolerados, seja por passarem despercebidos do usuário ou por manterem, de certa forma, a coerência do conteúdo sendo exibido.

Porém, se esta interrupção é não programada, efetua cortes perceptíveis no conteúdo ou, o que seria pior, não conclui a exibição do conteúdo, a percepção que o usuário tem do serviço é fortemente degradada. Isso pode ser percebido pelos usuários de um programa em uma TV por assinatura via satélite, onde em dias de forte chuva pode levá-los a assistir a apenas parte do programa.

Em uma transmissão de vídeo sob demanda através de redes de computadores, várias variáveis estão envolvidas na garantia de não interrupção. A primeira questão é se o serviço é **tolerante a falhas**, ou seja, se algum componente da cadeia de distribuição, ou melhor, da **rede de distribuição de conteúdo** — CDN, vide Cap. 2 — falhar não haverá impacto na continuidade da exibição para o usuário. Este assunto será novamente discutido na Seção 6.2.1.

Mesmo em serviços sem previsão de falha (os componentes são considerados estáveis ou o serviço simplesmente não oferece garantia contra falha), a garantia de não interrupção gera requisitos para o serviço de VoD. A quantidade de recursos disponível é limitada em um dado momento, seja pela capacidade dos canais de transmissão, pela vazão de disco dos servidores ou pela quantidade de memória dos componentes da CDN, entre outros fatores. Aceitar usuários além dos limites impostos pelos recursos disponíveis tem o potencial de degradar o sistema e, em última instância, interromper completamente o serviço.

O serviço de VoD precisa garantir que está operando dentro dos limites, em qualquer momento dado, para que se possa oferecer garantia de não interrupção aos usuários. Isso implica em um **controle de admissão** dos usuários ao serviço, conforme analisado na Seção 6.2.2.

Ainda que o serviço de VoD seja tolerante a falhas, tenha controle de admissão e a operação do serviço esteja dentro dos parâmetros esperados — em quantidade de falhas, usuários admitidos, etc. — estes elementos não garantem a entrega ininterrupta de conteúdo ao usuário. O sistema como um todo pode estar dentro dos parâmetros contido também é importante que **cada componente** esteja operando dentro de seus parâmetros previstos. Se um determinado componente estiver sobrecarregado, enquanto outro que poderia dividir a carga está ocioso, ocorre o mesmo problema que o previsto sem um controle de admissão. Assim, o **balanceamento de carga** entre os componentes é um fator importante para a garantia de não interrupção de exibição, conforme visto na Seção 6.2.3.

Quando há opção de uso de redes sem fio, alguns aspectos precisam ser considerados pelo serviço de VoD. Em geral, a simples troca de pontos de acesso é tratada pela própria rede, não necessitando de interferência do serviço de VoD. Contudo, conforme visto na Seção 2.4.1, a interferência do serviço de VoD pode ser necessária. As questões sobre **mobilidade** serão discutidas na Seção 6.2.4.

6.2.1 Tolerância a falhas

O tema da tolerância a falhas é um assunto estudado há muito tempo em computação. Usualmente a tolerância a falhas é implementada com a existência de componentes redundantes para o atendimento para o provimento de uma determinada funcionalidade.

Vários elementos da cadeia de distribuição de conteúdo podem falhar, impedindo a entrega de conteúdo ao usuário. Em primeiro lugar, o próprio equipamento do usuário

pode apresentar uma falha que impeça a exibição do conteúdo. Falhas no sistema operacional, quebras de disco, falhas de memória e quedas na rede são apenas alguns dos eventos que podem interromper a exibição¹. Contornar as falhas no cliente requer soluções mais complexas, pois envolvem implicações ergonômicas. Afinal, o usuário está interagindo — olhando para o monitor, utilizando o mouse/controlador remoto, etc. — diretamente com o equipamento. Tratar a falha do equipamento cliente foge ao escopo deste trabalho.

Outro ponto de falha é a conexão do cliente com a CDN. Tratar este tipo de falha também exige redundância, com ao menos dois caminhos de rede distintos. A grosso modo, é um problema similar à falha de conexão entre os componentes da CDN. Esta dissertação não enfoca, diretamente, os problemas de tolerância a falhas em redes. Indiretamente, esta situação será considerada como um falha do componente ao qual se estava conectado. Em algumas situações específicas, se fará uma tentativa de reconexão, mas esta abordagem é superficial e não representa um mecanismo propriamente dito de tolerância a falhas na rede.

Em relação aos componentes da CDN, diversas partes destes componentes podem falhar, da mesma forma que descrito para a máquina cliente. Este trabalho não considera as falhas em parte de um componente, como um disco com defeito. Outras soluções de tolerância devem ser usadas, como RAID por exemplo, se for necessária — ou desejável — a tolerância a falhas em partes dos componentes. Porém, tratar um componente da CDN como um elemento atômico, que funciona ou falha completamente, mantém a coerência em termos de uso de máquinas comuns (COTS - *Commercial off-the-shelf*) para a distribuição de conteúdo multimídia, conforme proposto pelo projeto GloVE.

Identificar uma falha em um componente da CDN pode ser obtido através de alguns artifícios simples. Um deles é a adoção de um *heartbeat* entre os componentes e uma gestão ativa por parte de quem utiliza o componente. Por exemplo, um cliente que receba vídeo de um distribuidor, mantém um sinal de *heartbeat*, informando ao distribuidor que está em funcionamento. O cliente sabe que o distribuidor está no ar pois recebe conteúdo através dele. Quando o cliente detecta uma interrupção na recepção, informa ao Supervisor da CDN (responsável pela gestão da CDN, conforme descrito na Seção 4.4) que o distribuidor não está mais enviando conteúdo, solicitando um outro distribuidor. Se o distribuidor deixa de receber o *heartbeat* do cliente, infere que este não está mais

¹Assim como a falha no próprio sistema de VoD. Neste caso foram consideradas somente as falhas externas

funcionando, interrompendo o envio de vídeo para este cliente. Este procedimento evita desperdício de recursos.

Por ser uma aplicação de *soft realtime*, o serviço de VoD coloca um desafio extra para a solução com *heartbeat*. Não se pode transcorrer muito tempo entre a falha e a detecção da falha. Se um distribuidor falha, todos os clientes atendido por ele deixarão de receber conteúdo. O cliente precisa identificar que o distribuidor parou de enviar conteúdo, informar ao Supervisor que precisa de um outro provedor, receber a resposta, solicitar conteúdo para o novo provedor e este iniciar a transmissão. Todos estes passos devem ocorrer em tempo pequeno o suficiente para que apenas o *buffer* do cliente possa atender à exibição do vídeo.

O tempo de *prefetch* e a taxa de transferência com que o vídeo é recebido são determinantes para o tempo de *buffer* que estará disponível para o tratamento da falha. Em resumo, o **tempo de migração** — transcorrido entre a detecção da falha e o início do recebimento pelo novo provedor — é um dos fatores a serem considerados no cálculo do *prefetch* necessário.

6.2.2 Controle de Admissão

Controlar a utilização dos recursos disponíveis para o serviço de VoD é fundamental para a garantia de não interrupção da exibição. Extrapolar os recursos alocados ao serviço levará a uma degradação na distribuição de conteúdo. Por exemplo, utilizar mais vazão de rede do que o previsto para um distribuidor levará a uma taxa efetiva de transferência menor para alguns ou todos os clientes. Esta redução na taxa de transferência acarretará uma redução na quantidade de conteúdo disponível no *buffer* do cliente. Em última instância, este *buffer* se esvaziará, causando uma interrupção na exibição.

Não havendo controle nos recursos utilizados, a quantidade de clientes em atendimento pode ser maior que **toda** a capacidade de atendimento do serviço, levando a uma potencial degradação geral. No pior caso, a degradação será uniforme, interferindo no atendimento de **todos** os clientes.

Para evitar este cenário, utiliza-se um **controle de admissão** ao serviço. Este controle precisa estimar:

- A quantidade de recursos disponíveis em cada componente
- A quantidade de recursos necessária para o cliente em admissão
- A quantidade de recursos em uso por cada componente

Os recursos controlados envolvem um ou mais itens como: vazão de rede, vazão de disco, memória disponível, CPU disponível, etc. Quanto mais itens forem controlados, melhor a precisão no controle e, conseqüentemente, menor a possibilidade de exaustão dos recursos alocados ao serviço de VoD.

6.2.3 Balanceamento de Carga

Controlar os recursos disponíveis de maneira global auxilia à operação do serviço de VoD. Contudo, os problemas decorrentes da exaustão dos recursos totais alocados ao serviço podem ser percebidos mesmo se ocorrem no âmbito de um único componente. O controle de admissão pode admitir um cliente ao serviço pois a quantidade total de recursos disponíveis é suficiente. Mas, ao alocar o novo cliente a um componente sobrecarregado, o serviço prestado por este componente começa a degradar, afetando ao novo cliente ou, no pior caso, a todos os clientes atendidos pelo componente sobrecarregado.

Aplicar um **balanceamento de carga** entre os componentes é um pouco mais complexo em uma aplicação de VoD, ao se buscar as alternativas mais eficientes. Simplesmente ter todos os componentes com a mesma quantidade de recursos utilizados ou de clientes em atendimento não é suficiente para se obter uma eficiência melhor.

Por exemplo, para melhorar a escalabilidade, o GloVE utiliza a característica de popularidade dos vídeos para alocar provedores aos clientes. Na alternativa *peer-to-peer*, aloca-se um outro cliente que esteja assistindo ao mesmo vídeo. Se estes clientes (provedor e receptor) estão na mesma rede e esta suporta *multicast* (ou *broadcast*), é mais eficiente, e consome-se menos recursos, se este provedor for utilizado para atender um terceiro cliente que esteja assistindo ao mesmo ponto no vídeo. Mesmo se a rede não suportar *multicast*, utilizar o mesmo provedor será mais eficiente do que alocar um outro distribuidor (ou outra entrada do mesmo distribuidor) para o provimento destes clientes.

Na alternativa utilizando os distribuidores, alocar um distribuidor que já esteja atendendo àquele vídeo é mais eficiente que alocar um segundo distribuidor. Utilizando o mesmo distribuidor, em pontos próximos do vídeo, a quantidade de fluxos oriundos do servidor se mantém a mesma, reduzindo o uso de recursos do serviço de VoD.

Assim, um balanceamento de carga eficiente para um serviço de VoD não pode simplesmente alocar os recursos de forma uniforme entre os componentes. A gestão eficiente da solução adotada para a **escalabilidade** do serviço precisa ser levada em consideração.

Também a solução adotada para atender os requisitos de **tolerância a falhas** influenciam a forma mais eficiente para o balanceamento de carga. Se temos recursos redun-

dantes, por exemplo dois distribuidores que são espelhos um do outro, o atendimento real de um cliente dispense maior esforço de um componente que a **reserva** dos mesmos recursos para uma possível falha. Se um distribuidor com 1GB de memória — analisemos apenas um recurso e num cenário uniforme, para simplificar o raciocínio — pode atender a 100 clientes, a alocação de dois distribuidores iguais, para fins de tolerância a falhas, atende aos mesmos 100 clientes. Porém, é melhor alocar 50 clientes para cada distribuidor, em lugar de alocar os 100 clientes para um distribuidor e deixar o outro ocioso, apenas aguardando uma falha do primeiro.

Isso ocorre pois, dependendo da situação, reservar espaço para atender uma futura falha cria, na verdade, recurso disponível para uso temporário no distribuidor. Com esse espaço extra, o distribuidor pode melhorar o nível de serviço para os clientes em atendimento. Assim, a solução de VoD pode ter uma qualidade de serviço maior que a garantida, nos casos normais. Em caso de falha, têm-se a qualidade garantida, no mínimo.

Para melhorar a eficiência do balanceamento, os diversos parâmetros do serviço de VoD precisam ser equilibrados, uma vez que podem levar a situações conflitantes. Em prol da melhor escalabilidade, pode ser melhor alocar os 100 clientes do exemplo acima num único distribuidor, pois todos assistem ao mesmo ponto do vídeo e o distribuidor pode enviar o conteúdo utilizando multicast. Este cenário reduz a quantidade de fluxos do servidor (no caso, apenas um fluxo). Porém, recuperar uma falha neste ponto seria pior, pois seria necessário migrar os 100 clientes para o distribuidor espelho. Ter 50 clientes alocados em cada máquina, assistindo ao mesmo ponto no vídeo, aumenta o número de fluxos do servidor (dois) e o número de canais de multicast em uso (também dois). Porém, a quantidade de clientes que precisam ser migrados cai pela metade. E como os distribuidores tem mais recursos sobrando, esta migração pode ser melhor preparada.

6.2.4 Mobilidade

Com a popularização das redes sem fio, é cada vez mais comum utilizar esta tecnologia na ligação entre o cliente e a CDN. Com o equipamento cliente conectado através de uma rede sem fio, o usuário tem a possibilidade de se mover ao longo da rede. Esta **mobilidade** dos usuários traz novos desafios para a distribuição de conteúdo de mídia contínua.

O modelo mais utilizado para redes sem fio é o de Pontos de Acesso (AP - *Access Points*) ligados à rede cabeada. A movimentação dos usuários ao longo da rede envolve a **migração** — ou o termo mais usado, *hand-off* — da conexão de um AP para o outro, de acordo com o alcance e a intensidade do sinal de transmissão do AP e do equipamento do

usuário. Usualmente, esta migração de rede é transparente para a aplicação de VoD. Este cenário é descrito na Seção 2.4.

Em relação à garantia de não interrupção, a mobilidade dos usuários pode ser, em princípio, ignorada pela aplicação de VoD, uma vez que camadas inferiores do protocolo de rede tratam a migração do usuário de um AP para outro e a transmissão é mantida independente da aplicação. Alguns ajustes são necessários na aplicação, tais como a preparação para um número maior de perdas de pacote na transmissão. Também a reconexão e/ou alterações de endereço podem ser necessárias para atender à nova configuração da rede sem fio.

6.3 Segurança

Em uma primeira consideração, a transmissão de conteúdo de mídia contínua não inspira requisitos de segurança como os necessários a outras aplicações, como por exemplo transações bancárias. Contudo, alguns elementos tornam a segurança uma condição necessária (ou ao menos desejável) para a transmissão de conteúdo de mídia contínua sob demanda.

Um destes elementos é a garantia do direito de propriedade intelectual. O conteúdo de mídia contínua (filmes, músicas, etc.) é um dos que mais despertam preocupações neste âmbito, pois o valor do conteúdo está no conteúdo em si. Em transações bancárias, o valor não está diretamente na informação que trafega pela rede. Descobrir uma senha de acesso a uma conta bancária abre a possibilidade de se transferir recursos financeiros da conta invadida para outra. Assim, o valor não está na senha em si mas nas operações que ela permite fazer. Interceptar operações bancárias efetuadas por uma grande instituição permite a identificação antecipada de uma tendência de mercado, abrindo possibilidades de ganhos. A perda de que a falta de segurança acarreta pode ser extremamente elevada mas, usualmente, é indireta.

Em uma transmissão de conteúdo de mídia contínua, o conteúdo transmitido é em si o objeto que carrega valor. Capturar uma transmissão de um lançamento de filme simultâneo com o cinema, por exemplo, é suficiente, em um cenário sem segurança, para se obter o filme. Como a cópia de conteúdo digital não afeta a qualidade do conteúdo (original ou cópia), distribuir este conteúdo gratuitamente ou cobrando-se uma quantia menor é simples. Assim, os ganhos gerados pelo conteúdo não são revertidos para os seus donos de direito. E, dependendo do tipo de conteúdo, a própria quebra do sigilo pode

gerar valor futuro, como a descoberta de uma senha bancária.

A segurança da propriedade intelectual é mantida através de mecanismos de DRM (*Digital Rights Management*), que procuram assegurar que apenas os usuários com licença para tal terão acesso ao conteúdo.

A utilização da segurança é um fator que pode contribuir para o bom aproveitamento do serviço de VoD. A garantia de não interrupção prevê algum mecanismo de controle de admissão (Seção 6.2.2). Este controle é responsável pela utilização dos recursos dentro dos limites previstos para o serviço de VoD. Porém, é comum que o dimensionamento dos recursos tenha em vista um conjunto restrito de usuários **autorizados**. Permitir que outros usuários tenham acesso ao serviço pode fazer com que usuários autorizados não consigam utilizar o serviço de VoD.

Desta forma, ter a **autenticação** dos usuários passa a ser importante, bem como um mecanismo de **autorização** de acesso ao serviço. Desta forma, apenas os usuários autorizados poderão utilizar o serviço de VoD.

Capítulo 7

Trabalhos relacionados

7.1 Introdução

A utilização migração de streams não é um tema novo. A transmissão de mídia contínua através de redes sem fio, especialmente relacionadas a telefones celulares, demanda que tratemos o problema do *hand-off*. Ao trocar de célula, em uma rede celular, ou, mais genericamente, ser atendido por um outro ponto de acesso, o usuário precisa reestabelecer a sua conexão na rede. Usualmente, este processo é efetuado automaticamente pelos dispositivos, sem a intervenção do usuário. A retransmissão da mídia para a nova conexão ocorre, então, em nível de rede, sem a intervenção da aplicação. Este procedimento é efetivo, se as condições de transmissão de rede são equivalentes.

7.2 Migração no contexto de rede

Algumas propostas procuram resolver o problema da migração sem a intervenção da aplicação. A vantagem de evitar uma modificação na aplicação usuária pode ser ofuscada. É mais complexo prover uma solução que tenha que servir a diversos tipos de aplicação. Contudo, a utilização desta abordagem pode ser utilizada em redes de celulares, onde não podemos inferir quais as aplicações em uso. Adicionalmente, o próprio mecanismo do *hand-off* em redes celulares já exige a migração da conexão do usuário para uma nova célula.

7.2.1 Branch connections

Em LEE [20] é apresentado uma proposta de utilização de *branch connections* para preparar um rede celular para a migração de um usuário. Em uma rede celular, o usuário pode migrar de uma célula para uma célula vizinha. No momento da troca de célula (*hand-off*),

a infra-estrutura de rede precisa descobrir uma nova rota para o usuário, conectando-o até o ponto de origem da mídia. Para economizar tempo no instante da migração, alocam-se *branch connections* antecipadamente, nas células vizinhas àquela onde o usuário está. Assim, quando a migração ocorre, a rede só precisa **ativar** uma conexão já existente, economizando o tempo de descobrir a rota.

7.3 Migração no contexto de aplicação

7.3.1 Nogueras 2001

Adotando o enfoque em nível de aplicação, temos em NOGUERAS [21] a utilização de migração de streams para o aproveitamento de recursos disponíveis na localização do usuário. Em lugar de tratar especificamente o *hand-off*, a abordagem proposta considera os recursos físicos disponíveis no dispositivo móvel do usuário mais limitados que os disponíveis nos dispositivos localizados por onde o usuário passa. Assim, ao entrar em uma sala com um aparelho de *home-theater*, o som e a imagem migrariam do dispositivo do usuário para a tela e caixas de som do *home-theater*.

Capítulo 8

Conclusões e trabalhos futuros

8.1 Conclusões finais

Utilizar as técnicas de CVC (*Cooperative Video Cache*) e C-CVC (*Collapsed CVC*) melhora a escalabilidade da solução de VoD, conforme demonstrado por ISHIKAWA e AMORIM [11, 7, 3], PINHO [12] e BRAGATO [4]. Nesta dissertação se efetuou a integração das implementações utilizando a CVC (GloVE-LAN [11, 12]) e C-CVC (GloVE-WAN [7, 4]).

Em cenários em que há grande quantidade de usuários utilizando Vídeo sob Demanda em uma rede local, a aplicação da técnica de CVC já se mostrou eficiente, com uma redução significativa da utilização do Servidor. Quanto mais reaproveitamos o fluxo de vídeo oriundo do Servidor, menor a sobrecarga sobre os recursos alocados no Servidor e maior o número de clientes que podem ser atendidos. Em resumo, a **escalabilidade** da solução de VoD aumenta. Por usar técnicas *peer-to-peer*, a CVC se beneficia da agregação de novos recursos que ocorrem com a entrada de novos clientes.

Para sanar a deficiência da CVC para atender clientes em uma rede pública, especialmente com vazão assimétrica (ADSL, por exemplo), a introdução da C-CVC permite se obter os benefícios da CVC em uma única máquina, que passa a agregar os *buffers* dos clientes. Este componente, chamado de Distribuidor no GloVE-WAN, reaproveita os fluxos oriundos do Servidor, em memória, economizando vazão de rede e disco. Porém, a agregação de recursos oferecida por soluções *peer-to-peer* (como a CVC) é perdida. A grosso modo, para os componentes Cliente, o Distribuidor atua como um “Servidor” mais eficiente.

A integração das versões, efetuada nesta dissertação, permite que as técnicas de CVC e C-CVC sejam utilizadas em conjunto, aumentando ainda mais a escalabilidade da solução. O desempenho em cenários em que um grupo de usuários está no mesmo trecho

de rede local, acessíveis uns aos outros, é melhorado uma vez que um fluxo oriundo do Distribuidor — que idealmente já foi reaproveitado para mais de um usuário — é reaproveitado para os usuários do trecho em questão. Assim, os requisitos de vazão de rede entre o Distribuidor e os clientes são reduzidos, assim como a utilização dos recursos no Distribuidor e, conseqüentemente, dos recursos no Servidor.

Os experimentos mostram que a versão integrada tem potencial para redução dos fluxos oriundos de Distribuidor, especialmente nos pontos de pico de demanda. Altas taxas de chegada de clientes, alta concentração de clientes por vídeo e tamanho de *buffer* relativamente grande favorecem a parte CVC da versão integrada, reduzindo a quantidade de fluxos oriundos de Distribuidor. Em resumo, quando o tamanho do *buffer* — em segundos — é maior que o intervalo entre clientes para um mesmo vídeo, a probabilidade de uso da técnica CVC é maior. Caso contrário, um Distribuidor precisa ser alocado.

Entretanto, mesmo nos casos em que a C-CVC precisa ser acionada, a quantidade de fluxos oriundos de Servidor é mantida próximo dos níveis mínimos necessários. Nos experimentos, como todos os vídeos disponíveis são utilizados, esta quantidade mínima é igual à quantidade de vídeos oferecidos durante o experimento. Em situações críticas, taxas extremamente elevadas (60 clientes/min) podem causar que os clientes provedores terminam a exibição do vídeo (e sua propagação a outro cliente) antes do cliente receptor ter o final do vídeo em *buffer*. Nestes casos, o(s) Distribuidor(es) são acionados, causando uma elevação nos fluxos oriundos de Distribuidor. Mesmo nestes casos, a quantidade de fluxos oriundos de Servidor se mantém próxima do mínimo, demonstrando a confiabilidade da solução.

8.2 Trabalhos futuros

A integração entre as versões GloVE-LAN e GloVE-WAN é apenas um passo em direção a uma solução completa de VoD. Algumas ações podem ser tomadas para aumentar ainda mais a qualidade do serviço de VoD que pode ser oferecido pelo GloVE.

8.2.1 Aproveitamento do Grafo de Distribuição

Como todos os componentes do sistema registram sua presença no Supervisor, o Grafo de Distribuição de Vídeo passa a ser conhecido. Este conhecimento pode ser melhor aproveitado para melhorar a eficiência do serviço de VoD.

Utilizar a informação de provimento para vídeos distintos pode auxiliar na eficiência

da distribuição de vídeo. É sabido que as técnicas de CVC e C-CVC funcionam melhor quanto maior a concentração de usuários assistindo ao mesmo vídeo. Assim, pode-se implantar uma política que **concentre** usuários em um mesmo Distribuidor, em lugar de alocá-los de forma dispersa entre os Distribuidores do grafo.

8.2.2 Grupos de Distribuidores

A alocação em grupos pode ser efetuada da seguinte forma. Num primeiro momento, simplesmente seleciona-se o distribuidor que já está provendo o vídeo solicitado. Alocando-se o cliente a este distribuidor, o vídeo que precisa ser recebido do servidor tem maior chance de já estar na memória do distribuidor. Assim, a quantidade de fluxos oriundos dos servidores tende a diminuir, pois o vídeo tenderá a estar no *buffer* do distribuidor.

Não há garantia que o vídeo estará em *buffer*, pois o ponto solicitado pelo novo cliente pode ter sido descartado pelo distribuidor, se os demais clientes atendidos por ele estiverem em um outro ponto distante de exibição. Uma variação da política é identificar o trecho do vídeo que o distribuidor possui, buscando alocar o distribuidor que ainda tem o vídeo em *buffer*. Caso não se encontre o distribuidor que tenha o vídeo, aloca-se aquele que tenha o ponto mais próximo, para permitir que o conteúdo solicitado seja agregado ao que estava disponível. A idéia é evitar descartar vídeo que poderá ser usado pelo novo cliente, se este estiver num ponto anterior ao exibido, ou obter vídeo que será utilizado pelos atuais clientes, se o novo cliente estiver num ponto à frente.

Uma variação mais elaborada permite que se combine tolerância a falhas e balanceamento de carga entre os distribuidores. Em lugar de se concentrar vídeo em um distribuidor, concentra-se o vídeo em um **grupo** de distribuidores. No momento da alocação do novo distribuidor, monta-se um grupo com, digamos, dois distribuidores. A primeira alocação ao distribuidor deste grupo será aleatória. Para o segundo cliente, ao se buscar um distribuidor disponível, detecta-se que um dos elementos do grupo já está provendo o vídeo requisitado. Porém, aloca-se o novo cliente ao distribuidor com menor carga dentro do grupo. Neste exemplo, será o segundo distribuidor do grupo, pois este ainda não está servindo a cliente algum. Uma forma simples de fazer a alocação dentro do grupo é usar *round-robin* entre os distribuidores do grupo. Outra forma é utilizar uma escolha aleatória, mais simples de ser implantada e com uma eficiência a longo prazo tão boa quanto a anterior.

Em resumo, a **política de concentração de vídeo em grupos de N distribuidores** seria algo do tipo:

- Buscar distribuidor com o vídeo solicitado
- Se encontrado
 - Buscar grupo do distribuidor
- senão
 - Criar grupo com N distribuidores (de M distribuidores disponíveis, $1 \leq N \leq M$)
- Escolher um distribuidor do grupo (política aleatória, round-robin, menor carga, etc.)
- Alocar o cliente ao distribuidor escolhido

Na política acima, se $N = 1$, aloca-se o cliente ao distribuidor que já está provendo o vídeo. Se $N = M$, distribui-se a carga do vídeo por todos os distribuidores disponíveis. Se $1 < N < M$, apenas uma parte dos distribuidores estará servindo um determinado vídeo. Neste caso, é importante ter os grupos com distribuidores distintos, de forma que cada distribuidor faça parte de pelo menos um grupo, evitando-se distribuidores ociosos. Também é necessário balancear o número de grupos ao qual cada distribuidor faz parte, de forma a não sobrecarregar algum distribuidor em particular.

Uma forma simples de se escolher os distribuidores, na formação de um grupo, é:

1. Listar os distribuidores em ordem crescente de acordo com o número de grupos a que faz parte (número de vídeos que o distribuidor provê)
2. Alocar ao novo grupo os N primeiros da lista

A mesma sistemática pode ser utilizada para balancear a carga dentro do grupo.

8.2.3 Tolerância a falhas

O tratamento de falhas de provimento é efetuado na versão integrada, mas ainda de forma incipiente. Não há uma gestão pró-ativa da falha, ou seja, apenas quando a interrupção de provimento já ameaça a exibição (ao atingir o índice de *Underflow*) que a troca é efetuada.

Uma alternativa é analisar as condições de recepção e identificar a falha o mais cedo possível. Com esta identificação solicita-se um novo provedor, potencialmente com uma folga maior no *buffer*, permitindo inclusive a recuperação de mais de uma falha.

Esta recuperação antecipada também pode ser auxiliada pelo Supervisor, uma vez que os componentes estão em contato permanente com ele. Ao detectar que um provedor falhou, o Supervisor pode escolher novos provedores para os componentes que recebiam vídeo do provedor em falha. E enviar a cada um destes componentes a informação de um novo provedor. Também pode contatar os novos provedores (assumindo que esta migração possa ser distribuída), permitindo que estes se preparem para o provimento. O caminho para esta implantação estão colocados, uma vez que os canais de comunicação já existem.

8.2.4 Reserva de recursos

A garantia efetiva de Qualidade de Serviço (*QoS - Quality of Service*) passa pela **reserva de recursos** ao longo do Grafo de Distribuição. Esta reserva é necessária pois se o sistema estiver operando no limite quando uma falha ocorrer, não haverá recursos suficientes para recuperação.

Em princípio, a utilização dos Distribuidores em grupos auxilia nesta tarefa. Com a concentração de usuários assistindo a um mesmo vídeo no mesmo Distribuidor, migrar os clientes para outro Distribuidor servindo o mesmo vídeo utilizará (potencialmente) apenas mais vazão de rede no Distribuidor. A probabilidade do vídeo já estar em *buffer* é alta.

A reserva de recursos combinada com a alocação dos distribuidores em grupo pode reduzir o Tempo de Escalonamento do Distribuidor. Atualmente, quando um novo cliente é alocado ao Distribuidor, este precisa carregar uma parte do vídeo em *buffer (prefetch)* antes de iniciar o provimento. Para o início do vídeo, aplicou-se a solução de efetuar esta carga antecipadamente, independente se há cliente solicitando ou não. Mas esta carga precisa ocorrer de qualquer forma, se um cliente solicita um ponto mais à frente, ou se retiramos este *prefetch* inicial para privilegiar a alocação dinâmica de Distribuidores.

Com a reserva de recursos, esta carga inicial seria fruto de uma escolha dirigida, preparando o Distribuidor para receber futuros clientes. A reserva deve ser bem planejada para evitar que o volume de recursos utilizado seja o dobro (ou mais) dos necessários para atender aos clientes previstos. Por exemplo, se a cada novo cliente aloca-se dois provedores, um em reserva e outro efetivamente provendo, a quantidade de clientes que pode ser atendida é 50% da que seria possível sem a reserva de recursos.

Análises estatísticas podem ser utilizadas, como as efetuadas para tratamento de *hand-off* em redes de celulares, para se dimensionar a reserva necessária. Mais uma vez, a utilização das características de concentração de uso do vídeo — a exemplo da proposta

de uso de grupos de Distribuidores — pode contribuir significativamente para a redução dos recursos reservados e não utilizados.

Referências Bibliográficas

- [1] E. Ishikawa and C. L. Amorim, “Memoria Cooperativa Distribuída para Sistemas de VoD peer-to-peer,” in *Proceedings of the Brazilian Symposium on Computer Networks (SBRC)*, (Florianopolis, SC, Brazil), pp. 822–837, May 2001.
- [2] L. B. Pinho, E. Ishikawa, and C. L. Amorim, “GloVE: A Distributed Environment for Low Cost Scalable VoD Systems,” in *Proceedings of the 14th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, (Vitoria, ES, Brazil), pp. 117–124, IEEE Computer Society Press, October 2002.
- [3] E. Ishikawa, *Memória Cooperativa Distribuída para Sistemas de VoD*. PhD thesis, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Outubro 2003.
- [4] L. L. S. Bragato, “Implementação e Avaliação de um Sistema de Vídeo sob Demanda Baseado em Cache Cooperativa Colapsada de Vídeo,” Master’s thesis, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Setembro 2006.
- [5] L. B. Pinho and C. L. Amorim, “Impact of the clients’ position on performance of VoD systems based on collocated APs,” in *Proceedings of 12th Brazilian Symposium on Multimedia and the Web (WebMedia)*, vol. 1, (Natal, RN, Brazil), pp. 253–262, November 2006.
- [6] L. B. Pinho and C. L. Amorim, “Building Scalable Wireless VoD Systems through Efficient Management of Collocated Access Points,” *Journal of Network and Systems Management (JNSM)*, vol. 16, March 2008. ISSN: 1064-7570, Springer (Tentative schedule).
- [7] E. Ishikawa and C. L. Amorim, “Collapsed Cooperative Video Cache for Content Distribution Networks,” in *Proceedings of the Brazilian Symposium on Computer Networks (SBRC)*, (Natal, RN, Brazil), pp. 249–264, May 2003.

- [8] L. B. Pinho and C. L. Amorim, “CINAPS: A Scalable Solution for VoD over WLANs Using Application-Friendly Association Control,” Tech. Rep. ES-681/05, COPPE/UFRJ Systems Engineering Program, May 2005.
- [9] L. B. Pinho, E. Ishikawa, and C. L. Amorim, “GloVE: A Distributed Environment for Low Cost Scalable VoD Systems,” in *Proceedings of the 14th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, (Vitória, ES, Brazil), pp. 117–124, IEEE Computer Society Press, October 2002.
- [10] P. Symes, *Video Compression Demystified*, ch. 1, pp. 8–11. McGraw Hill, 2001.
- [11] E. Ishikawa and C. L. Amorim, “Memória Cooperativa Distribuída para Sistemas de VoD peer-to-peer,” in *Proceedings of the Brazilian Symposium on Computer Networks (SBRC)*, (Florianópolis, SC, Brazil), May 2001. (in Portuguese).
- [12] L. B. Pinho, “Implementação e Avaliação de um Sistema de Vídeo sob Demanda Baseado em Cache de Vídeo Cooperativa,” Master’s thesis, COPPE/UFRJ, Rio de Janeiro, RJ, Brazil, May 2002.
- [13] Seagate Technology LLC., “Cheetah 15K.5 Data Sheet,” 2006. http://www.seagate.com/docs/pdf/datasheet/disc/ds_cheetah_15k_5.pdf. Acessado em 08/09/2007.
- [14] S. Sheu, K. A. Hua, and W. Tavanapong, “Chaining: A Generalized Batching Technique for Video-On-Demand Systems,” in *1997 International Conference on Multimedia Computing and Systems (ICMCS’97)*, pp. 110–117, 1997.
- [15] “MPlayer - The Movie Player.” <http://www.mplayerhq.hu>.
- [16] “Travis-QoS.” <http://www.lcp.coppe.ufrj.br/projetos/Travis-QoS>.
- [17] D. de Souza Diaz Carrion, “Avaliação de Protocolos de Autenticação em Redes Sem Fio,” Master’s thesis, COPPE/UFRJ, Agosto 2005. <http://www.airstrike.ravel.ufrj.br>.
- [18] “MySQL AB.” <http://www.mysql.com>.
- [19] “Laboratório de Computação Paralela - COPPE/UFRJ.” <http://www.lcp.coppe.ufrj.br>.

- [20] K. Lee, "Supporting mobile multimedia in integrated services networks," *Wireless Networks*, vol. 2, no. 3, pp. 205–217, 1996.
- [21] J. R. Nogueras, "A stream redirection architecture for pervasive computing environments," Master's thesis, Massachusetts Institute of Technology, June 2001.
- [22] D. Bulterman, "The SMIL 2.1 Media Object Modules." <http://www.w3.org/TR/2005/PR-SMIL2-20050927/extended-media-object.html#media-Definitions>, September 2005.
- [23] K. Hua, M. Tantaoui, and W. Tavanapong, "Video delivery technologies for large-scale deployment of multimedia applications," in *Proceedings of the IEEE*, pp. 1439–1451, September 2004.
- [24] E. Ishikawa and C. L. Amorim, "Cooperative video caching for interactive and scalable vod systems," in *Networking - ICN 2001, First International Conference, Colmar, France, July 9-13, 2001 Proceedings, Part 2* (P. Lorenz, ed.), vol. 2094 of *Lecture Notes in Computer Science*, pp. 776–785, Springer, 2001. ISBN: 3-540-42303-6.
- [25] L. B. Pinho and C. L. Amorim, "Assessing the efficiency of stream reuse techniques in P2P video-on-demand systems," *Journal of Network Computing Applications (JNCA)*, vol. 29, pp. 25–45, January 2006. ISSN: 1084-8045, available online since December/2004.
- [26] "Laboratório de Redes de Alta Velocidade - COPPE/UFRJ." <http://www.ravel.ufrj.br>.
- [27] "Departamento de Informática - UFES." <http://www.inf.ufes.br>.

Apêndice A

Conceitos básicos

A.1 Introdução

A distribuição em formato digital de conteúdo multimídia está cada vez mais presente em nosso dia-a-dia. Quando este conteúdo possui um tempo mensurável e compreensível, este é chamado de **mídia contínua**[22]. Por exemplo, para um filme, ou mesmo um programa ao vivo de TV, o conceito de tempo decorrido tem sentido. O mesmo não pode ser dito para um texto ou uma foto.

Basicamente, a entrega de conteúdo de mídia contínua se faz de duas formas: Através da **transmissão** — onde o usuário recebe o conteúdo conforme o assiste — e da **distribuição física**, usualmente por CDs ou DVDs — onde o usuário recebe todo o conteúdo antes de utilizá-lo.

A transmissão de conteúdo restringe as opções disponíveis ao usuário, pois é necessária uma **fonte transmissora** para cada conteúdo. Outra limitação é que, geralmente, são utilizados meios compartilhados — ondas eletromagnéticas através do ar, por exemplo — para transmissão. Isso acarreta a limitação dos **canais** disponíveis para distribuição. Por estes motivos é inviável termos interatividade¹ em transmissão de conteúdo, para uma grande quantidade de usuários.

A **transmissão por difusão** (ou *broadcast*) é a forma empregada pelas emissoras de TVs e estações de rádio — abertas, por assinatura ou comunitárias — para entregar conteúdo a seus clientes, que escolhem o que assistir selecionando entre diversos **canais** ou **estações** disponíveis. A transmissão é enviada para todos.

Para que somente os cliente habilitados tenham acesso à programação, são utilizados mecanismos de proteção. O conteúdo é transmitido de forma codificada e somente os receptores habilitados pela operadora conseguem decodificar. As emissoras de TV

¹Quando usamos somente transmissão

aberta transmitem sua programação sem codificação especial, podendo ser assistidas em qualquer aparelho de TV.

A distribuição física, através de CDs ou DVDs, é utilizada quando o usuário requer mais liberdade de opções que uma seleção de canais. Nesta **utilização individual**, além de poder escolher **quando** assistir, o cliente pode interferir na exibição, interrompendo, saltando trechos, exibindo em velocidade maior/menor e etc. Esta interação não é possível com uma transmissão por difusão convencional. Contudo, ao combinarmos as duas técnicas, guardando o conteúdo transmitido em algum dispositivo de armazenamento (fita, disco, memória), transformamos transmissão em distribuição física, permitindo a interação.

Apesar de oferecer liberdade e interatividade maiores que a transmissão por difusão, a distribuição física de conteúdo apresenta algumas desvantagens. O cliente não pode usufruir do conteúdo imediatamente. Ao decidir que deseja assistir a um filme ou ouvir uma música, o usuário precisa ir até à locadora/loja para alugar/comprar o DVD/CD em questão².

Este problema é minimizado com o uso de entregas em domicílio, que permitem ao cliente solicitar o conteúdo por carta, telefone ou Internet. Ainda que as empresas se esforcem continuamente para reduzir os prazos de entrega destes serviços, estes podem usualmente ser medidos em horas (para entregas locais, em serviços de aluguel), dias (para entregas nacionais ou expressas internacionais) ou semanas (para entregas internacionais). A **latência** — aqui considerada como o tempo decorrido entre a solicitação do usuário e o início da exibição — envolvida neste processo pode ser muito grande.

A.1.1 Redes de computadores para entrega de mídia contínua

A presença crescente do acesso em banda larga à Internet, especialmente nos grandes centros, abre campo para uma nova forma de distribuição de mídia contínua. Seguindo os mesmos princípios da **distribuição física**, o usuário pode **descarregar** o conteúdo desejado para seu computador pessoal (ou outro dispositivo, como um *iPod* ou *set-top box*), permitindo interatividade ilimitada. Isto reduz a latência na entrega do conteúdo, mesmo para entregas internacionais. Porém, os tamanhos dos arquivos de mídia contínua digital, em particular vídeo, são consideráveis. Mesmo tendo uma vazão de rede média disponível de 4 Mbps, um arquivo típico, com 2 GB para 2 horas de vídeo, levaria em média 70 minutos para ser recebido.

²Há, é claro, a possibilidade do usuário já possuir o disco. Neste caso, a espera é praticamente nula.

Uma solução comum é o aumento na **taxa de compressão** do vídeo. Como os algoritmos de compressão mais eficientes para vídeo são com perdas — retiram informações menos relevantes, ou redundantes, do vídeo — aumentar a compressão implica em uma redução gradual na qualidade do vídeo. Este procedimento pode representar uma grande economia de tempo. Mantendo a vazão de rede disponível de 4 Mbps, mas aumentando a compressão, podemos reduzir a 4 minutos o tempo médio para se obter um arquivo de vídeo. Contudo, a diferença qualitativa é perceptível, pois é necessária uma alta taxa de compressão (mais de 94%) para reduzir um vídeo de 2 GB com 2 horas de vídeo, ou seja codificado a uma taxa média de 2 Mbps, a um vídeo de 109 MB, codificado a uma taxa média de 128 Kbps.

Ainda assim, podemos ter um tempo de espera maior do que este. Afinal, estamos considerando que a **taxa nominal** informada pelo provedor de serviço é sustentada por todo o tempo de transferência do conteúdo. Se a localização original do vídeo estiver geograficamente distante (outra cidade, por exemplo) ou, o que é mais comum, ocorrer congestionamento nas linhas de transmissão, esta premissa pode não ser verdadeira.

Se o tempo de espera for previsível, o cliente pode se programar, solicitando o conteúdo antecipadamente. Afinal, utilizar um meio alternativo, como a ida a uma locadora, poderia ter uma espera igual ou maior. Adicionalmente, os acréscimos regulares na largura de banda disponível podem reduzir este tempo de forma significativa. Uma vazão média de rede de 8 Mbps reduziria o tempo, no exemplo anterior, de 70 para 35 minutos.

Podemos reduzir a espera inicial sensivelmente se, em lugar de aguardarmos o fim da transferência, exibirmos o conteúdo **enquanto** o vídeo está sendo transmitido. O problema com esta estratégia — também conhecido como *streaming* — é que precisamos garantir que a taxa com que recebemos o conteúdo seja maior ou igual à taxa em que este é exibido. Em certos casos, vídeos distribuídos em DVD, utilizando o padrão MPEG-2 (o que é usual), apresentam picos de 8 Mbps de taxa de exibição.

Apesar da espera ser desconfortável para o usuário, este não é o maior problema de uma **transmissão individualizada**. O maior obstáculo a esta técnica é a falta de **escalabilidade**. A **vazão de rede** necessária em um servidor que transmita individualmente a cada usuário é $O(n)$, onde n é a quantidade de usuários. Assim, na tabela A.1 temos a progressão da vazão de rede necessária para se oferecer transmissão individualizada de vídeos codificados à taxa média de 2 Mbps.

Vale lembrar que esta taxa de transferência não se sustenta na infra-estrutura de Inter-

Tabela A.1: Progressão em $O(n)$ da vazão necessária

Usuários	Vazão
1	2 Mbps
10	20 Mbps
100	200 Mbps
1000	2 Gbps

net disponível atualmente. Apenas com um servidor diretamente ligado ao usuário através de uma linha dedicada, isso seria teoricamente possível. Na prática, há outros fatores envolvidos, tais como taxa de transferência de disco, ocupação do servidor, entre outros, que tornam a vazão efetiva recebida pelo usuário menor que o limite nominal declarado de sua linha de recepção. Usualmente, a soma das larguras de banda disponíveis aos clientes é maior que a largura de banda entre outros nós ao longo da rede de distribuição.

A.1.2 Utilização de redes sem fio

Outro fator relevante no contexto de redes de computadores é a utilização de tecnologias sem fio. Seja para permitir a mobilidade dos usuários, para simplificar a instalação da rede (com a eliminação dos cabos) ou com motivação estética (menos cabos aparentes), a utilização de redes sem fio tem se tornado cada vez mais presente.

Meios sem fio para distribuição de conteúdo de mídia contínua são utilizados desde a aparição das emissoras de rádio e TV. Porém, utilizar tecnologia sem fio em redes de computadores, especialmente se considerarmos a questão da mobilidade, é mais próximo da tecnologia de **telefones celulares**. Na transmissão de rádio e TV o método mais utilizado é a **difusão**, com frequências (canais) diferentes sendo utilizados por emissoras diferentes. Em telefones celulares, a utilização é **individual**, utilizando, potencialmente, uma quantidade muito maior de frequências. Estas são recursos compartilhados (e limitados), sendo fruto de regulamentação governamental quanto à concessão de faixas de frequência para certos serviços e, dentro destes, de determinadas frequências para cada emissora/operadora.

Na telefonia celular, de forma simplificada, reduz-se a quantidade de frequências distintas necessárias com o reaproveitamento destas. A região coberta pelo serviço é dividida em **células** — daí o nome de telefonia celular — com um conjunto de frequências sendo utilizado em cada célula. Evita-se utilizar uma mesma frequência em células vizinhas, para reduzir as interferências. Contudo, em células não contíguas, uma mesma frequên-

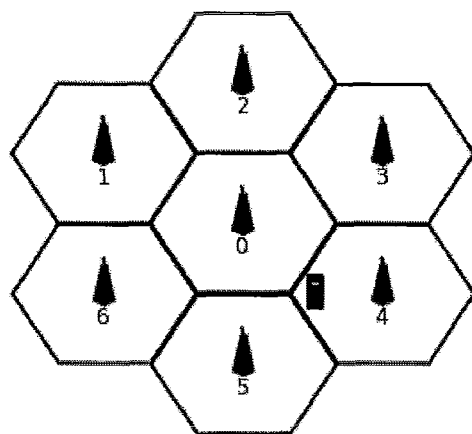


Figura A.1: Rede celular - O aparelho recebe sinal das estações base 0, 4 e 5. Mas, como o sinal da estação 4 é mais forte, ele se associa à célula 4.

cia pode ser reutilizada por um outro aparelho. Cada célula é composta por uma **estação base**, que é o ponto de entrada na rede. Usualmente há uma estrutura de rede cabeada até a estação base. Os aparelhos dos usuários fazem busca, via rádio-frequência, da estação base com sinal mais forte, para que possam identificar a célula a que pertencem. Ao se conectar com a estação base, o terminal passa a fazer parte da rede. Se a estação está congestionada, com todas as frequências disponíveis ocupadas, a conexão do usuário é recusada. O alcance da estação base determinará a quantidade de células necessárias, bem como a taxa de reaproveitamento das frequências. Quando maior o alcance, menos células e menos reaproveitamento.

Na figura A.1, temos um exemplo de uma rede celular. A divisão em células hexagonais aumenta a reutilização das frequências nas células vizinhas. O sinal de uma célula é recebido nas fronteiras das células contíguas. Contudo, o terminal pode medir a intensidade do sinal para descobrir a que célula deve se associar. As células 4 e 1, por exemplo, podem utilizar o mesmo conjunto de canais. Basicamente, a escolha de frequências se reduz a um problema de coloração em grafos.

Em redes sem fio de computadores, os conceitos são parecidos. Os **pontos de acesso** (ou *AP - Access Points*) fazem o papel das estações base. Ao se associar a um AP, é designado um endereço de rede ao usuário. Este endereço pode ser sempre o mesmo, como em um celular, onde o número não é alterado com a mudança de célula. Ou pode ser diferente, refletindo a troca de seção de rede em que o usuário se encontra.

Transmissões utilizando redes sem fio sofrem com uma quantidade maior de interferência. Obstáculos presentes no ambiente podem comprometer ou até impedir uma

transmissão. Outros dispositivos sem fio, ou mesmo aparelhos não relacionados, como um forno de microondas, podem interferir na qualidade da conexão. Adicionalmente, a quantidade de frequências disponíveis é limitada, influenciando a quantidade de aparelhos que podem estar ativos em uma célula.

Enquanto em uma rede cabeada a quantidade máxima de usuários conectados diretamente é determinística, em uma rede sem fio isto é influenciado pelo padrão de mobilidade dos usuários. Uma rede doméstica interligando computadores, móveis ou não, e outros dispositivos como *iPods* e *handhelds*, tem quantidade de componentes quase constante. O mesmo pode ser dito de uma rede departamental, em que o cabeamento foi substituído por acesso sem fio.

Porém, ao considerarmos redes maiores, a mobilidade dos usuários pode forçar um segmento da rede a atingir o seu limite de capacidade. Uma rede, projetada para ter 100 usuários divididos em 10 segmentos de 10 usuários cada, pode se tornar inoperante se todos os usuários de um segmento migrarem para o segmento vizinho. Nestes casos é usual que um usuário que já esteja sendo servido pela rede tenha prioridade sobre outro iniciando uma conexão. Como exemplo, em telefonia celular é preferível ter uma ligação nova recusada a ter uma ligação em curso interrompida.

A.2 Mídia contínua sob demanda

Quando combinamos a liberdade de escolha oferecida pela **distribuição física em discos** — ou através da obtenção de arquivos com o conteúdo — com a baixa **latência** envolvida na **transmissão por difusão**, temos a chamada **mídia contínua sob demanda**. O usuário pode escolher o que deseja assistir/ouvir e a espera até o início da exibição é pequena (o mais próximo possível de zero). A interatividade não é obrigatória, mas tê-la pode ser um fator determinante para a satisfação do usuário. Em HUA et al [23] são referenciadas várias técnicas para distribuição de mídia contínua sob demanda, especialmente tratando de aspectos de escalabilidade.

Para sedimentarmos os problemas e soluções básicos envolvidos em entrega de mídia contínua, analisaremos um cenário progressivo de entrega de vídeo³. É importante ressaltar que, em todos os casos, o vídeo será entregue por demanda, conforme a necessidade da exibição.

³As considerações se aplicam a mídias contínuas em geral. Mas, por simplificação, utilizaremos o termo “vídeo” em lugar de “mídias contínuas”

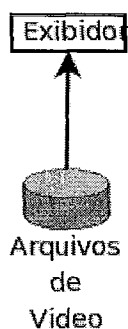


Figura A.2: Um usuário isolado

A.2.1 Cenário inicial

Primeiramente, consideremos um usuário com uma máquina isolada contendo um conjunto de arquivos de vídeo. A figura A.2 mostra o caso geral de exibição. O vídeo é carregado de um meio de armazenamento qualquer (DVD, CD, disco rígido) e é entregue ao exibidor (ou *player*⁴) adequado.

Ainda que o cenário seja extremamente simples, algumas considerações podem ser feitas. Assumindo que o vídeo é carregado do meio de armazenamento **sob demanda**, a taxa de transferência disponível deste meio para o exibidor precisa ser maior ou igual à taxa de exibição do vídeo. Caso contrário, o exibidor consumirá o vídeo mais rápido que o meio é capaz de prover. Se considerarmos um CD-ROM 1x, a taxa máxima do vídeo poderá ser 1,2 Mbps⁵. Para um vídeo de qualidade de DVD (MPEG-2), esta taxa é insuficiente. Nestes casos o vídeo pode chegar a picos de 8 Mbps.

Para contornar este problema, podemos utilizar um *buffer* no exibidor (ou antes dele). Assim, o vídeo seria carregado inicialmente para o *buffer* e deste para o exibidor. Mesmo assim, esta solução é limitada. Afinal, se a taxa média do vídeo é maior que a taxa média do meio de armazenamento, o *buffer* apenas adia o problema. A utilização do *buffer* é eficaz quando a taxa média do meio e do exibidor são equivalentes. Neste caso, o *buffer* atua como uma garantia contra as variações na taxa de consumo do exibidor ou da taxa de transferência do meio de armazenamento.

⁴Os detalhes necessários à decodificação do vídeo e sua exibição estão fora do escopo deste documento

⁵150 KB/s=1,2 Mbps

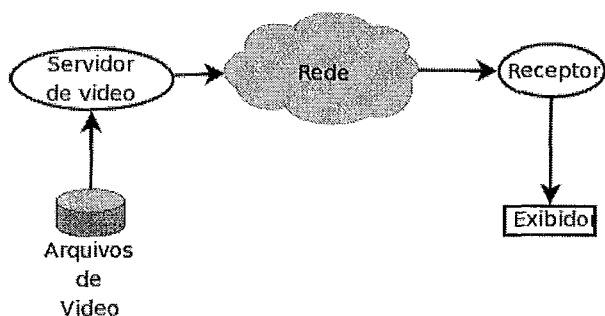


Figura A.3: Provendo o vídeo através da rede

A.2.2 Provendo vídeo através da rede

O repositório com os arquivos de vídeo pode ser compartilhado por mais de um usuário. Desta forma, podemos deslocá-lo para um servidor em uma rede local. Assim, a figura A.3 mostra o provimento de vídeo através de uma rede.

Neste caso, além da taxa de transferência disponível pelo meio de armazenamento, temos o limitador da vazão da rede. Em geral, esta é menor que a taxa provida pelo meio de armazenamento, tornando-se o gargalo no sistema. Adicionalmente, ao utilizarmos uma rede com **roteamento de pacotes**, podemos ter pacotes de dados chegando ao receptor fora de ordem. É responsabilidade do Receptor, ou do protocolo de transporte de rede, reordená-los para que possam ser utilizados pelo Exibidor. É comum utilizarmos um protocolo de rede que permita perdas e entrega fora de ordem (UDP, por exemplo). Isso ocorre pois os protocolos com entrega garantida e em ordem (TCP, por exemplo) adicionam penalidades em termos de vazão de rede. Sendo essa um recurso escasso, todo custo adicional deve ser suprimido.

O uso de *buffers* neste cenário é amplamente disseminado, uma vez que as características da rede não permitiriam que o fluxo recebido pelo Receptor fosse entregue diretamente ao Exibidor. Isso faria com que o Exibidor precisasse “pausar” a exibição periodicamente, quando a taxa de exibição fosse maior que a taxa de entrega. Essa situação seria desconfortável para o usuário.

A figura A.4 mostra o servidor provendo vídeo para vários usuários. Para simplificar a figura, a rede não é representada. Repare que, neste cenário, o provimento segue a estrutura de uma **árvore**, com o servidor como raiz e os clientes como folhas.

Esta solução sofre no quesito escalabilidade. Considerando que todos os clientes estão utilizando vídeos distintos, e sendo n o número de clientes, a taxa de transferência exigida

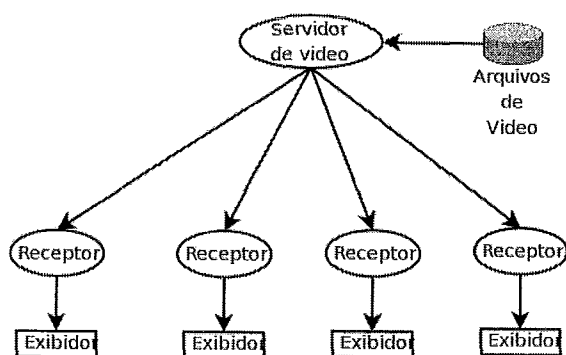


Figura A.4: Provendo o vídeo para vários usuários através da rede

do meio de armazenamento é $O(n)$, assim como a vazão de rede necessária à transmissão de vídeo. Logo, a uma taxa de 2 Mbps de vídeo, e 4 clientes, é necessária uma taxa de transferência de 8 Mbps do meio de armazenamento e na rede de transmissão. Sendo o meio de armazenamento — ou **fonte primária** — um disco rígido e a rede uma Gigabit Ethernet, esta demanda é atendida sem maiores dificuldades. Entretanto, assumindo um disco rígido *serial-ata* disponível em 2005, podemos considerar uma taxa média de 440 Mbps. Isto permitiria um máximo teórico de 220 usuários simultâneos, mesmo havendo sobra de vazão disponível na rede. Se, por outro lado, considerarmos que alguns clientes estão utilizando o mesmo vídeo, o servidor pode ter uma *cache* em memória dos blocos de vídeo. Assim, a taxa que o meio de armazenamento precisa prover é reduzida.

Quando a taxa de transferência disponível na rede é maior que a taxa oferecida pelo meio de armazenamento, podemos adotar algumas soluções. Com a utilização de mais de um servidor de vídeo, conforme visto na figura A.5, podemos ter várias árvores de provimento utilizando a mesma infra-estrutura de rede. Alternativamente, podemos aumentar a quantidade de discos/controladoras no servidor, aumentando a vazão disponível para a fonte primária. Conceitualmente, as soluções são equivalentes. A estrutura de provimento ainda é uma árvore. Só que podemos considerar como uma floresta (vários servidores), ou que a taxa de transferência oferecida pelo meio de armazenamenro (discos) é maior.

A.2.3 Colocando distribuidores próximos aos usuários

Considerando redes de larga escala, a distância entre os usuários pode inviabilizar a distribuição. Em redes públicas, a oferta de banda na rede é menor que em uma rede local. É comum encontrarmos oferta de 512 Kpbs ou 1 Mpbs de largura de banda disponível para residências. Esta taxa só é sustentada se temos o servidor de vídeo próximo à co-

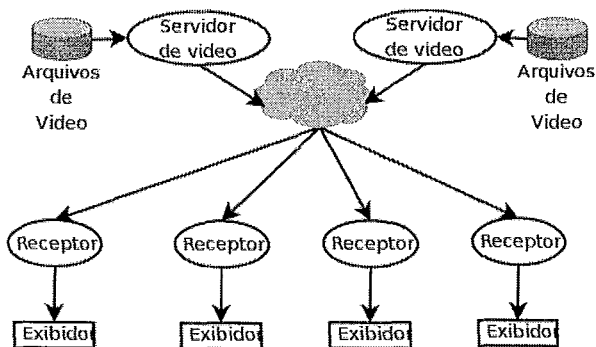


Figura A.5: Provendo o vídeo para vários usuários com mais de um servidor

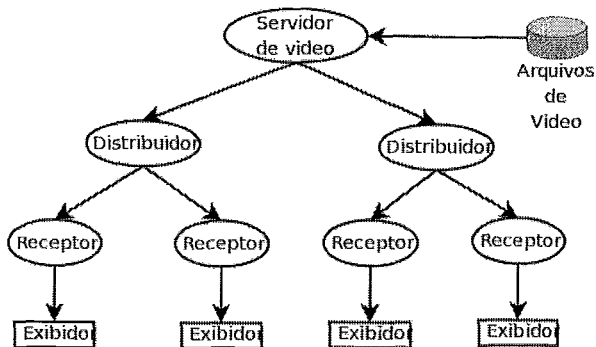


Figura A.6: Distribuidores para auxiliar no provimento

nexão que segue diretamente até o usuário. Isso nos limitaria a ter um repositório de vídeo independente para cada central de ligação com os clientes.

Para evitar esta necessidade, nós adicionamos um nível intermediário à estrutura de provimento. Conforme visto na figura A.6, os **distribuidores** atuam como *proxies dinâmicos* [23, 7], ou seja, comportam-se como clientes, em relação ao servidor, e como servidores, em relação aos clientes. Os distribuidores armazenam partes do vídeo próximo aos clientes, reduzindo a necessidade de vazão do servidor.

Esta solução difere de termos vários servidores diretamente ligados aos usuários. Neste caso, deveríamos administrar a base de vídeo em todos os servidores, aumentando o custo de manutenção da solução. Com os distribuidores, temos um “servidor” próximo dos clientes e um cliente que se antecipa às solicitações (ou armazena de um cliente para outro, como uma *cache*). Esta é a função de um *proxy*. Adicionalmente, por serem *proxies dinâmicos*, o conteúdo armazenado ou solicitado é obtido sob demanda, conforme a solicitação (ou a antecipação desta) dos clientes ao distribuidor.

A utilização de um *proxy dinâmico* tem um custo menor que a de um *proxy estático*

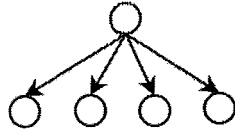


Figura A.7: Árvore simples

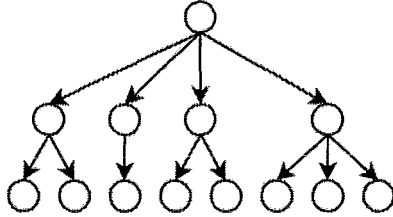


Figura A.8: Árvore

pois não é necessário replicar o repositório de vídeo, total ou parcialmente, nos *proxies*. A obtenção do vídeo é automática, reduzindo os custos de manutenção da solução. Contudo, precisamos lidar com a latência envolvida na obtenção do vídeo, caso este ainda não esteja no âmbito do *proxy* dinâmico.

Em particular, o projeto GloVE [24, 7, 9] propõe uma forma eficiente de transmissão de vídeo sob demanda. Para evitar que as deficiências no meio de transmissão afetem a exibição, é utilizado um *buffer* para armazenar uma parte do vídeo. A versão *peer-to-peer* [25] do GloVE prevê a utilização destes espaços no cliente para efetuarmos o provimento de vídeo, entre clientes. A versão rede aberta [7] utiliza o conceito de *proxies dinâmicos*, com a utilização unicamente de *buffers* em memória. Desta forma, temos equipamentos próximos à conexão do cliente, nos permitindo utilizar, de forma mais efetiva, a vazão disponível individualmente a cada um deles.

A.3 Provimento de vídeo como um grafo

O provimento de vídeo pode ser visto como um grafo. Na sua forma mais simples, temos uma árvore com raiz no servidor e os clientes como folhas (Figura A.7). Adicionando os distribuidores, a estrutura ainda é a de uma árvore com raiz no servidor (Figura A.8).

Contudo, a estrutura de provimento pode ser mais complexa do que esta. Se considerarmos as seguintes possibilidades:

- Mais de um fornecedor de vídeo para cada receptor

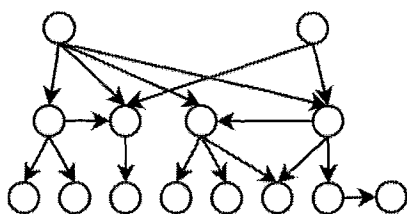


Figura A.9: Um grafo complexo de provimento

- O fornecedor pode estar no mesmo nível da árvore que o receptor

Temos o grafo da figura A.9. Mesmo neste grafo, algumas restrições precisam ser atendidas. Em primeiro lugar, o grafo dirigido é acíclico, evitando que um provedor de vídeo receba, direta ou indiretamente, de si mesmo. Do ponto de vista de um cliente, todos os caminhos dirigidos a ele se originam, em última instância, em uma fonte primária (p.e. disco). Em segundo lugar, a soma da vazão utilizada pelas arestas de um nó não pode exceder a largura de banda total disponível para este nó.

Se os clientes não fazem provimento de vídeo, e os servidores são sempre raiz de um caminho dirigido, temos um caso especial em que os clientes são *sinks* — sendo apenas destino de arestas dirigidas — e os servidores são **fontes** — sendo apenas origens de arestas dirigidas. Se considerarmos o grafo de um único vídeo, em um dado instante, precisamos ter **sempre** ao menos um *sink* e ao menos uma fonte. Caso contrário, teríamos um ciclo.

Cada vídeo tem o seu grafo de distribuição independente. Em uma implementação *peer-to-peer*, os vídeos poderiam estar distribuídos nos clientes e estes seriam as fontes primárias para o mesmo. Ainda assim, poderíamos utilizar a estrutura de distribuição em árvore. Quando o cliente é fonte primária de um vídeo, participa como raiz da árvore de distribuição.

Utilizando a estrutura de distribuição em grafo, possibilitamos que partes do vídeo sejam providas por nós distintos. Esta abordagem permite **somar as vazões disponíveis**. Na figura A.10 vemos um cenário em que o distribuidor recebe parte do vídeo de outro distribuidor e parte do servidor. No caso em que a vazão servidor-distribuidor é menor que a distribuidor-distribuidor, esta utilização pode agilizar a entrada de um novo componente no grafo de provimento.

Outra situação semelhante seria ter um distribuidor que não recebesse vídeo do servidor, mas somente de outro distribuidor. Isto ocorreria quando os clientes do primeiro distribuidor estivessem assistindo ao mesmo vídeo que os clientes do segundo distribui-

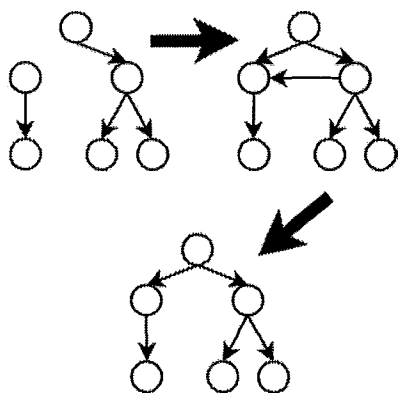


Figura A.10: Somando as vazões disponíveis

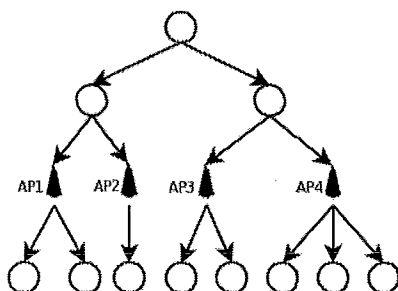


Figura A.11: Grafo de distribuição sem fio, considerando os APs

dor. Contudo, o segundo distribuidor já não possui os primeiros blocos do vídeo. O primeiro distribuidor iria receber os primeiros blocos do servidor e os demais blocos do segundo distribuidor. Após este processo inicial de sincronização, a banda disponível para o servidor poderia ser melhor utilizada, com o servidor provendo um distribuidor e este repassando para os demais distribuidores provendo o mesmo vídeo, em sua rede.

A.3.1 Grafo de distribuição em redes sem fio

Ao considerarmos a utilização de redes sem fio na distribuição de mídia contínua, é importante considerar se mudanças de ponto de acesso irão influenciar a transmissão ou não. No caso geral, um distribuidor pode estar servindo clientes que se associem à rede por mais de um ponto de acesso. Na figura A.11 é mostrado um cenário como este. Se a mudança de AP não reflete alterações para a distribuição do conteúdo, o grafo a ser considerado pela aplicação é o da figura A.12. Caso contrário, os APs serão realmente considerados nós no grafo.

Um caso específico é onde os distribuidores estão servindo a um único ponto de

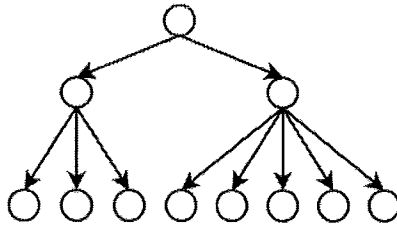


Figura A.12: Grafo de distribuição sem fio, sem APs

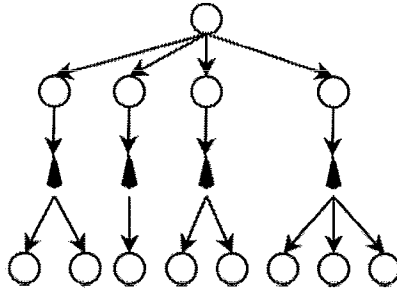


Figura A.13: Grafo de distribuição sem fio, com os distribuidores nas estações base

acesso. Em uma rede de celulares, colocar o distribuidor no mesmo local que o AP pode ser uma forma de maximizar a vazão de rede oferecida na transmissão de conteúdo. Assim, os distribuidores estariam co-localizados com os “pontos de acesso”, que seriam as estações base. A figura A.13 retrata este cenário. Também neste caso, os APs poderiam ser desconsiderados e o grafo seria equivalente ao da figura A.8.

A.3.2 Componentes da distribuição de mídia contínua

A distribuição de mídia contínua, quando observada como um grafo, é uma sucessão de nós repassando conteúdo adiante. Sob o ponto de vista de um único cliente, se o grafo for, no caso mais simples, uma árvore, a cadeia de distribuição do vídeo é um grafo linear, conforme visto na figura A.14. A raiz é a fonte primária, passando pelo(s) distribuidor(es) e chegando até o cliente (folha).

Alguns componentes podem ser identificados nesta transmissão:

- **Leitor de vídeo em disco** - Obtém os blocos de vídeo necessários em disco. Este componente é, basicamente, utilizado no servidor de vídeo (fonte primária);
- **Transmissor de blocos** - Envia blocos a outros nós do sistema de distribuição. Pode ser utilizado pelo servidor de vídeo ou pelo distribuidor. Se um cliente provê vídeo, utiliza este componente para enviá-lo;

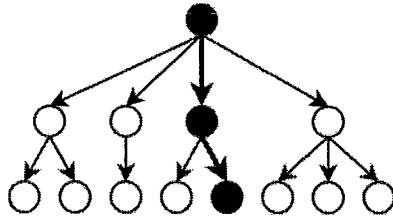


Figura A.14: Distribuição como grafo linear

- **Receptor de blocos** - Recebe blocos de um outro nó do grafo de distribuição. Utilizado por nós que não são fonte primária do vídeo;
- **Escalonador de blocos** - Escalona os blocos para envio. Há basicamente duas formas de servirmos blocos na rede. No modelo *push*, o requisitante informa que quer o vídeo, e o servidor envia os blocos automaticamente, sem necessidade de novas solicitações. No seu complementar, o modelo *pull*, o requisitante solicita um ou mais blocos e este pedido é atendido pelo servidor. Ao utilizarmos o modelo *push*, o escalonador de blocos é o responsável por adequar o envio dos blocos às taxas necessárias pelo cliente e disponíveis no servidor. Quando adotando o modelo *push*, provedores de vídeo (servidores, distribuidores ou clientes) são nós que o utilizam;
- **Exibidor (*player*)** - Decodifica os blocos recebidos e os apresenta ao usuário. Utilizado primordialmente no cliente;
- **Buffer de retenção de blocos** - Armazena blocos recebidos/lidos para posterior utilização. Utilizado no cliente, para compensar variações de taxa na recepção/exibição, e no distribuidor para reter os blocos recebidos, agilizando o envio aos clientes.

A.3.3 Cadeia de distribuição

Se a estrutura de distribuição, do ponto de vista do cliente, é um grafo linear, a latência de recepção do vídeo aumenta, inicialmente, pois os **buffers de retenção de blocos** precisam ser preenchidos antes do envio do vídeo.

Para contornar este problema podemos, nos nós intermediários, efetuar uma carga inicial de blocos, considerando os vídeos conhecidos. Desta forma, não só evitamos o aumento da latência causado pela carga inicial, mas também reduzimos a latência total.

Afinal, a solicitação não precisa trafegar até a fonte primária para ser atendida. O nó intermediário que possuir o *buffer* com os blocos corretos já pode iniciar o atendimento. Porém, se a lista de vídeos é muito grande, esta opção pode ser inviável, dados os recursos consumidos pelo *buffer*.

Alternativamente, ou de forma complementar, podemos efetuar a transmissão do vídeo, nos nós intermediários, imediatamente após a recepção de cada bloco. Assim, em lugar de aguardarmos que o *buffer* dos nós intermediários seja preenchido, podemos efetuar as duas ações **simultaneamente**. A latência ainda é maior que quando o *buffer* já está preenchido, mas é menor que a gerada pela espera do *buffer* completo.

A aplicação desta alternativa pode ser uma solução de compromisso entre latência no cliente e consumo de recursos com *buffer* nos nós intermediários. Podemos ter algum espaço reservado para vídeos sabidamente mais requisitados, efetuando a carga antecipadamente. Outra porção seria liberada para vídeos menos solicitados. Estes seriam carregados utilizando a carga/envio simultâneos.

Outra forma de adotar esta solução de compromisso é considerar o *buffer* nos nós intermediários como uma *cache*. Podemos considerar todo o *buffer* como liberado para ser preenchido de acordo com as solicitações dos clientes. Contudo, para os vídeos que supomos mais requisitados, efetuamos uma carga inicial. Conforme as requisições são atendidas, se esta suposição for verdadeira, os conteúdos mais requisitados sempre estarão em *cache*. Se a suposição não for acurada, o próprio funcionamento da *cache* fará com que os blocos corretos ocupem seus lugares no *buffer*.

A.4 Qualidade de Serviço

Quando o grafo linear de transmissão é estabelecido, o envio de vídeo pode consumir uma taxa determinística. Afinal, após a latência inicial, o vídeo é lido, enviado, recebido e exibido na taxa requisitada pela exibição. A abstração envolvida aqui é a de que o exibidor do arquivo o está lendo em seu disco local. Esta abstração é que faz a transmissão de mídia contínua ser considerada uma aplicação de **tempo real**. Um atraso não planejado na transmissão pode comprometer a exibição de vídeo para o cliente.

Isso suscita o problema do que fazer se uma das conexões for interrompida. Se o cliente perder a conexão com o distribuidor, por exemplo, o exibidor ainda pode passar vídeo sem que o usuário perceba esta perda. Basta consumir o que está em seu *buffer* local. Isto permite que o cliente reconecte com o distribuidor, retomando a recepção do

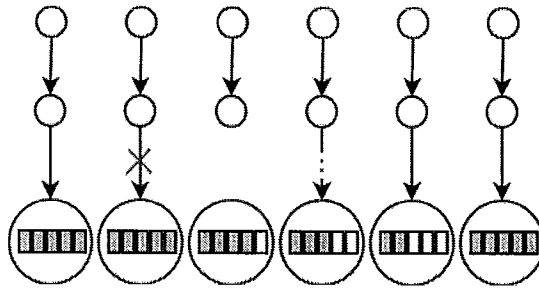


Figura A.15: Utilização do *buffer* em uma perda de conexão. Apesar do que está apresentado na figura, caso a taxa de recepção se mantenha exatamente igual à taxa média de exibição, o *buffer* pode não retornar ao estado de total preenchimento. Uma vez que as taxas seriam iguais, a taxa de preenchimento do *buffer* tenderia a se manter estável.

vídeo (Figura A.15).

O fato do cliente não perceber a perda de conexão pode ser entendida como um nível de **qualidade de serviço** oferecido pela solução. Se a suspensão fosse tal que interrompesse a exibição, o cliente teria uma percepção diferente — subjetivamente pior — da transmissão de vídeo.

Em um cenário simples, de transmissão de conteúdo em rede local, a **garantia de qualidade de serviço** pode ser simplesmente certificar ao usuário que o vídeo será transmitido até o final, sem interrupções. Este **QoS** — qualidade de serviço ou *Quality of Service*, na sigla em inglês — simples requer algumas preocupações na solução. Em primeiro lugar, é necessário que tenhamos um **controle de admissão**, com **reserva de recursos**. Afinal, precisamos garantir que haverá vazão suficiente no servidor, tanto em disco como em rede, para prover o conteúdo ao cliente. Se não efetuarmos o controle de admissão, é possível ter mais usuários que o máximo suportado pelo sistema. Com isso não seria viável garantir qualidade de serviço a nenhum deles pois, de forma não determinística, a solução estaria sujeita a congestionamentos de rede e atrasos no envio dos blocos, entre outros.

Com a adição dos distribuidores (ou outros níveis intermediários extras), o mesmo problema se apresenta. O controle de admissão é indispensável e, conseqüentemente, precisamos reservar recursos em todos os nós intermediários envolvidos. Contudo, os distribuidores podem — e devem — reduzir a demanda nos servidores, uma vez que adicionam recursos ao sistema. Como, idealmente, a parte necessária do vídeo está disponível no distribuidor para entrega a mais de um cliente, a demanda por vazão (de rede e disco) nos servidores é reduzida.

A.4.1 Outros fatores que influenciam QoS

Além dos recursos consumidos para o atendimento aos clientes, outros fatores influenciam a qualidade de serviço. Se um dos nós na cadeia de distribuição perde conectividade com o nó vizinho, a transmissão fica comprometida. Uma solução simples é restabelecer a conexão perdida, retomando a transmissão. Contudo, a falha pode ter sido na infra-estrutura de rede, não permitindo uma recuperação pela aplicação, utilizando o mesmo canal de comunicação. Outro fator que influencia no QoS é a ocorrência de falhas, de hardware ou software, nos nós da cadeia de transmissão. Uma falha no servidor, por exemplo, comprometerá toda a árvore de distribuição, pois o servidor é a raiz da mesma. Se a falha for irreversível, por um defeito em disco ou travamento no sistema operacional, o sistema inteiro será afetado.

Questões de segurança de rede também influenciam a qualidade de serviço oferecida pelo sistema. O controle de admissão é responsável pelo controle da quantidade de usuários atendidas pelo sistema. Porém, precisamos garantir que estes usuários estão **autorizados** a utilizá-lo. Prover uma **autenticação** é fundamental. Assim, evitamos aceitar usuários não autorizados, que consumiriam recursos no sistema de forma não planejada. Nos casos em que o sistema é público, disponível a qualquer usuário que queira utilizá-lo — como uma rede de TV aberta, por exemplo — podemos considerar a autenticação/autorização como um serviço “vazio”, que sempre permite o acesso.

A priorização de tráfego na rede, ou até mesmo a reserva de largura de banda, é uma outra forma de se garantir QoS em um sistema de mídia contínua sob demanda. Desta forma podemos evitar — ou ao menos minimizar — a ocorrência de congestionamento na transmissão do conteúdo. Assim, o *buffer* necessário para tratar as deficiências na transmissão pela rede pode ser reduzido.

No projeto Travis-QoS[16], um trabalho conjunto dos laboratórios LCP[19] e Ravel[26] da COPPE/UFRJ com o Departamento de Informática [27] da UFES, são previstas soluções para garantia de qualidade de serviço aos usuários de um sistema de mídia contínua sob demanda. Criado, inicialmente, a partir de soluções fornecidas pelo LCP e Ravel — tais como GloVE [24, 7, 9] e AirStrike [17] — neste projeto são tratados mecanismos de acesso seguro a sistemas de mídia contínua sob demanda, em redes sem fio. Também são analisados aspectos de escalabilidade, tanto na utilização eficiente dos recursos da rede sem fio como na infra-estrutura, possivelmente em rede cabeada, de distribuição de vídeo até os pontos de acesso ao sistema.

Tabela A.2: Um exemplo de condições de transmissão

Exibição	Disponível	Folga
2 Mbps	512 KB	2 segundos
2 Mbps	4 MB	16 segundos
512 Kbps	1 MB	16 segundos
2 Mbps	16 MB	67 segundos

A.4.2 Tempo disponível para recuperação de falhas

Em geral, o QoS está relacionado com a percepção do usuário na exibição do conteúdo. Interrupções na transmissão, por quaisquer motivos, usualmente não comprometem a qualidade de serviço, se não causarem impacto na exibição do conteúdo. Assim, se a conectividade é perdida — causando interrupção na transmissão — porém logo recuperada, os blocos podem ser consumidos do *buffer* (Figura A.15), não interrompendo a exibição. Desta forma, podemos considerar que a qualidade de serviço foi mantida.

Assim, a recuperação de uma falha deve manter a taxa de preenchimento do *buffer* em um nível aceitável. Isso significa que o exibidor deve ter blocos disponíveis em quantidade suficiente para exibir, enquanto o processo de recuperação ocorre. Este tempo pode ser muito curto, dependendo de vários fatores, tais como: taxa de exibição, tamanho do *buffer*, preenchimento do *buffer* no momento da falha, taxa de transmissão, preenchimento do *buffer* no novo provedor e etc. A tabela A.2 mostra alguns exemplos, calculados de acordo com a fórmula:

$$folga = (disponível * 8) / taxa$$

Os valores utilizados são:

- **Taxa de exibição** - Taxa média, em bits por segundo, utilizada pelo exibidor. Ex: 2 Mbps=2 * 10⁶ bits por segundo.
- **Disponível** - Quantidade de vídeo disponível no *buffer*, em bytes. Ex: 16 MB=16 * 2²⁰ bytes. Podemos utilizar, de forma equivalente, o tamanho do *buffer* multiplicado pelo seu percentual de preenchimento.
- **Folga** - Tempo, em segundos, disponível para a recuperação. Este é o tempo em que o exibidor poderá consumir o *buffer* sem interrupção.

Os valores de folga na tabela A.2 parecem bastante satisfatórios. O problema é que a folga obtida para a recuperação precisa ser alcançada de duas maneiras (que podem ser combinadas):

- **Pagando o tempo em “folga” como latência inicial mínima.** Precisaríamos aguardar por n segundos antes de exibir, apenas preenchendo o *buffer*. Assim, teríamos uma folga de n segundos, assumindo que a taxa de recepção é igual à taxa de exibição.
- **Aumentando a taxa de recepção, para que seja maior que a taxa de exibição.** Desta forma, o *buffer* seria preenchido mais rapidamente do que é consumido, aumentando seu percentual de preenchimento. Contudo, no limite, teríamos que interromper o provimento, pois a capacidade do *buffer* seria esgotada. Alternativamente, poderíamos aumentar a capacidade do *buffer*, mas esta opção poderia levar a termos todo o conteúdo em *buffer*, potencialmente exaurindo os recursos na máquina cliente.

Assim, a folga disponível para efetuarmos a recuperação precisa ser “comprada” antecipadamente, seja com o aumento da latência inicial, seja com o acréscimo de vazão de rede (ou ambos). Além disso, os nós intermediários precisam estar preparados para enviar o conteúdo ao usuário solicitante.

A.4.3 Migração de streams

Para garantia de QoS, em alguns casos, não será possível recuperar uma conexão. O nó pode ter encerrado abruptamente ou, em caso de redes móveis, a conectividade com o nó pode não mais existir. Para contornar este problema, seria necessário **migrar** para outro provedor de vídeo (assumindo, é claro, que o cliente ainda tenha conectividade com algum outro nó).

Para que não haja impacto na exibição, a migração de provedor deve manter a taxa de preenchimento do *buffer* em um nível aceitável, conforme visto na seção A.4.2. Contudo, o uso da migração não precisa ser, necessariamente, transparente para o usuário. Migrar uma transmissão para um outro nó, interrompendo a exibição, ainda pode ser útil. Mesmo ocasionando uma pausa, a migração seria empregada para garantir a **continuidade** do serviço.

A **migração** de streams pode ter aplicações relevantes para a aplicação de mídia contínua sob demanda. Em cenários com **mobilidade**, esta técnica pode ser utilizada para que

o cliente utilize um distribuidor mais adequado à transmissão, após ter mudado de ponto de acesso. Quando falamos em **escalabilidade**, a migração serviria para balancear carga entre nós. Por exemplo, com a entrada de um (ou mais) distribuidor adicional, poderíamos redistribuir os clientes de um distribuidor sobre-carregado. Em um cenário de **tolerância a falhas**, migrar streams que estavam sendo atendidas por um nó intermediário para outro pode ser a garantia da continuidade do serviço.

Utilizar o conceito de migração é especialmente importante quando combinado com algum mecanismo de controle de admissão. Usualmente, são tratados de forma diferente um novo usuário no sistema e um usuário já conectado migrando de um nó para outro. Assim, tratando mobilidade por exemplo, um cliente que esteja mudando de ponto de acesso teria prioridade na transmissão do conteúdo que já está assistindo, em detrimento de um novo usuário se conectando ao sistema.

Há algumas técnicas para prepararmos um grafo de distribuição para migração de streams. Em LEE [20], é apresentada uma técnica para previsão de migração de conexão, em uma rede de telefones celulares. Neste artigo, apresenta-se a idéia de *branch connections*, indicando às células vizinhas a possibilidade de um usuário entrar em uma delas.

No contexto de migração de streams, o processo é um pouco mais complicado. Além de garantir conectividade, precisamos preparar o *buffer* dos nós intermediários vizinhos com o vídeo utilizado. Esta preparação implicaria em consumo de recursos possivelmente desnecessário nos nós vizinhos. Equilibrar o consumo de recursos e a baixa latência na migração é fundamental.