

UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

JOÃO DA MATA LIBÓRIO FILHO

GERENCIAMENTO DE CONTEÚDO MULTIMÍDIA  
EM REDES CDN-P2P

Manaus/AM  
2012

JOÃO DA MATA LIBÓRIO FILHO

GERENCIAMENTO DE CONTEÚDO MULTIMÍDIA  
EM REDES CDN-P2P

Dissertação apresentada ao Programa de Pós-graduação em Informática da Universidade Federal do Amazonas como requisito parcial para obtenção do grau de Mestre em Informática.

**Área de concentração:** Redes de computadores e telecomunicações.

**Orientador:**

Prof. Dr. César Augusto Viana Melo

Manaus/AM

2012

Ficha Catalográfica  
(Catalogação realizada pela Biblioteca Central da UFAM)

L696g Libório Filho, João da Mata

Gerenciamento de conteúdo multimídia em redes CDN-P2P/João da Mata Libório Filho. – Manaus, AM : UFAM, 2012.

85f.: il. color.

Dissertação (Mestrado em Informática) – Universidade Federal do Amazonas, 2012.

Orientador: Prof<sup>o</sup> Dr<sup>o</sup>. César Augusto Viana Melo

1. Informática- Rede de distribuição 2. Políticas - Gerenciamento 3. Computação - Multimídia I. Melo, César Augusto Viana (Orient.) II. Universidade Federal do Amazonas III. Título

CDU (1997) 004.03(043.3)



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



## FOLHA DE APROVAÇÃO

“Gerenciamento de Conteúdo Multimídia em redes CDN-P2P”

JOÃO DA MATA LIBÓRIO FILHO

Dissertação defendida e aprovada pela banca examinadora  
constituída pelos Professores:

PROF. CÉSAR AUGUSTO VIANA MELO – PRESIDENTE

PROF. EDUARDO JAMES PEREIRA SOUTO – MEMBRO

PROF. NELSON LUIS SALDANHA DA FONSECA – MEMBRO

Manaus, 22 de março de 2012.

*Dedico esta dissertação aos meus pais,  
ao meu filho Juan Victor e minha  
filha Cristine Victória.*

## AGRADECIMENTOS

Meus sinceros agradecimentos:

– a Deus pai, soberano de todo o universo, por ter iluminado meus passos até onde cheguei, e por ter sempre me fortalecido na fé. Obrigado Senhor, graças a ti tenho tido forças pra vencer todos os obstáculos da vida;

– ao Prof. César Melo, não só pela orientação, mas também pelo incentivo, confiança e apoio nesses dois anos de trabalho, valeu meu orientador;

– ao Prof. Cícero Mota, graças aos seus conselhos e apoio foi possível a minha vinda a Manaus cursar o mestrado em informática, a você minha gratidão;

– ao Prof. Antonio Cauper (UEA) que desde a graduação sempre nos incentivou e fez acreditar que poderíamos alcançar vãos mais altos. Você foi uma das pessoas que contribuíram para que eu ingressasse ao mestrado, meu muito obrigado;

– a todos os professores e demais funcionários do ICOMP/UFAM, pelo empenho em fazer esse instituto sempre melhor;

– ao Prof. Nelson Luis da Fonseca (UNICAMP), membro externo da banca, por ter me dado a honra de avaliar meu trabalho e ter participado e contribuído em uma de nossas publicações;

– à Fundação de Amparo à Pesquisa do Estado do Amazonas e a Universidade Federal do Amazonas, é claro, pelo suporte financeiro, sem esse incentivo seria impossível me manter;

– aos meus filhos, Juan Victor e Cristine Victória, a quem dedico este trabalho, fiquei devendo a vocês mais de minha atenção, mais idas ao cinema e passeios. Obrigado pela

compreensão, amo vocês;

– aos meus pais, João Libório e Maria Viana, por terem investido em minha educação, e por ter sempre acreditado em meu sucesso. Agradeço por tudo, sei que com vocês eu sempre posso contar;

– a todos os meus colegas e amigos, com os quais passei a maior parte do tempo nesses dois anos de mestrado. Irão ficar na eterna lembrança os fins de semana que passamos escrevendo artigos e resolvendo listas de PAA e FTC, Joacir Marques, Henry Bilby, Juan Gabriel e Antônio Alberto que o diga :). Valeu pela companhia e ajuda nas horas que precisamos;

– Agradeço também meu amigo e irmão Jhonathan Araújo que desde a graduação tem sido um “camarada da responsabilidade” (palavras dele) :, com quem sempre compartilhamos ideias e em vários momentos pude contar com sua ajuda;

– Agradeço ainda às amigas Teresa Nascimento e Dayanna Monteiro por terem revisado esta dissertação;

– Finalmente, a todas as pessoas que contribuíram direto e indiretamente para que eu pudesse obter o grau de mestre.

*“Science is what we understand  
well enough to explain to a computer.*

*Art is everything else we do.”*

***Donald Knuth***

A escalabilidade e a alta demanda por recursos são os principais desafios que os provedores de conteúdo enfrentam na viabilização de aplicações de vídeos sob demanda (VoD). O site mais popular de compartilhamento de vídeos, o YouTube, tem mais de 4 bilhões de visualizações por dia e 60 horas de vídeo são armazenadas a cada minuto em seus servidores. Sistemas híbridos (CDN-P2P) têm sido apresentado como uma solução escalável para distribuição de VoD. Nesses sistemas, pares compartilham seus recursos diminuindo a demanda sobre a infraestrutura da rede de distribuição de conteúdo (CDN). Por outro lado, os servidores da CDN garantem a disponibilidade de conteúdo quando as contribuições dos pares são limitadas pelo *churn*, ou quando o conteúdo for inédito aos pares da rede par a par (P2P). No entanto, o conteúdo distribuído nesses sistemas precisa ser gerenciado, de forma que a carga de trabalho submetida aos servidores da CDN seja minimizada. Uma questão a ser investigada nesse sistema é o impacto do *churn*, isto é, o efeito criado pelo ciclo de entrada e saída dos pares sobre os mecanismos de gerência do conteúdo distribuído pelo sistema. Nesta dissertação avalia-se o impacto do *churn* no desempenho de políticas de gerenciamento de objetos em sistemas CDN-P2P; verificou-se que o impacto no desempenho das políticas diminui com o aumento da capacidade de armazenamento dos pares, no entanto, esse aumento não impacta proporcionalmente a performance das políticas. Em seguida, são propostas, implementadas e avaliadas quatro políticas de gerenciamento de objetos derivadas a partir de dados reais obtidos de coleções de vídeos do YouTube. Essas políticas exploram informações deixadas pelos usuários ou geradas pelo sistema de distribuição dos vídeos para mensurar o valor de um objeto. A suposição acerca da efetividade dessas informações na valoração dos objetos é devido à influência do sistema de recomendação do YouTube no acesso a seu conteúdo, pois esse sistema utiliza-se dessas informações para indicar vídeos aos usuários. As políticas propostas foram capazes de melhorar a disponibilidade do conteúdo em mais de 70%, comparada à disponibilidade proporcionada pela política LFU e mais de 50%, comparada à política GDSP.

**Palavras-chave:** Redes de distribuição de conteúdo, Redes par a par, Conteúdo gerado por usuários, Políticas de gerenciamento e Vídeo sob demanda.

Scalability and high demand for resources are the main challenges that content providers face to deploy Video-on-Demand applications. The most popular site for sharing videos, YouTube, has over 4 billion videos viewed a day and 60 hours of video are uploaded every minute. Hybrid systems (CDN-P2P) have been proposed as a scalable and cost effective solution for VoD distribution. In these systems, peers share their resources decreasing demand on the content distribution network infrastructure (CDN). On the other hand, the CDN's servers guarantee the availability of content when peers' contributions are limited by churn. However, in these systems, the content distributed must be managed so that the CDN servers workload is minimized. An issue to be investigated is the impact of churn, i.e the effect of cycle of peers join and leave, on management policies. Carried studies showed that the performance of policies improves as the storage capacity of peers increases. However, this increasement does not impact proportionately the performance of policies. Later on, we proposed, implemented and evaluated four object management policies derived from data in YouTube video collections. These policies use information left by users or generated by the video distribution system to measure the value of objects. We found that the proposed policies were able to improve the availability of content in more than 70%, compared to the LFU policy, and more than 50% compared to GDSP policy.

**Keywords-** Content Distribution Networks, Peer-to-Peer Networks, User's Generated Content, and Video-on-Demand.

## LISTA DE FIGURAS

2.1	Arquiteturas de distribuição de VoD. . . . .	p. 19
2.2	Transmissão de vídeo ao vivo. . . . .	p. 21
2.3	Transmissão de vídeo sob demanda. . . . .	p. 22
2.4	Cliente-servidor tradicional. . . . .	p. 23
2.5	Cliente-servidor com <i>proxy</i> . . . . .	p. 24
2.6	Rede de distribuição de conteúdo. . . . .	p. 25
2.7	Dinâmica da computação P2P. . . . .	p. 28
2.8	Árvore simples para fluxo de vídeo em P2P. . . . .	p. 29
2.9	Processo de recuperação em árvore. . . . .	p. 30
2.10	Árvore múltipla para fluxo de vídeo em P2P. . . . .	p. 31
2.11	Distribuição de conteúdo por malha. . . . .	p. 32
2.12	Redes sobrepostas semânticas. . . . .	p. 33
2.13	Modelos de redes sobrepostas semânticas baseadas em campos. . . . .	p. 34
2.14	Arquitetura híbrida CDN-P2P. . . . .	p. 35
3.1	Dinâmica da política LRU. . . . .	p. 38
3.2	Dinâmica da política LFU. . . . .	p. 39
3.3	Dinâmica da política GD. . . . .	p. 41
3.4	Dinâmica de políticas baseadas em contexto e orientadas a conteúdo. . . . .	p. 42

4.1	Registros da coleção de vídeos. . . . .	p. 45
4.2	Distribuição dos vídeos por popularidade para a coleção Ent06. . . . .	p. 46
4.3	Histograma do tempo de duração dos vídeos. . . . .	p. 46
4.4	Metadado da coleção de vídeo. . . . .	p. 48
4.5	Ajuste de curva para o padrão de acesso das coleções de vídeos. . . . .	p. 49
4.6	Histograma do tempo de duração dos vídeos. . . . .	p. 51
5.1	A infraestrutura do sistema de distribuição CDN-P2P. . . . .	p. 54
5.2	Estrutura geral do simulador. . . . .	p. 55
5.3	CCDF da duração das sessões de UP e DOWN para os modelos de <i>churn</i> exponencial e cauda longa. . . . .	p. 58
5.4	Taxa de decréscimo de acerto das políticas de gerenciamento de objetos. . . . .	p. 59
5.5	A diminuição do número de pares ativos. . . . .	p. 60
5.6	Quantidade de dados replicados manipulados pelas políticas. . . . .	p. 61
5.7	Taxa de decréscimo de acertos das políticas de gerenciamento de objetos em um sistema sem replicação de dados. . . . .	p. 62
6.1	Modelagem do problema de gerência de vídeo em sistema CDN-P2P. . . . .	p. 66
6.2	Ganho sobre a política LFU para o cenário convergente entre padrões de acesso local e global. . . . .	p. 74
6.3	Ganho sobre a política LFU para o cenário divergente entre padrões de acesso local e global. . . . .	p. 76

## LISTA DE ABREVIATURAS E SIGLAS

VoD	Video-on-demand,	p. 10
QoE	Quality of Experience,	p. 10
CDNs	Content Delivery Networks,	p. 10
VoIP	Voice over IP,	p. 15
ISP	Internet Service Provider,	p. 20
TTL	Time to Live,	p. 30
LRU	Least-Recently Used,	p. 35
LFU	Least-Frequently Used,	p. 37
GDS	Greedy Dual Size,	p. 39
GDSF	Greedy Dual Size with Frequency,	p. 40
GDSP	GDS-Popularity,	p. 40
PARTIAL	Proportional Partial Caching Algorithm for P2P traffic,	p. 40
UGC	User Generated Content,	p. 41
POP	Management Policy Content-oriented Popularity-based,	p. 41
CCDF	Complementary Cumulative Distribution Function,	p. 48
GDVF	Greedy-Dual Size View Frequency-based,	p. 70
GDSR	Greedy-Dual Size Rating-based,	p. 70
GDSVR	Greedy-Dual Size View Rate-based,	p. 71
GDSA	Greedy-Dual Size Adaptive Context,	p. 73

<b>1</b>	<b>Introdução</b>	p. 15
1.1	Principais Contribuições	p. 16
1.2	Artigos Publicados	p. 17
1.3	Estrutura da Dissertação	p. 17
<b>2</b>	<b>Distribuição de Vídeo sob Demanda</b>	p. 19
2.1	Tipos de Fluxo de Vídeo	p. 20
2.1.1	Fluxo de Vídeo ao Vivo	p. 21
2.1.2	Fluxo de Vídeo sob Demanda	p. 21
2.2	Arquitetura Cliente-Servidor	p. 22
2.2.1	A arquitetura Cliente-Servidor Tradicional	p. 23
2.2.2	A Arquitetura Cliente-Servidor Baseada em Proxy	p. 23
2.2.3	Redes de Distribuição de Conteúdo (CDNs)	p. 24
2.3	Arquitetura Par a Par	p. 28
2.3.1	Distribuição de Fluxo de Vídeo Baseada em Árvore	p. 29
2.3.2	Distribuição de Fluxo de Vídeo Baseada em Malha	p. 31
2.3.3	Distribuição de Fluxo de Vídeo Baseada em Semântica de Conteúdo	p. 33
2.4	Arquitetura Híbrida (CDN-P2P)	p. 34

2.5	Considerações Finais do Capítulo 2 . . . . .	p. 35
<b>3</b>	<b>Políticas de Gerenciamento de Objetos para Sistemas de Distribuição de Conteúdo</b>	<b>p. 37</b>
3.1	Políticas Baseadas na Localidade Temporal . . . . .	p. 37
3.2	Políticas Baseadas na Frequência de Acesso . . . . .	p. 39
3.3	Políticas Baseadas em Custo . . . . .	p. 40
3.4	Políticas Baseadas em Contexto e Orientada a Conteúdo . . . . .	p. 42
3.5	Considerações Finais do Capítulo 3 . . . . .	p. 43
<b>4</b>	<b>As Coleções de Vídeos do YouTube</b>	<b>p. 44</b>
4.1	Coleção Ent06 . . . . .	p. 44
4.2	Coleções Ent11, Esp11 e Mus11 . . . . .	p. 47
4.3	Considerações Finais do Capítulo 4 . . . . .	p. 52
<b>5</b>	<b>Avaliação do Impacto do <i>Churn</i> nas Políticas de Gerenciamento de Objetos</b>	<b>p. 53</b>
5.1	A Distribuição de Conteúdo . . . . .	p. 54
5.2	Padrão de Acesso aos Vídeos . . . . .	p. 55
5.3	Modelos de <i>Churn</i> . . . . .	p. 56
5.4	Resultados Numéricos . . . . .	p. 58
5.5	Considerações Finais do Capítulo 5 . . . . .	p. 63
<b>6</b>	<b>Políticas de Gerenciamento de Objetos em Sistemas CDN-P2P</b>	<b>p. 64</b>
6.1	Caracterizando a Distribuição de Conteúdo em um Sistema CDN-P2P . . . . .	p. 65
6.2	Políticas para Gerenciamento de Objetos Baseadas em Custo e em Conteúdo . . . . .	p. 67
6.2.1	Fator de Custo Baseado em Frequência de Visualizações . . . . .	p. 68
6.2.2	Fator de Custo Baseado em Qualificação . . . . .	p. 69

6.2.3	Fator de Custo Baseado em Taxa de Visualizações . . . . .	p. 69
6.2.4	O Fator de Custo Adaptativo Baseado em Frequência de Visualização . . . . .	p. 70
6.3	Resultados Numéricos . . . . .	p. 72
6.4	Considerações Finais do Capítulo 6 . . . . .	p. 77
<b>7</b>	<b>Considerações Finais</b>	<b>p. 78</b>
7.1	Trabalhos Futuros . . . . .	p. 79
	<b>Referências</b>	<b>p. 80</b>

# CAPÍTULO 1

## INTRODUÇÃO

Aplicações de vídeo sob demanda, do inglês *Video-on-demand (VoD)*, têm alcançado cada vez mais popularidade na Internet. Devido ao seu alto consumo de banda e elevado número de acessos, a viabilização dessas aplicações representa um constante desafio aos provedores desses conteúdos. O YouTube, o mais popular site de compartilhamento de vídeos, tem mais de 4 bilhões de visualizações por dia e 60 horas de vídeo são armazenadas a cada minuto em seus servidores (YOUTUBE, 2012). Sistemas de distribuição que sejam escaláveis, que apresentem menor custo financeiro e que ofereçam boa qualidade de experiência (*Quality of Experience - QoE*) aos usuários têm sido alvo de pesquisas recentes (PASSARELLA, 2012).

Redes de distribuição de conteúdo (*Content Delivery Networks - CDNs*) têm sido utilizadas na distribuição de VoD por oferecer alta disponibilidade, grande capacidade de armazenamento e pequeno atraso. No entanto, as CDNs têm alto custo financeiro de implantação e manutenção, fruto do esforço empreendido para melhorar escalabilidade do sistema (HAREESH; MANJIAH, 2011). Devido ao baixo custo de implantação, robustez e natural escalabilidade, sistemas par-a-par (*Peer-to-Peer - P2P*) têm se mostrado uma solução atrativa na distribuição de VoD em larga escala pela Internet. Porém, em tais sistemas, voltado para distribuição de VoD, a disponibilidade de conteúdo está diretamente relacionada com o número de pares conectados simultaneamente no sistema. Se o número de pares for reduzido, os usuários do sistema podem experimentar baixa QoE, com as aplicações sofrendo longos atrasos e frequentes perdas de continuidade.

Para tentar superar essas limitações sistemas híbridos têm sido projetado para explorar as vantagens de ambas as tecnologias, CDN e P2P. Nesses sistemas a CDN garante a

disponibilidade do conteúdo e a rede P2P auxilia sua distribuição, compartilhando recursos de seus pares, diminuindo assim a carga de trabalho imposta aos servidores da CDN. No entanto, sistemas híbridos, por si só, não garantem a efetiva distribuição de conteúdo se não forem gerenciados adequadamente. Por exemplo, objetos replicados em excesso nos pares da parte P2P desses sistemas podem dominar a ocupação dos *buffers*, diminuindo a disponibilidade de conteúdo. Conseqüentemente, identificar objetos valiosos e torná-los disponível, na parte P2P de tais sistemas, tende a reduzir a carga de trabalho submetida aos servidores da CDN.

Nesta dissertação aborda-se o problema de gerência dos objetos na parte P2P de um sistema híbrido CDN-P2P. Para isso, a partir de dados reais coletados do site YouTube, exploram-se informações deixadas pelos usuários ou geradas pelo sistema de distribuição dos vídeos, tais como visualizações, qualificação e idade dos vídeos. Tais informações são utilizadas como critério de valoração em políticas de gerenciamento de *buffers* que identificam os objetos a serem mantidos nos pares de um sistema CDN-P2P. A suposição acerca da efetividade das informações de qualificação, visualizações e idade na identificação de objetos importantes, a serem mantidos nos pares, deve-se a influência do sistema de recomendação do YouTube no acesso a seu conteúdo. Esse sistema faz uso dessas informações para indicar vídeos que sejam relevantes aos usuários, sendo responsável por 60% de todos os cliques a vídeos a partir da página inicial do YouTube (DAVIDSON et al., 2010).

## 1.1 Principais Contribuições

As principais contribuições deste trabalho são apresentadas a seguir, na ordem em que aparecem nesta dissertação:

- Apresentação do estado da arte de sistemas de distribuição de vídeos, descrevendo as principais arquiteturas desses sistemas e abordando as vantagens e desvantagens de cada uma;
- Apresentação do estado da arte das políticas de gerenciamento de objetos utilizadas em sistemas de distribuição de conteúdos;
- Análise do padrão de acesso e da distribuição da duração dos vídeos nas coleções do YouTube utilizadas nos experimentos numéricos. Essa análise contribui para oportunizar melhorias nos sistemas de distribuição de vídeos;

- Avaliação do impacto do *churn* no desempenho de políticas de gerenciamento de objetos em sistemas CDN-P2P;
- Verificação do aumento da capacidade individual de armazenamento dos pares, como uma medida para manter o desempenho das políticas;
- Avaliação do quanto à replicação de dados afeta o desempenho das políticas, analisando-se um sistema com e sem replicação;
- Proposta, implementação e avaliação de políticas de gerenciamento de objetos em redes CDN-P2P baseadas em sinais disponíveis no site provedor do conteúdo. Avaliação do desempenho dessas políticas mostra que, em alguns casos, a disponibilidade de conteúdo pode ser melhorada em mais de 70%, comparada à gerência realizada com a política LFU, superando a política GDSP em mais de 50%.

## 1.2 Artigos Publicados

- MELO, C. A. V.; VIEIRA, D.; LIBORIO FILHO, J. M. Impact of churn on object management policies. In: Communications (LATINCOM), 2011 IEEE Latin American Conference on. 2011. p. 1 -6, ISBN 978-1-4673-0277-7.
- MELO, C. A. V.; VIEIRA, D.; LIBORIO FILHO, J. M. Impact of churn on object management policies. IEEE Latin America Transactions. 2012.
- MELO, C. A. V.; LIBORIO FILHO, J. M.; FONSECA, N. L. S. Object management policies in P2P-CDN Networks. In: *Communications (ICC), 2012 IEEE International Conference on*. 2012.
- LIBORIO FILHO, J. M.; MELO, C. A. V.; FONSECA, N. L. S. Políticas de gerenciamento de conteúdo gerado por usuários em sistemas CDN-P2P. in: *WP2P SBRC 2012*. 2012.

## 1.3 Estrutura da Dissertação

Esta dissertação está dividida em sete capítulos. Os capítulos 2 e 3 apresentam revisão do estado da arte de arquiteturas de sistemas de distribuição de conteúdo e políticas de gerenciamento de objetos. No Capítulo 2 apresenta-se uma visão geral sobre os sistemas de distribuição de VoD, destacando-se as arquiteturas cliente-servidor, P2P e abordagens híbridas CDN-P2P.

No Capítulo 3 apresenta-se uma revisão das políticas de gerência de objetos. As políticas são abordadas de acordo com o critério de priorização usado para manter os objetos armazenados. Assim, classificam-se essas políticas como: baseadas no tempo em que o objeto foi acessado, baseadas na frequência de acesso, baseadas nos custos dos objetos e baseadas em contexto e orientada a conteúdo.

O Capítulo 4 apresenta o estudo das coleções de vídeo do YouTube utilizadas nos experimentos numéricos realizados nesta dissertação. Analisa-se o padrão de acesso e a distribuição da duração dos vídeos nas coleções. O estudo desses dados permite identificar estatísticas importantes a serem utilizadas na elaboração de políticas de gerenciamento de vídeos em sistema de distribuição.

No Capítulo 5 é avaliado o impacto do *churn* nas políticas de gerenciamento de objetos em redes CDN-P2P. As principais contribuições apresentadas nesse capítulo são duas. Primeiro, avalia-se o aumento da capacidade individual de armazenamento dos pares como uma medida para melhorar o desempenho das políticas de gerenciamento. Depois avalia-se o quanto a replicação de dados afeta o desempenho das políticas, analisando-se um sistema com e sem replicação.

No Capítulo 6 modela-se o problema de gerência de VoD em sistemas CDN-P2P como o problema da mochila múltipla 0-1. Ainda nesse capítulo são apresentadas e avaliadas quatro novas políticas de gerenciamento de objetos em sistemas CDN-P2P. Essas políticas utilizam dados dispostos no site de compartilhamento de vídeo YouTube para melhorar a disponibilidade dos vídeos na parte P2P do sistema de distribuição.

Finalmente, no Capítulo 7 são apresentados comentários gerais sobre o problema estudado, soluções obtidas e as conclusões sobre os resultados. Além disso, são apresentadas as limitações das soluções obtidas e possíveis extensões desta pesquisa.

## CAPÍTULO 2

# DISTRIBUIÇÃO DE VÍDEO SOB DEMANDA

O acesso cada vez maior às infraestruturas de redes com altas taxas de transmissão e a facilidade oferecida pelos serviços da *web 2.0* têm encorajado os usuários a compartilharem e acessarem conteúdo multimídia na Internet. Essa nova realidade impõe desafio de escalabilidade aos sistemas de distribuição desses conteúdos, exigindo que novas estratégias para essa distribuição sejam implementadas de forma a reduzir a carga de trabalho dos servidores e o tráfego nas infraestruturas de transmissão de dados.

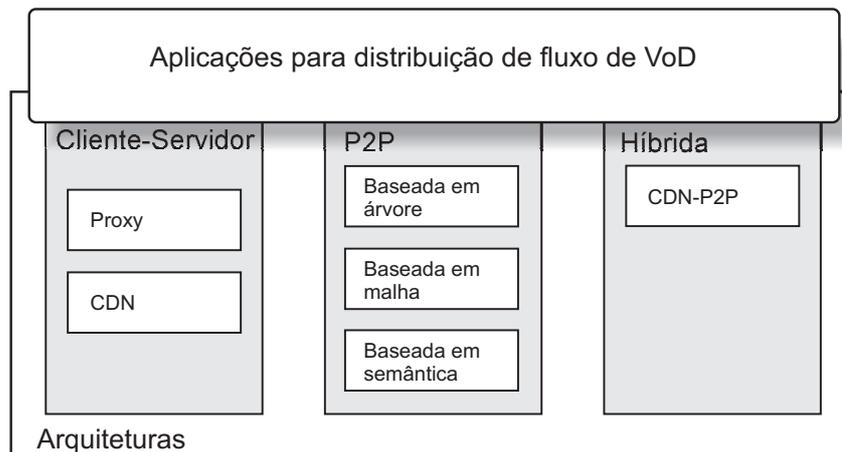


Figura 2.1: Arquiteturas de distribuição de VoD.

Entre as diversas aplicações de transmissão de conteúdo multimídia destacam-se: VoD, fluxo de vídeo ao vivo (*Live Video Streaming*) e voz sobre IP, do inglês *Voice over IP (VoIP)*. Aplicações de VoD, tais como, o Youtube e o Dailymotion (DAILYMOTION, 2012) têm-se caracterizado pelo grande consumo de banda e recursos da Internet (CHENG; LIU, 2009). Na Figura 2.1 ilustra-se uma taxonomia das arquiteturas de aplicação utili-

zadas para construir os sistemas de distribuição de VoD. A primeira delas é a baseada em cliente-servidor, que apresenta duas variações: as que utilizam *proxy* e as CDNs. Outra arquitetura utilizada é a P2P, cujas implementações são agrupadas conforme as estratégias utilizadas na entrega de conteúdo, isto é, baseada em árvore, baseada em malha e baseada em semântica. Para oportunizar-se das vantagens das duas arquiteturas anteriores (cliente-servidor e P2P), uma arquitetura híbrida CDN-P2P tem sido também estudada.

Nas seções seguintes aborda-se com mais detalhes cada uma dessas arquiteturas. Na Seção 2.1 são apresentadas características dos tipos de fluxo de vídeo, ao vivo e sob demanda. Na Seção 2.2 são abordadas as variações da arquitetura cliente-servidor, enquanto que na Seção 2.3 apresenta-se a arquitetura P2P e na Seção 2.4 discute-se a arquitetura híbrida (CDN-P2P). Finalmente na Seção 2.5 apresentam-se as considerações finais deste capítulo.

## 2.1 Tipos de Fluxo de Vídeo

Fluxo de vídeo é uma forma de distribuição de mídia que se caracteriza pela reprodução do conteúdo à medida que esse é disponibilizado. Em outras palavras, o arquivo com o conteúdo é dividido em segmentos e a reprodução acontece a partir dos segmentos que foram recebidos. Fluxo de vídeo tem sido utilizado para disponibilizar vídeos na Internet, permitindo que o usuário assista a esses vídeos sem a necessidade de armazená-los no computador.

Há basicamente dois tipos de fluxo de vídeo, quando se considera o tempo em que esses estão na rede: ao vivo e sob demanda. Para entender melhor a diferença entre esses dois tipos de fluxo de vídeo, considere uma partida de futebol transmitida em tempo real por fluxo de vídeo e uma biblioteca *online* que disponibiliza vídeoaulas, para que seus usuários possam assistir a essas vídeoaulas a qualquer momento. O primeiro é chamado de fluxo de vídeo ao vivo, pois é transmitido em tempo real, sendo necessário que as imagens cheguem ao usuário com o mínimo de atraso. O segundo é caracterizado como fluxo de vídeo sob demanda devido os vídeos ficarem armazenados no servidor, sendo possível assistir a esses vídeos a qualquer instante. Além disso, diferente do fluxo de vídeo ao vivo, operações típicas de avanço e retrocesso são permitidas em fluxo de vídeo sob demanda.

### 2.1.1 Fluxo de Vídeo ao Vivo

Nas aplicações de fluxo de vídeo ao vivo o conteúdo é distribuído em tempo real a todos os usuários que estejam conectados e a reprodução acontece de forma síncrona. Esse tipo de aplicação é utilizado principalmente para reprodução de vídeos que precisam ser acessados em tempo real, tais como, a transmissão ao vivo de um programa de TV ou uma partida de um evento esportivo. Na Figura 2.2 é ilustrada a distribuição de vídeo ao vivo. Observa-se que em todos os usuários a reprodução ocorre de forma sincronizada, ou seja, o segmento de reprodução do vídeo precisa chegar com o menor atraso possível.

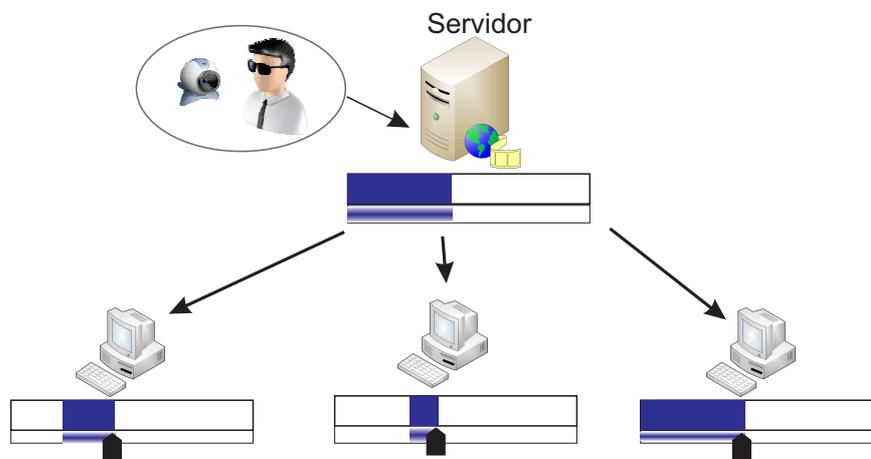


Figura 2.2: Transmissão de vídeo ao vivo.

Aplicações *multicast* têm sido propostas como uma maneira eficiente de distribuição de fluxo de vídeo ao vivo. Funções *multicast* têm sido implementadas na camada de aplicação, sob duas formas principais: baseada em árvore (Seção 2.3.1) e baseada em malha (Seção 2.3.2).

### 2.1.2 Fluxo de Vídeo sob Demanda

Diferente das aplicações de fluxo de vídeo ao vivo, as aplicações de fluxo sob demanda disponibilizam os vídeos para que os usuários possam assistir a eles a qualquer momento. Assim, a transmissão de um dado vídeo ocorre em tempos diferentes para cada usuário, ou seja, a reprodução ocorre de forma assíncrona.

Na Figura 2.3 ilustra-se a distribuição de VoD, onde cada usuário pode assistir a vídeos diferentes em um dado momento, ou um grupo de usuários pode assistir ao mesmo vídeo, sendo que cada usuário pode assistir a partes diferentes.

Vídeos sob demanda têm sido utilizado para a distribuição de diferentes tipos de

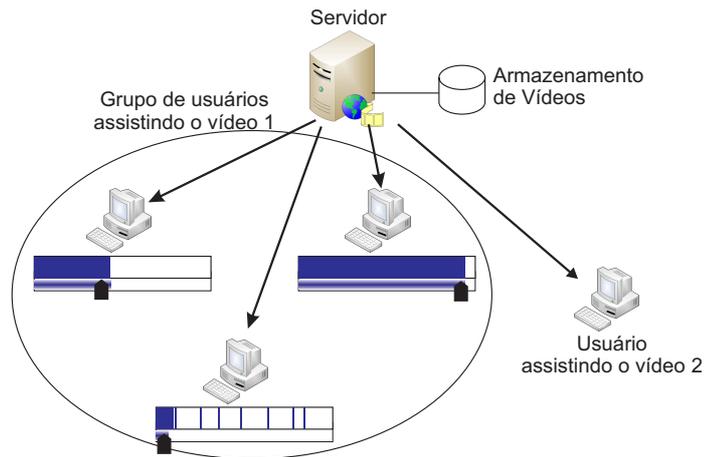


Figura 2.3: Transmissão de vídeo sob demanda.

conteúdo multimídia, tais como, entretenimento, bibliotecas de vídeos digitais, ensino a distância e comércio eletrônico. O YouTube e o Dailymotion, são exemplos típicos de aplicações que implementam VoD.

## 2.2 Arquitetura Cliente-Servidor

A arquitetura cliente-servidor tem sido amplamente utilizada na implementação de aplicações para a Internet, por exemplo, comércio eletrônico, Internet *banking* e redes sociais. Uma aplicação que implementa essa arquitetura, caracteriza-se por suas funcionalidades estarem distribuídas em duas partes: cliente e servidora. A parte cliente requisita serviços da parte servidora e esta se encarrega de responder aos pedidos de vários clientes.

Nessa arquitetura, os clientes são meros consumidores dos recursos do sistema. De fato, os recursos, isto é, capacidade de armazenamento, processamento e transmissão, que a aplicação demanda são majoritariamente fornecidos pela parte servidora da aplicação. Por isso, quando o número de clientes que acessam o conteúdo simultaneamente aumenta além da capacidade de provisionamento do servidor, ocorre lentidão na prestação do serviço ou até mesmo a sua negação. Além disso, o servidor pode estar geograficamente distante de um cliente e a comunicação entre cliente e servidor apresentar longos atrasos, tornando, em alguns casos, inviável a utilização do serviço. A construção de aplicações a partir da arquitetura cliente-servidor apresenta algumas variações que são discutidas a seguir.

### 2.2.1 A arquitetura Cliente-Servidor Tradicional

A arquitetura cliente-servidor é tradicionalmente a mais utilizada na construção de aplicações de vídeos sob demanda (LIU; GUO; LIANG, 2008). Nessa arquitetura o servidor mantém os vídeos armazenados e entrega diretamente aos clientes quando estes realizam requisições. Na Figura 2.4 é ilustrada a distribuição de vídeo nessa arquitetura. Observa-se que um único servidor atende requisições de diversos clientes utilizando conexão *unicast*.

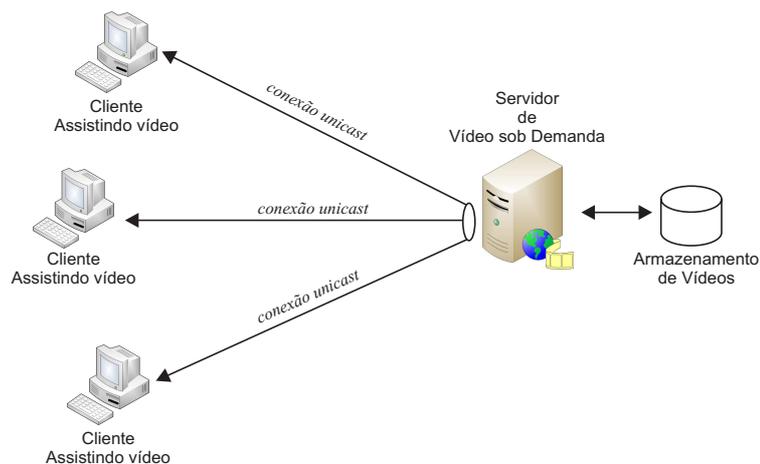


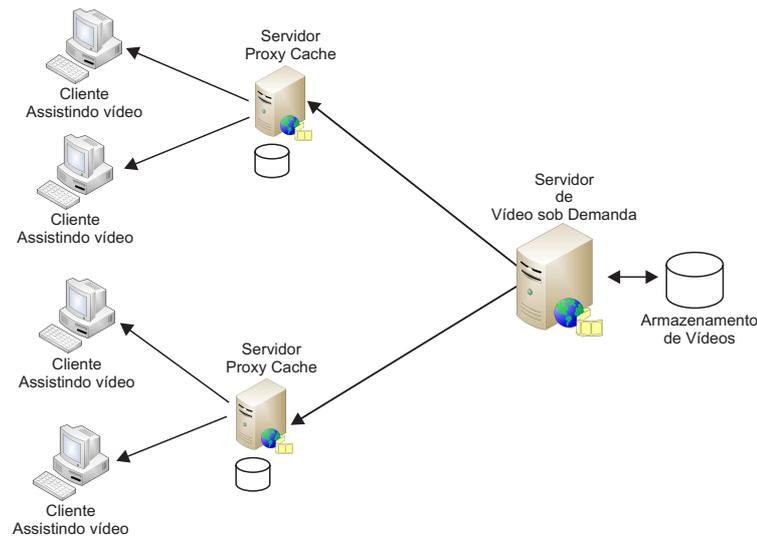
Figura 2.4: Cliente-servidor tradicional.

O grande desafio dessa arquitetura é a escalabilidade da aplicação, isto é, manter funcional a aplicação quando o volume de conteúdo disposto no servidor cresce, juntamente com o volume de requisições simultâneas a tais conteúdos. Realizar atualização que aumente o volume de recursos do servidor pode ser uma solução de custo financeiro elevado quando a quantidade de usuários do sistema chega a milhões.

### 2.2.2 A Arquitetura Cliente-Servidor Baseada em Proxy

Quando o usuário final está distante do servidor a conexão fica vulnerável a alta latência e constantes perdas de pacotes, o que pode degradar a qualidade de experiência (QoE) desse usuário. Para minimizar a degradação dos serviços soluções baseadas em servidores *proxys* têm sido implantadas por provedor de Internet (*Internet Service Provider - ISP*). Esses *proxys* ficam localizados nas redes de bordas e buscam minimizar a latência, reduzir o tráfego que flui pela conexão de saída dos ISPs e, conseqüentemente, reduzem a carga de requisições que chegam diretamente ao servidor (MIR; NATARAJA; RAVIKRISHNAN, 2011).

Na abordagem ilustrada na Figura 2.5, o servidor *proxy* intercepta as requisições

Figura 2.5: Cliente-servidor com *proxy*.

direcionadas aos servidores principais e armazena em sua cache o conteúdo requisitado pelos clientes. Quando é realizada requisição de um conteúdo que o servidor *proxy* já possua em sua cache, tal conteúdo é servido diretamente ao cliente, diminuindo assim, a latência e reduzindo o tráfego de saída do ISP.

### 2.2.3 Redes de Distribuição de Conteúdo (CDNs)

O sucesso da Internet e sua gama de serviços junto com a explosão do uso de conteúdos multimídia *online* incentivaram o surgimento e o crescente interesse pelas CDNs (CALZAROSSA; GELENBE, 2004). Tais redes têm sido propostas para reduzir a carga de trabalho do servidor principal, melhorar a acessibilidade e manter correções por meio da replicação de conteúdos (VAKALI; PALLIS, 2003). Com o uso de CDNs os conteúdos são distribuídos aos servidores localizados próximo do usuário, resultando em aplicações *web* com menor tempo de resposta e com alta disponibilidade.

As CDNs mantêm pontos de presença (*Point of Presence - PoP*), em locais estratégicos com servidores substitutos interligados, nos quais os conteúdos são replicados. Quando um usuário requisita algum conteúdo, sua requisição é atendida pelo servidor substituto que se encontra mais próximo, como é ilustrado na Figura 2.6.

Como os servidores substitutos possuem alta capacidade de armazenamento e boa disponibilidade de banda nas conexões que formam a CDN, os conteúdos replicados nessas redes não são necessariamente definidos pela demanda de clientes e podem utilizar políticas de replicação mais ricas e flexíveis (PATHAN; BUYYA; VAKALI, 2008). Segundo Passarella (2012) o desenvolvimento e manutenção das CDNs não são acessíveis aos edito-

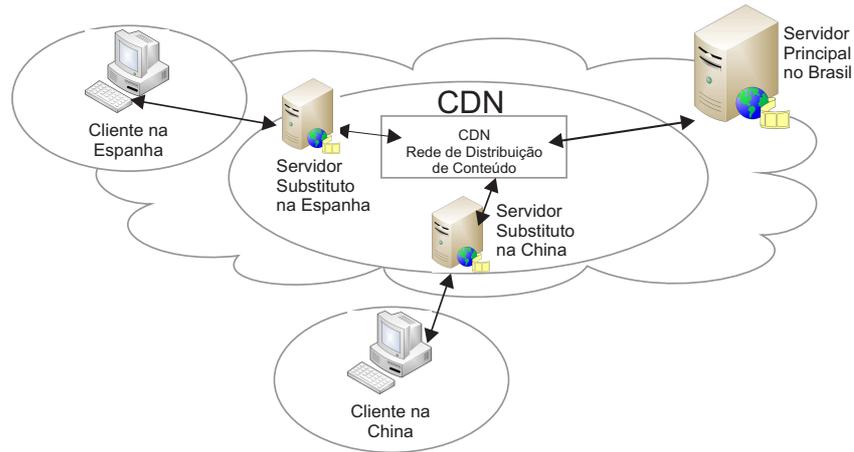


Figura 2.6: Rede de distribuição de conteúdo.

res de conteúdo. Essas redes, na maioria das vezes, são implementadas e comercializadas por companhias dedicadas a esses serviços, por exemplo, a Akamai (AKAMAI, 2012) e a Edgecast (EDGECAST, 2012).

A distribuição de conteúdo por meio de CDN lida basicamente com três problemas: *i*) identificar qual a melhor localização dos servidores substitutos; *ii*) identificar que conteúdo replicar e *iii*) identificar em qual servidor replicar cada parte do conteúdo. A seguir, é discutido com mais detalhes cada um desses problemas.

**Onde posicionar os servidores substitutos?** Em lugares estratégicos na Internet tal que esses servidores estejam mais próximos do cliente que solicita o serviço. A determinação desses locais caracteriza um problema de otimização que pertence à classe dos problemas NP-Difícil (PASSARELLA, 2012).

Em Karlsson e Karamanolis (2004) esse problema é modelado como uma variação do problema de cobertura de conjuntos (VAZIRANI, 2001) onde é apresentado um método para auxiliar projetistas de sistemas a escolherem heurísticas de posicionamento que melhor atendam suas metas de desempenho com um custo de infraestrutura mais baixo possível. A seguir são apresentados os principais algoritmos de posicionamento de servidores substitutos.

- **Algoritmo baseado em árvore** (LI et al., 1999): é baseado na suposição de que a topologia das redes subjacentes são árvores. Em linhas gerais, o algoritmo divide uma árvore  $T$  em diversas pequenas árvores  $T_i$  e mostra que a melhor maneira de posicionar os servidores  $t > 1$  na árvore  $T$  é posicionar o servidor  $t'_i$  da melhor maneira em cada árvore  $T_i$ , onde  $\sum_i t'_i = t$ .
- **Algoritmo Greedy** (QIU; PADMANABHAN; VOELKER, 2001): supõe-se que

seja preciso escolher  $M$  servidores substitutos entre  $N$  servidores potenciais. É escolhido um substituto de cada vez. Na primeira iteração é avaliado cada um dos  $N$  servidores individualmente para determinar qual o melhor para hospedar uma réplica. Calcula-se o custo de cada local, considerando os acessos de clientes realizados nesse servidor e escolhe-se o servidor que gera o menor custo. Na segunda iteração, busca-se um servidor que produza um menor custo combinado com o já escolhido anteriormente. A iteração continua até que seja escolhido  $M$  servidores.

- **Hot Spot** (QIU; PADMANABHAN; VOELKER, 2001): posiciona os servidores substitutos próximo dos clientes que geram maior carga, sendo os locais classificados de acordo com a quantidade de tráfego gerado na sua proximidade.
- **Algoritmo super ótimo** (QIU; PADMANABHAN; VOELKER, 2001): é baseado no relaxamento Lagrangeano com a otimização do subgradiente (CORNUJOLS; FISHER; NEMHAUSER, 1977). Resolve o problema do  $k$ -mediano mínimo (MIRCHANDANI; FRANCIS, 1990) para o qual o problema de localização pode ser modelado como segue:  $y_i$  pode assumir valores de 0-1 com  $i \in N$ , indicando se o local  $i$  é selecionado como um centro e  $x_{ij}$  variando de 0-1, onde  $i, j \in N$  indica se a localização  $j$  é atribuída para o centro de  $i$ . De maneira que seja mínimo o valor de  $\sum_{i,j \in N} d_j c_{ij} x_{ij}$ .

**Como replicar o conteúdo?** Parcialmente ou completamente. A técnica de replicar apenas parte do conteúdo é uma solução escalável, uma vez que replicar todo o conteúdo tem maior consumo da banda principalmente se esse for volumoso.

Em Fujita et al. (2004) é apresentada uma heurística para resolver o problema da replicação completa e parcial de conteúdo. Nessa heurística o conteúdo no servidor de origem é agregado em múltiplos grupos e a replicação é realizada por granularidade. Baseada na maneira como os grupos são criados, a heurística classifica os grupos como sendo de replicação estática e de replicação dinâmica.

- **Grupo de replicação estática:** nessa abordagem os objetos são antecipadamente agregados em grupos. Os grupos são formados por objetos pertencentes ao mesmo diretório e baseados em características estáticas, tais como, tipo de mídia e tamanho dos objetos.
- **Grupo de replicação dinâmica:** nos grupos de replicação dinâmica os conteúdos são agregados dinamicamente baseados em parâmetros dinâmicos, tais como, carga

do servidor e carga da rede. Em Fujita et al. (2004) é apresentado um cenário em que os objetos são agrupados com base na frequência de acesso. Periodicamente os objetos são ordenados pela frequência de acesso por unidade de tempo. O conjunto de objetos é dividido de maneira que cada grupo possua o mesmo número de objetos. Essa abordagem utiliza características dinâmicas dos objetos para criar os grupos, tal como, a popularidade. Tem melhor desempenho em relação ao esquema de grupos de replicação estática.

**Em qual servidor replicar?** No servidor substituto mais adequado. Essa técnica é conhecida como terceirização de conteúdo, e em Pallis e Vakali (2006) são apresentadas três abordagens para sua implementação.

- **Abordagem proativa e cooperativa:** nessa abordagem o conteúdo é enviado do servidor principal da CDN para os substitutos antecipadamente, ou seja, de forma proativa. O conteúdo é carregado na cache antes de ser acessado e então os servidores substitutos cooperam a fim de reduzir o custo de replicação e atualização. A CDN mantém um mapeamento entre conteúdo e servidores substitutos, assim, cada requisição é direcionada para o servidor mais próximo que contém o objeto solicitado, caso contrário, a requisição é atendida pelo servidor principal.
- **Abordagem reativa e não cooperativa:** nessa abordagem as requisições dos clientes são direcionadas para o servidor substituto mais próximo. Caso esse não possua o conteúdo solicitado, será requisitado do servidor principal, mantendo em sua cache réplica desse conteúdo para que, em solicitações futuras a esse mesmo conteúdo, o servidor substituto não necessite acessar novamente o servidor principal. A desvantagem dessa abordagem é que nem sempre é escolhido o servidor ótimo para responder por uma determinada requisição (JOHNSON et al., 2001).
- **Abordagem reativa e cooperativa:** essa abordagem difere da abordagem não cooperativa pelo fato de haver cooperação entre os servidores substitutos. Quando um servidor substituto não possui o conteúdo requisitado, ao invés de requisitar ao servidor principal, ele busca utilizando indexação distribuída em outro substituto mais próximo, diminuindo assim, a carga no servidor principal.

## 2.3 Arquitetura Par a Par

A computação realizada em sistemas que implementam a arquitetura P2P segue o paradigma da computação distribuída (VU et al., 2010). Nesses sistemas tem-se a descentralização das funções, com os pares desempenhando funções de cliente e de servidor, ou seja, um par tem capacidade tanto de servir quanto de consumir recursos do sistema.

Na Figura 2.7 ilustra-se a dinâmica da computação realizada em sistemas P2P. Cada par disponibiliza metadados que contêm informações sobre o conteúdo que esse compartilha, bem como, informações referentes aos dados de outros pares. Por exemplo, o par *A* (Figura 2.7) faz uma requisição de dados que o par *B* possui, porém, *A* localizou *B* através de outros pares da rede P2P. Quando *A* localiza *B* se estabelece uma conexão entre os dois e então as negociações ocorrem diretamente com o par *B*.

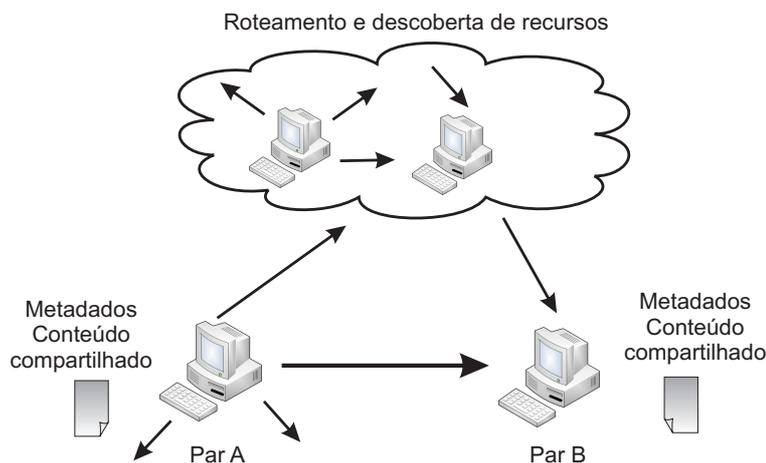


Figura 2.7: Dinâmica da computação P2P.

A arquitetura P2P tem sido usada para construção de sistemas de distribuição de fluxo de vídeo (LIU; GUO; LIANG, 2008). A principal vantagem dessa arquitetura é a capacidade de compartilhamento de recursos dos pares, por exemplo, armazenamento e banda de *upload*. Essa característica torna as aplicações naturalmente escaláveis, pois, à medida que a quantidade de usuários do sistema aumenta, seus recursos também aumentam, em face a contribuição de cada novo par.

As aplicações BitoS (VLAVIANOS; ILIOFOTOU; FALOUTSOS, 2006), PPLive (PPLIVE, 2012) e PPStream (PPSTREAM, 2012) são exemplos de implementações da arquitetura P2P voltada para distribuição de fluxo de vídeo. Baseada na forma como suas redes sobrepostas, do inglês *overlays*, são construídas, aplicações P2P para distribuição de fluxo de vídeo são agrupadas em duas categorias: baseada em árvore e baseada em malha (PELTOTALO et al., 2008). Em aplicações P2P os pares conectam-se a um conjunto de outros

pares randomicamente. Para influenciar a conexão, baseado em interesse, surgiram as redes sobrepostas semânticas que são abordadas no final desta seção.

### 2.3.1 Distribuição de Fluxo de Vídeo Baseada em Árvore

A distribuição de fluxo de vídeo a partir de uma rede sobreposta baseada em árvore, pode ser implementada de duas maneiras, utilizando árvores simples (*single-tree streaming*) e múltiplas árvores (*multi-tree streaming*).

**Árvore Simples.** Na distribuição baseada em árvore simples, a entrega de conteúdo ocorre de um pai para seus filhos e é similar a uma árvore *multicast IP* formada pelos roteadores na camada de rede. Nessa abordagem os pares que participam de uma mesma sessão de vídeo formam uma árvore na camada de aplicação, tendo como raiz o servidor (JANNOTTI et al., 2000).

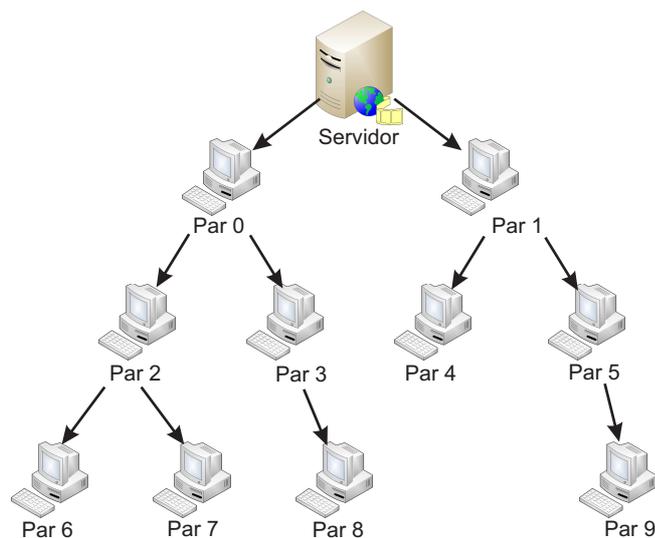
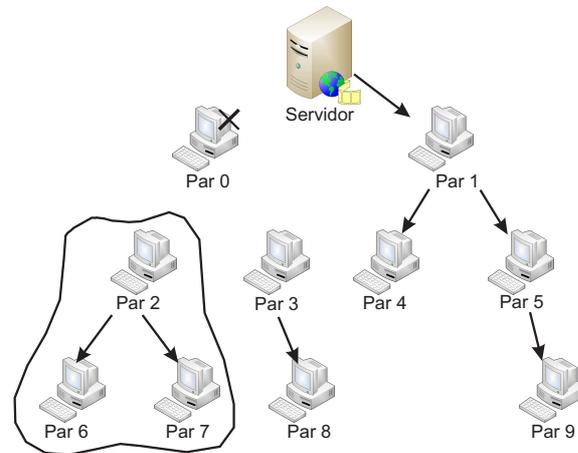


Figura 2.8: Árvore simples para fluxo de vídeo em P2P.

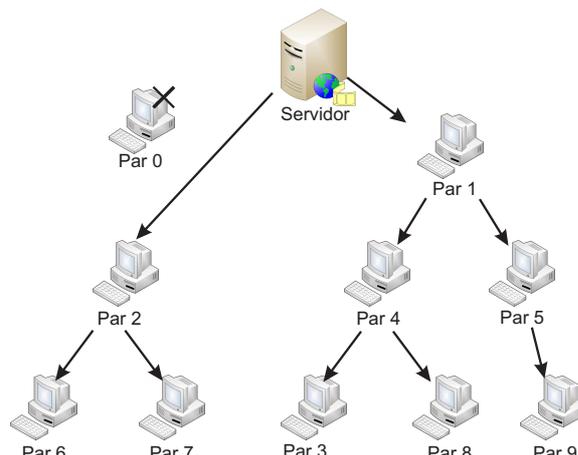
Na Figura 2.8 é apresentada a dinâmica utilizada na abordagem baseada em árvores simples para fluxo de vídeo em P2P. Pode-se observar que no primeiro nível, dois pares (par 0 e par 1) recebem o vídeo diretamente do servidor. No segundo nível, quatro pares recebem o vídeo dos seus pais e três desses entregam o vídeo para o último nível.

Para reduzir o atraso no tempo de entrega do vídeo é importante considerar a profundidade da árvore e aumentar o número de pares em cada nível. Um limitador da quantidade de filhos que um par pode possuir é a largura da sua banda de *upload*. Outro fator importante que deve ser observado é quando um desses pares abandona a rede. Se o par que acaba de sair estiver no último nível, sua saída não gera problemas para a estrutura pois ele não contribui com recursos para a rede. No entanto, se o par que sai da

rede for um par intermediário, seus descendentes também perderão acesso à rede, como é mostrado na Figura 2.9(a). Esse problema é resolvido adotando estratégia de recuperação da rede para que os pares afetados não percam a reprodução do vídeo.



(a) Impacto quando um par sai da rede.



(b) Reconstrução da árvore após a saída de um par.

Figura 2.9: Processo de recuperação em árvore.

Na Figura 2.9(b) é ilustrada a recuperação da árvore com o par afetado realizando uma reconexão com o sistema. A manutenção desse tipo de rede pode ser feita de forma centralizada ou descentralizada, no entanto, em qualquer uma delas, a recuperação em redes com pares muito dinâmicos tem se mostrado pouco ágil (LIU; GUO; LIANG, 2008). Outra deficiência dessa forma de distribuição é quanto a contribuição dos pares folhas (pares 4, 6, 7, 8 e 9 na Figura 2.8). Como esses pares não possuem filhos eles não contribuem com sua banda de *upload* para o sistema.

**Árvore Múltipla.** Para melhor utilizar a banda de *upload* dos pares, minimizando a deficiência de contribuição dos pares folhas apresentada nas árvores simples, em Castro et al. (2003) e em Kostić et al. (2003) foram propostas abordagens utilizando múltiplas

árvores para distribuir fluxo de vídeo.

Nas árvores múltiplas, o servidor divide o fluxo de vídeo em vários subfluxos e esses seguem por árvores distintas ou múltiplas árvores. Para receber o fluxo de vídeo, cada par se junta às diversas árvores e dessa forma participa da distribuição do vídeo em diversos níveis, de intermediário a folha. Assim, a maioria dos pares contribui com suas capacidades de transmissão.

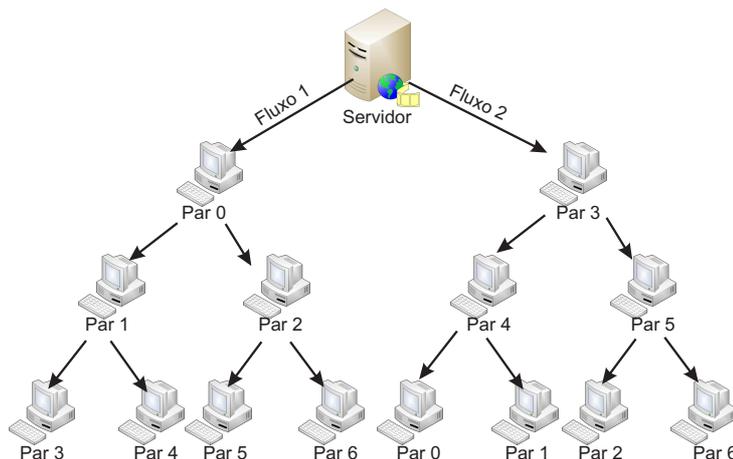


Figura 2.10: Árvore múltipla para fluxo de vídeo em P2P.

Na Figura 2.10 é apresentada uma árvore múltipla com dois subfluxos e sete pares. Observa-se que os pares folhas na árvore da esquerda (pares 3, 4 e 5) são pares internos na árvore da direita, e vice-versa (pares 0, 1 e 2). Assim, todos os pares contribuem com *upload*, com exceção do par 6, de fato, para uma árvore múltipla formada por dois subfluxos e sete pares, são necessários apenas seis pares fazendo *upload* para atender todos os pares da árvore, uma vez que o servidor também contribui com *upload*.

### 2.3.2 Distribuição de Fluxo de Vídeo Baseada em Malha

A Figura 2.11 ilustra a arquitetura de um sistema de distribuição de conteúdo baseada em malha. Um par conecta-se e desconecta-se com seus vizinhos dinamicamente, além de poder baixar e enviar conteúdo de/para diversos pares ao mesmo tempo. Assim, quando um par abandona a rede, os demais podem continuar acessando o conteúdo dos pares que ainda continuam conectados, sem que haja desconexão na rede, como acontece na distribuição baseada em árvores.

A manutenção nessa topologia de rede é realizada de forma centralizada, similar as redes do tipo *BitTorrent*. Quando um novo par se junta à rede que está assistindo a uma sessão de vídeo, esse envia ao *tracker* suas informações, isto é, endereço IP e o número de

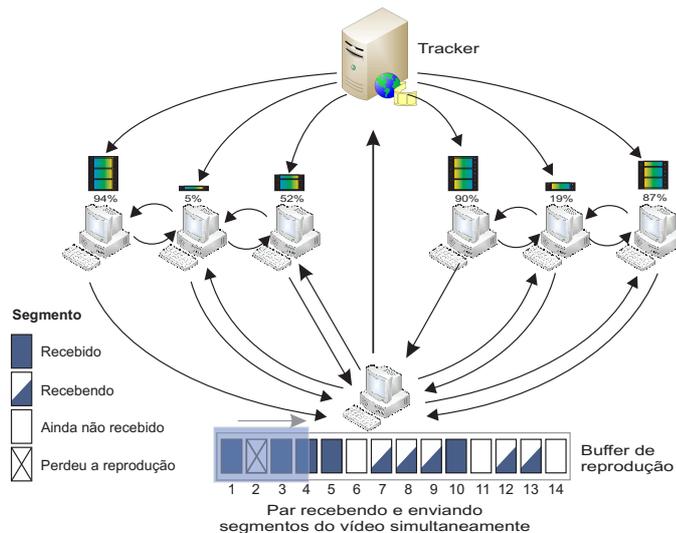


Figura 2.11: Distribuição de conteúdo por malha.

sua porta de comunicação. O *tracker*, por sua vez, devolve uma lista de endereços de pares que estão assistindo a mesma sessão. Em seguida, o novo par tenta estabelecer um certo número de conexões com os pares nessa lista, para então iniciar a troca de fluxo de vídeo com seus vizinhos.

Para lidar com as chegadas e as saídas dos pares no sistema, ou seja, o *churn*, cada par atualiza constantemente a sua lista de pares ativos através de solicitações feitas ao *tracker*, ou através de troca de lista de pares ativos com seus vizinhos. Quando um par abandona a rede normalmente, esse informa sua saída ao *tracker* e aos seus vizinhos, de forma que seu endereço seja removido da lista de pares ativos. Para lidar com as saídas inesperadas, os pares trocam entre si mensagens de manutenção e, assim, caso algum par deixe de responder por um período de tempo pré-configurado, esse é removido da lista de pares ativos do par que enviou a mensagem de manutenção.

O vídeo distribuído é dividido em pequenos segmentos, ver Figura 2.11, e os pares que vão recebendo os segmentos do vídeo vão compartilhando com os demais pares que também solicitam o vídeo. Para maximizar o uso da banda de *upload* dos pares e, conseqüentemente, melhorar a taxa de transmissão, a diversidade de segmentos nos pares precisa ser levada em conta. Assim, os pares vão recebendo os segmentos aleatoriamente e vão ordenando para reprodução. Os segmentos mais próximos do ponto de reprodução são priorizados na distribuição, no entanto, em VoD, diferentes pares podem assistir a partes diferentes do mesmo vídeo e, assim, pode ocorrer de o segmento do ponto de reprodução não estar disponível em nenhum dos pares no momento da reprodução (ANNAPUREDDY et al., 2007).

### 2.3.3 Distribuição de Fluxo de Vídeo Baseada em Semântica de Conteúdo

O processo de construção de uma rede P2P de compartilhamento de arquivos ocorre de maneira aleatória e por isso pode isolar pares com interesses similares. Esse isolamento restringe a eficácia das consultas que possuem um tempo de vida útil (*Time to live - TTL*) fazendo com que essas não sejam bem sucedidas, mesmo que, o conteúdo pesquisado esteja na rede. Além disso, em uma rede formada de maneira aleatória, as consultas são enviadas para todos os pares vizinhos, ocasionando o aumento do volume de dados retornado ao par que a submeteu e, assim, aumentando o tempo de resposta.

Crespo e Garcia-Molina (2005) estudaram a criação de redes sobrepostas com base na semântica de seus conteúdos. Nesse estudo, avaliou-se o compartilhamento de músicas, onde comunidades seriam criadas de acordo com os estilos musicais (Figura 2.12). Um par pode participar de mais de uma comunidade ao mesmo tempo, como é o caso dos pares *C*, *E* e *F* na Figura 2.12. Assim, o par *B* que compartilha músicas do estilo *rock* participa da rede formada por outros pares com o mesmo estilo. Quando qualquer um dos pares da rede *rock* realizar uma busca por uma música, do estilo *rock*, a chance deste conteúdo ser encontrado é maior e o tempo de resposta é menor, devido a consulta ser direcionada à comunidade relacionada ao conteúdo de interesse.

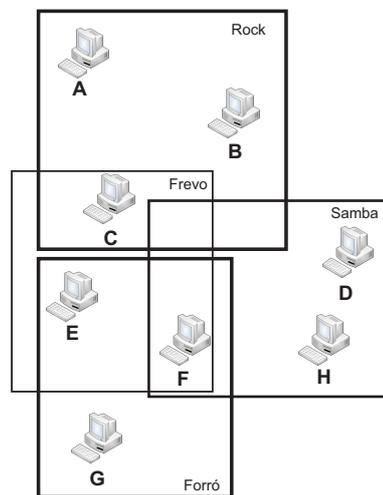


Figura 2.12: Redes sobrepostas semânticas.

Em Xia et al. (2008) é proposto um modelo (Figura 2.13) para construção de redes sobrepostas com base na semântica dos campos. Na Figura 2.13 é apresentada a estrutura da rede e a hierarquia dos pares. A hierarquia de um par está dividida em três camadas. Na camada mais inferior estão as fontes de informações que contém todos os recursos armazenados no par, como o sistema de arquivo local e o banco de dados. Na segunda

camada (Repositório Local) estão os metadados das fontes que possibilitam realizar consulta semântica e roteamento. A camada superior (Interface de Usuário) é a camada de interação com o usuário, é nessa camada que as consultas são transformadas em consultas semânticas padrão e encaminhadas aos pares correspondentes. Todo processo das camadas inferiores acontece de maneira transparente ao usuário. Pares com recursos semânticos idênticos são conectados formando linhas equipotenciais, e dessa forma, os recursos com semânticas idênticas são agrupados.

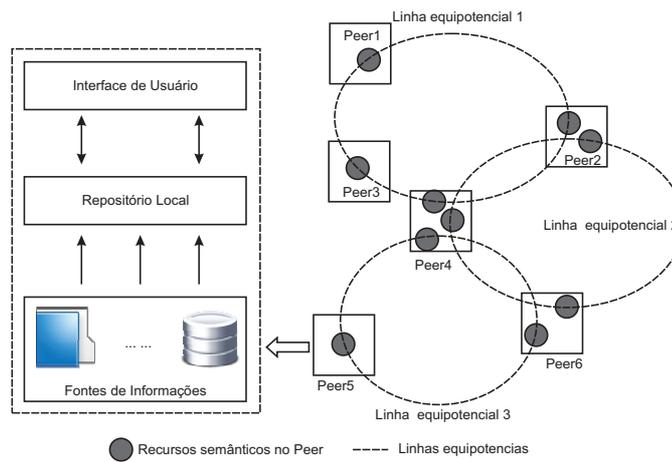


Figura 2.13: Modelos de redes sobrepostas semânticas baseadas em campos.

Neste modelo, uma consulta é encaminhada propositalmente para as linhas equipotenciais idênticas ou semelhantes, de acordo com a semântica da consulta. O estudo de Xia et al. (2008) comprovou que, o processo de roteamento foi mais seletivo e os encaminhamentos foram mais rápidos do que em consultas tradicionais.

## 2.4 Arquitetura Híbrida (CDN-P2P)

Aplicações de distribuição de conteúdo multimídia baseadas nas arquiteturas CDN e P2P têm alcançado relativo sucesso na Internet. No entanto, o uso dessas arquiteturas impõe limitações às aplicações (XU et al., 2006). Servidores CDN têm custo financeiro elevado para serem implantados e mantidos, conseqüentemente trazem custo para os provedores de mídia. Por outro lado, aplicações baseadas em P2P requerem um número significativo de pares para alavancar o processo de distribuição. Em comparação com servidores CDN, um par geralmente oferece banda de *upload* muito menor, exigindo, portanto, vários pares para transmitir com boa QoE uma mídia a um par solicitante. Sendo assim, nenhuma das arquiteturas (CDN e P2P), por si só, fornece uma solução eficiente e escalável para distribuição de fluxo de mídia na Internet (JIANG et al., 2008).

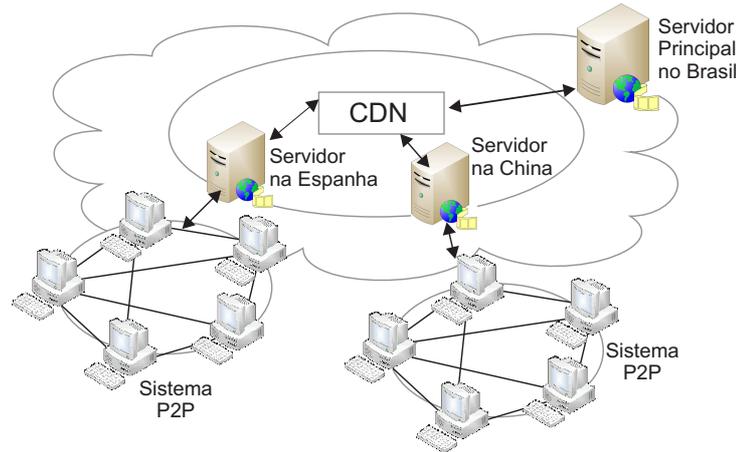


Figura 2.14: Arquitetura híbrida CDN-P2P.

Em Xu et al. (2006) é proposta uma arquitetura híbrida que integra distribuição de fluxo de vídeo por CDN com abordagens P2P. Nessa arquitetura as duas tecnologias se complementam. Quando uma mídia precisa ser distribuída para uma comunidade de pares, o arquivo é primeiramente distribuído para o servidor CDN e uma porção da capacidade de armazenamento do servidor é reservada para esse arquivo.

Quando um par requisita o arquivo, esse se torna uma semente. Juntos, o servidor e os pares sementes servem o fluxo de mídia com maior capacidade do que o servidor sozinho. Quando o sistema P2P possui uma quantidade de pares sementes que conseguem atender aos pedidos a determinando arquivo, esses assumem as requisições ao arquivo e, assim, diminuem a carga de trabalho que chega ao servidor. Quando a aplicação P2P assume as requisições a esse arquivo, a capacidade de armazenamento ocupada pelo arquivo no servidor CDN é liberada, economizando assim custo para o provedor. Na Figura 2.14 é ilustrada a arquitetura híbrida CDN-P2P, onde a CDN é assistida por aplicação P2P em nível de países. Comunidades P2P formam-se em torno dos servidores substitutos auxiliando esses com recursos, por exemplo, banda de *upload*, armazenamento e processamento.

## 2.5 Considerações Finais do Capítulo 2

Este capítulo apresentou as arquiteturas de distribuição de VoD na Internet. Inicialmente foi exposta a motivação de se estudar a transmissão VoD, em seguida caracterizaram-se os tipos de fluxos de vídeos distribuídos na Internet. Foram discutidas as arquiteturas cliente-servidor, P2P e a arquitetura híbrida (CDN-P2P).

A arquitetura híbrida combinada com outras novas tecnologias é vista como promiss-

---

sora para as redes futuras e a próxima geração da Internet (PAUL; PAN; JAIN, 2011). Para essa arquitetura ainda se encontram diversos problemas a serem estudados, dentre eles pode-se citar: *i*) a caracterização do impacto do *churn* no desempenho de políticas que gerenciam os objetos armazenados na parte P2P do sistema; e *ii*) a gerência de objetos armazenados nos pares, que são tratados nesta dissertação no Capítulo 5 e Capítulo 6, respectivamente.

## CAPÍTULO 3

# POLÍTICAS DE GERENCIAMENTO DE OBJETOS PARA SISTEMAS DE DISTRIBUIÇÃO DE CONTEÚDO

Políticas de gerenciamento de objetos foram bastante pesquisadas quando do surgimento dos primeiros sistemas *web*, tendo sido usadas principalmente em sistemas baseados em *proxy*. O surgimento de sistemas dinâmicos, no contexto da *web* 2.0, impõe novas demandas de gerenciamento de conteúdo que as políticas tradicionalmente usadas não atendem com mesmo desempenho de outrora. Em face a esse novo cenário, faz-se necessário a proposição de novas formas de gerenciar o conteúdo desses sistemas.

Neste capítulo é apresentado o estado da arte em políticas de gerenciamento de objetos. Nas seções seguintes discutem-se as principais classes de políticas de gerenciamento de objetos, que são baseadas: no tempo em que o objeto foi acessado, na frequência de acesso, nos custos dos objetos e em contexto e orientada a conteúdo.

### 3.1 Políticas Baseadas na Localidade Temporal

As políticas dessa classe baseiam-se na localidade temporal para decidir quais objetos devem ser mantidos na cache. A principal representante dessa classe é a política “menos recentemente usado”, do inglês *Least-Recently Used (LRU)*. A LRU mantém na cache objetos mais recentemente acessados e considera como estratégia que o acesso repetido aos objetos ocorre em uma escala de tempo curta, implicando que os objetos acessados recentemente têm alta probabilidade de serem acessados novamente. Colocando de outra forma, objetos que apresentam idade avançada, isto é, estão na cache bastante tempo,

possuem baixa probabilidade de voltarem a ser acessados. Tal observação caracteriza o conceito de localidade temporal verificada em fluxos de acesso estudados para a definição da política LRU.

Na Figura 3.1 apresenta-se a dinâmica da política LRU. Quando o espaço de armazenamento da cache estiver cheio, a política elimina o objeto que está na cache há mais tempo. Observa-se na Figura 3.1 que novos objetos são inseridos na cache e qualquer objeto acessado a partir da cache tem sua idade redefinida. Assim, objetos que vão envelhecendo na cache são candidatos à remoção.

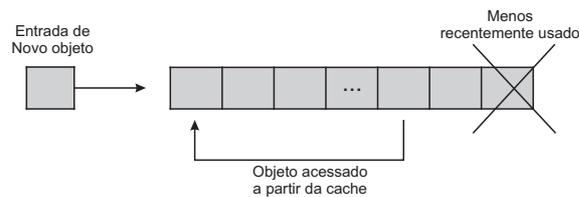


Figura 3.1: Dinâmica da política LRU.

A LRU tem sido utilizada em vários contextos, por exemplo, hierarquia de memória em sistema operacional e cache de páginas da *World Wide Web*. Seu desempenho é ruim, especialmente em termos de taxa de acerto de *byte*, quando os objetos gerenciados têm tamanho variado. A LRU não considera o tamanho dos objetos na dinâmica de substituição. Por outro lado, ela é adaptável às mudanças de carga de trabalho. Diversas outras políticas foram criadas a partir da LRU, entre elas destacam-se:

- *LRU-Threshold* (ABRAMS et al., 1995): Nessa estratégia objetos que o tamanho ultrapassa determinado limiar  $T$  não são armazenados na cache, os demais são gerenciados pela política LRU.
- *LRU-Min* (ABRAMS et al., 1995): Essa política tenta minimizar o número de substituição de objetos. Dado  $L$ , uma lista de objetos, e  $T$ , um tamanho limite. (1) Defina o tamanho  $T$  para  $S$ , onde  $S$  é o tamanho do objeto solicitado. (2) Defina  $L \forall S \geq T$ . (3) Aplicar a LRU até que  $L$  esteja vazia ou o espaço livre na cache seja no mínimo  $T$ . (4) Se o espaço livre na cache não for pelo menos  $S$  defina  $T = \frac{T}{2}$  e volte para o passo (2).
- *Pitkow/Recker strategy* (PITKOW; RECKER, 1994): Utiliza a política LRU, exceto para os documentos que foram acessados no mesmo dia, neste caso remove o maior deles.

Em Podlipnig e Böszörmenyi (2003) e em Wang (1999) há uma ampla descrição de outras políticas que utilizam o conceito de localidade temporal como estratégia de cache.

## 3.2 Políticas Baseadas na Frequência de Acesso

As políticas baseadas na frequência levam em consideração o número de acessos de cada objeto, mantendo os objetos mais acessados na cache. Essa estratégia considera que os objetos com maior frequência de acesso têm maior probabilidade de serem acessados em uma escala de tempo pequena.

A política “menos frequentemente usado”, do inglês *Least-Frequently Used (LFU)*, é a principal representante dessa classe. As requisições a cada objeto  $i$  são armazenadas em um contador  $f_i$ . Os objetos são ordenados pelo valor de  $f_i$  (Figura 3.2), quando um novo objeto for requisitado e o espaço de armazenamento da cache não for suficiente para armazenar o novo objeto, será removido o objeto com a menor frequência de acesso.

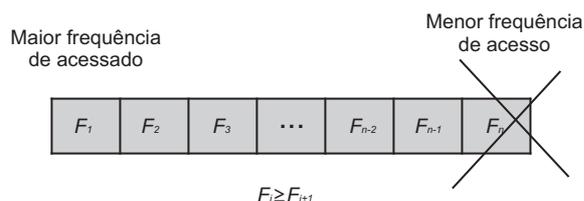


Figura 3.2: Dinâmica da política LFU.

A LFU tem duas versões: *i*) LFU perfeita, que conta os acessos dos objetos que estão ou já estiveram na cache, *ii*) LFU em cache, que conta os acessos somente dos objetos que estão na cache. A principal desvantagem da LFU é que objetos com elevado número de acesso permanecem na cache, mesmo que, esse objeto não seja mais alvo de novos acessos. Assim, como as políticas baseadas em localidade temporal, as políticas baseadas em frequência também não consideram o tamanho dos objetos, por isso, têm desempenho ruim em termos de taxa de acerto de *byte* quando o tamanho dos objetos apresenta alta variabilidade. Diversas variações da política LFU foram implementadas, e a seguir são apresentadas duas dessas variações.

- *LFU-Aging* (ARLITT; FRIEDRICH; JIN, 2000): Em cenários onde as requisições para um objeto são concentradas somente em certo período de tempo, a política LFU apresenta uma performance ruim. Isso ocorre devido a LFU manter na cache objetos com frequência de acesso alta, mesmo que esses objetos não estejam mais sendo acessados. Para minimizar essa poluição da cache, causada pelos obje-

tos com frequência alta de acesso, a política LFU-*Aging* adota uma estratégia de envelhecimento do objeto. É estabelecido um limiar superior  $A_{Max}$  para a média dos contadores de frequência de todos os objetos. Quando a média de todos os contadores de frequência ultrapassa esse limiar, tais contadores são reduzidos pela metade. Essa política utiliza dois parâmetros:  $A_{Max}$  que é um limiar superior para a média dos contadores de todos os objetos na cache e  $A_{Refs}$  que é um limiar superior para o contador de frequência de cada objeto.

- LFU-DA (ARLITT et al., 2000): O desempenho da política LFU-*Aging* depende da escolha dos parâmetros  $A_{Max}$  e  $A_{Refs}$ . Para tentar resolver esse problema a política LFU-DA (*LFU-Dynamic Aging*) recalcula dinamicamente a cada requisição de um objeto  $i$  um novo valor para a chave de prioridade  $k_i$  usando a equação  $k_i = f_i + L$ . Onde:
  - $L$  é o fator de envelhecimento que inicia com 0 e é atualizado a cada substituição ou exclusão de objetos.
  - $f_i$  é o contador de frequência do objeto  $i$ .

A exclusão dos objetos ocorre pelo menor valor  $k$ .

Em Podlipnig e Böszörmenyi (2003) e em Wang (1999) outras implementações baseadas na política LFU são apresentadas.

### 3.3 Políticas Baseadas em Custo

Nas políticas baseadas em custo a importância de um objeto é definida por uma função de valoração. Essa valoração pode estar associada a diferentes fatores, tais como, tempo desde o último acesso, tempo de entrada do objeto na cache, custo de transferência do objeto e tamanho.

A política GD (*Greedy-Dual*), originalmente proposta por Young (1994), representa a essência da estratégia baseada em custos. A estratégia baseada em custo lida com casos em que objetos na cache têm o mesmo tamanho, porém, os custos para a recuperação desses objetos são diferentes. O algoritmo calcula o custo  $k_i$  de cada objeto  $i$ , que é definido pelos recursos investidos para trazer esse objeto para a cache. Os objetos são ordenados pelo seu custo  $k_i$  (Figura 3.3) e quando for preciso substituir objetos na cache, aquele que possui o menor custo  $k_i$  é removido e todos os demais objetos terão seu custo

$k_i$  reduzidos pelo valor do custo do objeto removido. Se  $i$  for acessado em outro momento, seu custo  $k_i$  será restaurado para o valor inicial.

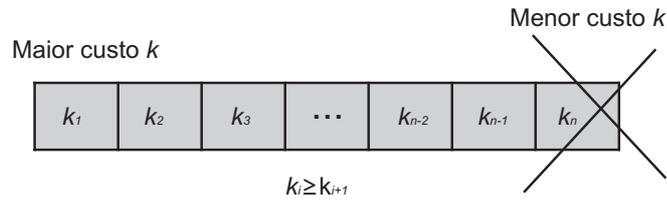


Figura 3.3: Dinâmica da política GD.

A seguir são apresentadas quatro políticas definidas a partir das ideias usadas pela GD.

- GDS (*GreedyDual-Size*) (CAO; IRANI, 1997): é uma variação da política GD, também ordena os objetos por um valor  $k_i$ , que é o custo do objeto calculado pela equação  $k_i = L + \frac{c_i}{s_i}$ .

Onde:

- $L$  é o fator de envelhecimento similar ao da *LFU-Aging* que inicia com 0 e é atualizado a cada substituição ou exclusão de objetos.
- $c_i$  é o custo de transferência do objeto
- $s_i$  é o tamanho do objeto

A substituição de objetos na cache ocorre pelos objetos com menor valor de  $k_i$ .

- GDSF (*GreedyDual-Size with Frequency*) (CHERKASOVA, 1998) : a política GDS não leva em consideração a frequência com que um objeto é acessado. Em um cenário onde dois objetos  $i_1$  e  $i_2$ , têm o mesmo custo, sendo que,  $i_1$  foi acessado  $n$  vezes e  $i_2$  foi acessado apenas uma vez, no pior caso,  $i_1$  pode ser removido, mesmo possuindo a maior frequência de acesso. A política GDSF utiliza um contador de frequência  $f_i$  para resolver esse problema, sendo  $k_i$  dado pela equação  $k_i = f_i \times \frac{c_i}{s_i} + L$ .
- GDSP (*GDS-Popularity*) (JIN; BESTAVROS, 2000): similar a GDSF, incorpora a frequência de acesso  $f_i$  para calcular o custo  $k_i$  do objeto. O valor de  $k_i$  é calculado pela equação  $k_i = f_i \times \frac{c_i}{s_i} + L$ . O que diferencia a política GDSP da GDSF é a manutenção na cache de informações de frequência de acesso de um dado objeto mesmo depois que esse tenha sido excluído. Ou seja, reserva-se um espaço da cache para manter histórico dos acessos de objetos removidos e, assim, quando algum desses objetos for acessado novamente recupera-se do histórico a frequência de acesso

do mesmo. Isso permite reduzir o impacto do ineditismo de um objeto no cálculo de  $k_i$ .

- PARTIAL (*Proportional Partial Caching Algorithm for P2P traffic*) (HEFEEDA; SALEH, 2008): gerencia objetos realizando armazenamento e exclusão parcial dos mesmos, face à dinâmica de conexão e desconexão, uma constante, em sistemas P2P. O algoritmo aumenta gradativamente a fração a ser armazenada de acordo com a popularidade do objeto. Os objetos são classificados na cache de acordo com o valor de  $\gamma_i$ , que é o número médio de vezes que cada *byte* do objeto  $i$  foi servido a partir da cache. As exclusões são realizadas parcialmente a partir de segmentos dos objetos de menor valor de  $\gamma$ .

### 3.4 Políticas Baseadas em Contexto e Orientada a Conteúdo

O conteúdo gerado por usuários, do inglês *User Generated Content (UGC)*, no que se convencionou chamar de *web 2.0*, tem criado novos padrões de visualizações e interação social (CHA et al., 2009) e tem sido utilizado para a definição de políticas de gerenciamento e distribuição de conteúdo que são sensíveis ao seu contexto. Como ilustrado na Figura 3.4, políticas baseadas em contexto e orientadas a conteúdo adicionam algum fator de custo associado ao contexto ou conteúdo no cálculo do custo  $k_i$  de cada objeto.

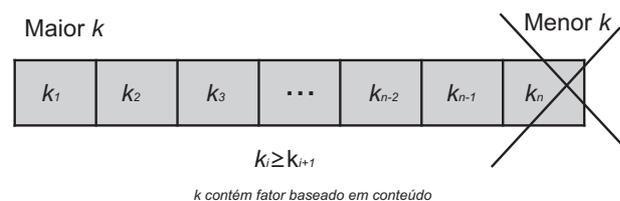


Figura 3.4: Dinâmica de políticas baseadas em contexto e orientadas a conteúdo.

Cheng e Liu (2009) exploraram uma estratégia de cache proativa, baseada na rede social existente entre os vídeos do site YouTube, para um sistema P2P de compartilhamento de vídeos de curta duração. Nessa política, usam-se as indicações do sistema de recomendação do YouTube, que sempre apresenta uma lista de vídeos relacionados a um vídeo acessado, para proativamente armazenar parte do conteúdo dos vídeos dessa lista. A escolha proativa dos vídeos é baseada na probabilidade destes serem acessados. Os autores mostraram que, o armazenamento proativo de 3 vídeos de uma lista de 10 relacionados pode alcançar uma taxa de acerto de 62%. Experimentos de simulação mostraram que

essa estratégia reduz bastante à carga nos servidores e melhora a qualidade de reprodução nos clientes, ao reduzir tempo de início da reprodução do vídeo.

Em Bezerra et al. (2009) considera-se que uma comunidade orientada a interesses possui informações que podem ser usadas para melhorar o desempenho de sistemas de cache em aplicações P2P. Nesse contexto, há uma alta probabilidade de um objeto acessado por um membro da comunidade ser acessado novamente por outros membros que estão na mesma comunidade. Bezerra et al. (2009) desenvolveram uma política de gerenciamento de objetos orientada a conteúdo e baseada em popularidade, a qual chamaram de POP, que classifica os vídeos em populares ou não populares, tendo como base o número de visualizações de um vídeo no site YouTube.

Em Kulkarni e Devetsikiotis (2010) foi proposto uma estratégia de cache que combina popularidade global de um vídeo com a popularidade dos vídeos relacionados a ele. Analisaram-se nas coleções de vídeos do YouTube as redes sociais criadas nos relacionamentos entre vídeos utilizando três métricas diferentes: grau de centralidade, proximidade do centro e centro intermediário. Os resultados mostraram que um mecanismo de cache distribuída baseado no método da centralidade reduz significativamente a carga na rede, mesmo para caches de tamanho moderado.

### 3.5 Considerações Finais do Capítulo 3

Neste capítulo as principais classes de políticas de gerenciamento de objetos foram apresentadas. Mostraram-se as políticas mais representativas de cada classe, tendo-se também abordado algumas variações de suas implementações.

Na literatura há diversas políticas de gerenciamento de conteúdo, além das citadas neste trabalho. Entretanto, acredita-se que uma nova geração de políticas de gerenciamento de conteúdo, especificamente de conteúdo multimídia, se faz necessária, face à riqueza de informações contidas nas propriedades do UGS, que são ignoradas pelas políticas tradicionais.

Identificou-se que sites baseados em UGC, entre eles os de VoD, tais como o YouTube, criam novos padrões de visualizações e interação social (CHA et al., 2009), tornando tal dinâmica um desafio para os algoritmos de gerenciamento de tais conteúdos. Dessa forma, surge necessidade de pesquisas que apontem soluções voltadas para esses conteúdos, uma vez que o fluxo de informações provindas desses sites gera elevado tráfego nas redes.

## CAPÍTULO 4

# AS COLEÇÕES DE VÍDEOS DO YOUTUBE

Neste capítulo discutem-se as coleções de vídeo utilizadas nos experimentos numéricos realizados nesta dissertação. Tais coleções são formadas por meta informações associadas aos vídeos do site YouTube. O estudo desses dados permite identificar estatísticas importantes a serem utilizadas na elaboração de políticas de gerenciamento de vídeos no sistema de distribuição. Assim, os estudos apresentados neste capítulo avaliam padrão de acesso e tempo de duração dos vídeos.

Ao longo desta pesquisa trabalhou-se com quatro coleções de vídeos do YouTube. No primeiro estudo (Capítulo 5), em que se avalia o impacto do *churn* nas políticas de gerenciamento de objetos, utilizou-se a coleção Ent06. Esta coleção foi obtida do YouTube no ano de 2006 por Cha et al. (2007) e encontra-se disponível na página dos autores <sup>1</sup>.

Para a realização do estudo apresentado no Capítulo 6, sobre políticas de gerenciamento de objetos em sistemas CDN-P2P, foram utilizadas três novas coleções (Ent11, Mus11 e Esp11). Estas coleções foram obtidas do YouTube em julho de 2011 a partir de um coletor desenvolvido pelo autor desta dissertação. Ambos, coletor e coleção, estão disponíveis para a comunidade mediante solicitação<sup>2</sup>.

### 4.1 Coleção Ent06

A coleção de vídeos Ent06 foi extraída do YouTube por Cha et al. (2007). Essa coleção foi utilizada no início desta pesquisa tendo-se usado as estatísticas coletadas nesta análise

<sup>1</sup>Coleção de vídeo Ent06 disponível em: <<http://an.kaist.ac.kr/traces/IMC2007.html>>

<sup>2</sup>e-mail: [liborio\\_filho@dcc.ufam.edu.br](mailto:liborio_filho@dcc.ufam.edu.br)

para avaliar o impacto do *churn* nas políticas de gerenciamento de objetos. Os vídeos dessa coleção pertencem à categoria entretenimento, tendo a coleção 1.687.506 registros, com as seguintes informações de um vídeo: duração, contagem de visualização, avaliações e qualificação. Na Figura 4.1 apresenta-se uma amostra dos registros dessa coleção de vídeos.

ID	Duração		Visual.	Aval.	Qual.
<b>vr3x_RRJdd4</b>	<b>3</b>	<b>39</b>	<b>8056321</b>	<b>2972</b>	<b>4.5</b>
<b>5bry10DU8bo</b>	<b>4</b>	<b>30</b>	<b>3793255</b>	<b>2527</b>	<b>3.5</b>
<b>o4x-VW_rCSE</b>	<b>1</b>	<b>14</b>	<b>3081102</b>	<b>8834</b>	<b>4.5</b>
<b>kyZRiEJnIag</b>	<b>0</b>	<b>32</b>	<b>3005105</b>	<b>1805</b>	<b>4.5</b>
<b>JGM5SdRve78</b>	<b>1</b>	<b>32</b>	<b>2091889</b>	<b>1893</b>	<b>4.5</b>
<b>MC8Zvl-8ziA</b>	<b>0</b>	<b>58</b>	<b>1948314</b>	<b>3883</b>	<b>4.5</b>

Figura 4.1: Registros da coleção de vídeos.

Cada entrada na coleção tem as seguintes informações:

1. Identificação: coluna 1, composta de 11 caracteres de 0-9, a-z, A-Z, - e \_;
2. Duração: coluna 2 e coluna 3, respectivamente minutos e segundos do tempo total do vídeo;
3. Visualizações: coluna 4, o número de vezes que o vídeo foi acessado;
4. Avaliações: coluna 5, a quantidade de avaliações que o vídeo obteve por parte dos usuários;
5. Qualificação: coluna 6, avaliações positivas atribuídas ao vídeo e varia de 0 a 5.

Para verificar como estão distribuídos os acessos na coleção, os vídeos foram agrupados em quatro conjuntos, ilustrados na Figura 4.2. No primeiro conjunto estão os vídeos que têm menos de 100 visualizações, que correspondem a 34,4% do total dos vídeos e 0,6% do total das visualizações. No segundo conjunto estão os vídeos que têm de 100 a 1.000 visualizações, e representam 40,5% de todos os vídeos e 7% das visualizações. No terceiro conjunto estão os vídeos que têm de 1.000 a 10.000 visualizações, correspondendo a 21,4% de todos os vídeos e 31,1% das visualizações. Finalmente, no quarto conjunto estão os vídeos que têm mais de 10.000 visualizações, esses representam 3,7% dos vídeos da coleção e possuem 61,3% do total das visualizações. A partir da análise representada na Figura 4.2,

percebe-se que mais de 60% dos acessos estão concentrados em uma pequena parte da coleção, menos de 4%, e que há predominância de muitos vídeos com baixa popularidade (conjunto 1 e 2).

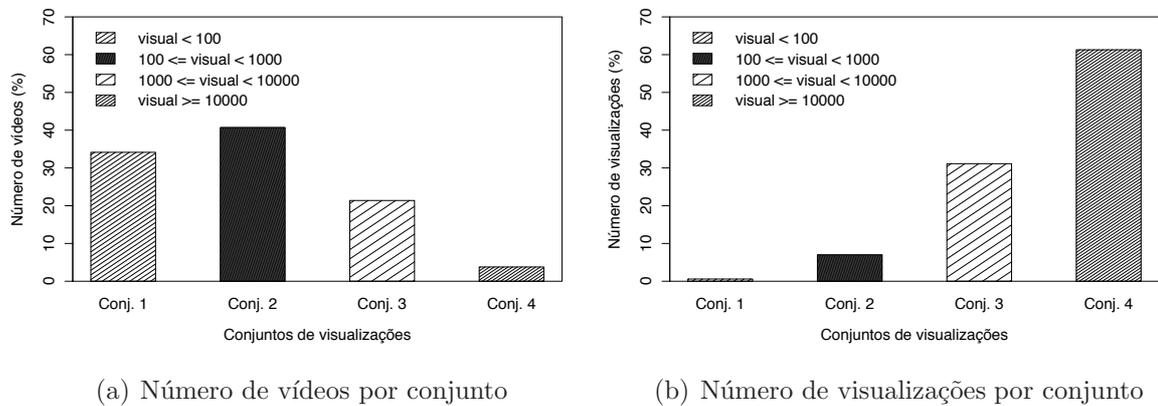


Figura 4.2: Distribuição dos vídeos por popularidade para a coleção Ent06.

Analisou-se também a variação na duração dos vídeos, ilustrada na Figura 4.3. Para essa coleção, há basicamente três picos de concentração de vídeos. A maior concentração é de vídeos de até 60 segundos. A segunda concentração está em torno de 180 segundos e a última em torno de 600 segundos, esta última, é devido ao limite de duração imposto pelo YouTube (até julho de 2010 limitava-se a duração dos vídeos em 10 minutos), o que incentivava os usuários a dividirem vídeos maiores em diversas partes de exatamente 10 minutos. Vídeos que aparecem com duração maior do que 600 segundos foram postados por usuários parceiros do YouTube, com permissão para postagem de vídeos de longa duração.

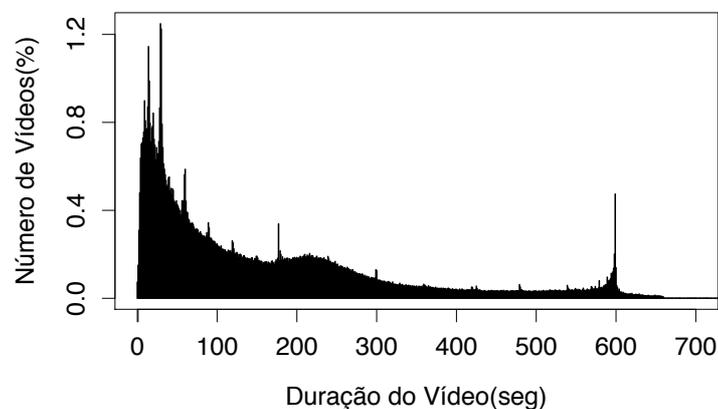


Figura 4.3: Histograma do tempo de duração dos vídeos.

Os vídeos dessa coleção têm a duração média de 133,3 segundos com desvio padrão de 124,1. Analisando esses dados, percebe-se que há certa variabilidade entre a duração dos vídeos dessa coleção, indicando que o tamanho de um vídeo pode ser uma medida importante a se avaliar para melhor gerência desses vídeos em sistemas de distribuição.

## 4.2 Coleções Ent11, Esp11 e Mus11

Nesta seção são analisadas as coleções de vídeos utilizadas nos estudos apresentados no Capítulo 6, sobre políticas de gerenciamento de objetos em sistemas CDN-P2P. O sistema utilizado para os estudos possui comunidades P2P criadas em torno de conteúdos, representados pelas categorias de vídeos do YouTube.

As coleções estudadas foram extraídas do YouTube no mês de julho de 2011, utilizando um coletor que implementa o método *snowball* (LEE; KIM; JEONG, 2006). Esse coletor utiliza a API de dados do YouTube (GOOGLE, 2012) que permite o acesso, por aplicações externas, ao conteúdo disponibilizado nas páginas do site YouTube.

---

**Algoritmo 1:** Método bola de neve utilizado para extrair metadados dos vídeos do Youtube.

---

**Entrada:**  $(S, P)$

**Saída:**  $G(M_i, R_i)$

$S$ : conjunto de vídeos coletados nos *feeds* padrões;

$P$ : profundidade máxima de busca;

$F$ : fila com vídeos descobertos;

$i$ : vídeo descoberto mas não explorado;

$r$ : vídeo relacionado a  $i$ ;

$M_i$ : metadados do vídeo  $i$ ;

$R_i$ : lista de vídeos relacionados a  $i$ ;

**para cada**  $i \in S$  **faça**

└ inserir  $i$  em  $F$ ;

**enquanto**  $F$  não vazia e  $P$  não seja atingida **faça**

└ obter cabeça de  $F$  :  $i$ ;

└  $G(M_i, R_i) \leftarrow M_i$ ;

└ **para cada**  $r$  não descoberto  $\in R_i$  **faça**

└└ inserir  $r$  em  $F$ ;

└ marcar  $i$  como explorado ;

└ remover  $i$  de  $F$ ;

---

O Algoritmo 1 apresenta o método bola de neve utilizado para extrair metadados de vídeos do YouTube. Cada vídeo obtido nos *feeds*, chamado de vídeo semente, é inserido em uma lista que é processada sequencialmente. Para cada vídeo dessa lista, um conjunto

de vídeos relacionados é coletado e adicionado à lista de vídeos para ser processado no futuro.

A coleção de vídeos do YouTube é explorada em largura até um certo nível ou até que todos os vídeos alcançáveis, a partir do conjunto de vídeos sementes, tenham sido visitados. Esse limite foi definido com base em uma análise prévia realizada em amostras das coleções de vídeos e com base na duração média das sessões de acesso ao YouTube (MONITORING, 2010). Os metadados extraídos para cada coleção são os mesmos já apresentados na Seção 4.1 com o acréscimo de mais uma informação que é a data de postagem do vídeo coluna 7 da Figura 4.4, pela qual se pode conhecer o tempo que o vídeo está disponível no site.

ID	Dur.	Visual.	Aval.	Qual.	Enviado
fAtuil0A6ek	3 56	14855	11355	4.20	2007-07-22T11:03:50.000Z
jwhT46ZpYeg	0 28	94601	63511	4.56	2007-06-06T04:48:12.000Z
6sJdpTMtzm4	4 12	30933	22431	5.00	2009-03-18T16:01:27.000Z
dzP0imTiV3U	7 38	97873	90112	5.00	2009-09-03T12:27:28.000Z
K03FVrCYUYw	3 42	85734	81425	5.00	2009-03-18T15:59:56.000Z
5xxuwzLDus	7 28	11020	10013	4.43	2008-04-02T10:38:25.000Z

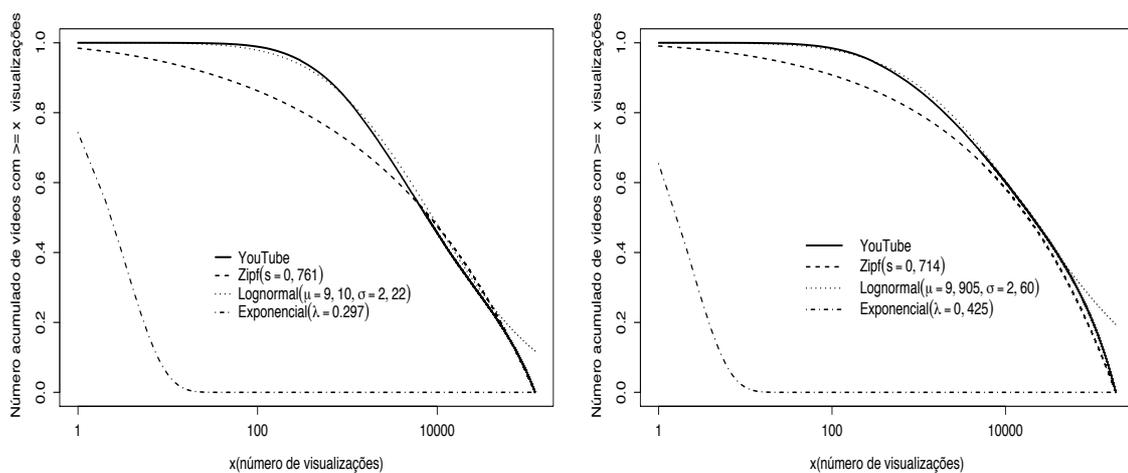
Figura 4.4: Metadado da coleção de vídeo.

Foram utilizados como semente para iniciar a coleta os *feeds* padrões do YouTube, que são: mais qualificados, mais incluídos entre os favoritos, mais vistos, mais populares, mais discutidos e mais respondidos. Tais *feeds* permitem filtrar as informações por categorias, países e escala de tempo (dia, semana, mês e todos os tempos). Para o contexto do sistema estudado foram obtidos metadados de vídeos das categorias esporte, entretenimento e música; localizados no Brasil para todas as escalas de tempo disponíveis.

Analisou-se cada coleção, a fim de que se pudesse entender o padrão de acesso registrado nos diversos vídeos coletados. Para isso, utilizou-se a técnica estatística de ajuste de modelos de regressão implementada em Yee (2008). Feito os ajustes dos modelos, foram representadas graficamente as visualizações versos a complementar do número acumulado de vídeos, o que permitiu analisar a distribuição dos acessos aos vídeos da coleção.

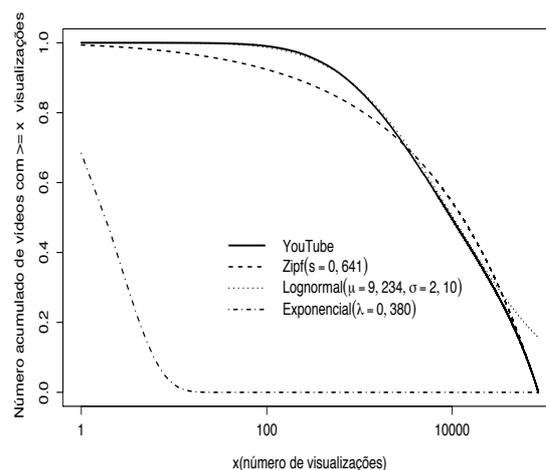
Na Figura 4.5 é apresentado o gráfico da distribuição dos acessos nas coleções e o ajuste com a função de distribuição acumulada complementar, do inglês *Complementary Cumulative Distribution Function (CCDF)*, para as distribuições Zipf, Lognormal e Exponencial. Para as três coleções, observa-se que o modelo Exponencial é o que menos

se ajusta aos acessos, pois menos de 80% dos vídeos possuem pelo menos uma visualização e nenhum vídeo chega a possuir 100 acessos, indicando que com esse modelo não há predominância de vídeos com alta popularidade. Para a distribuição Zipf observa-se um ajuste mais aproximado para os vídeos com alta popularidade, no entanto para os vídeos com poucos acessos percebe-se que as curvas não se ajustam com boa aderência. Para a distribuição Lognormal ocorre o melhor ajuste, porém percebe-se que com esse modelo quase 20% dos vídeos têm probabilidade de possuírem visualizações maior ou igual que a máxima visualização apresentada nas coleções do YouTube.



(a) Categoria entretenimento

(b) Categoria esporte



(c) Categoria música

Figura 4.5: Ajuste de curva para o padrão de acesso das coleções de vídeos.

Para verificar a aderência dos modelos analisados com o padrão de acesso do YouTube, utilizou-se o teste não paramétrico de Wilcoxon (BAUER, 1972) implementado no pacote estatístico da ferramenta R (R Development Core Team, 2010). O teste foi aplicado com

nível de significância de 5%, para o qual foi rejeitada a hipótese de que os acessos aos vídeos nessas coleções sejam modelados pelas distribuições analisadas.

A duração máxima dos vídeos aceita pelo YouTube era de 10 minutos até julho de 2010. A postagem de vídeos com duração maior que esse limite era permitida apenas para um grupo restrito de parceiros. A partir de julho de 2010, o YouTube passou a aceitar vídeos com até 15 minutos de duração, permitindo postagem de longa duração a qualquer usuário, bastando, para isso, a confirmação de gerência da conta através de mensagem de texto enviado ao celular do usuário.

Na Figura 4.6 é apresentado o histograma da duração dos vídeos para o intervalo de 1 a 1.000 segundos. Neste intervalo de tempo estão mais de 99% de todos os vídeos das três coleções analisadas. Para ser mais preciso, tem-se 99,0%, 99,8% e 99,8% do total dos vídeos das coleções entretenimento, esporte e música, respectivamente. Em análise realizada com 100% dos vídeos de cada coleção foi observado que as coleções apresentam poucos vídeos de longa duração; no entanto, há vídeos excessivamente longos. O mais longo da categoria música possui mais de 16 dias de duração; da categoria esporte possui mais de 12 dias e da categoria entretenimento atinge aproximadamente 2 dias de duração. Esses poucos vídeos com duração atípica (*outliers*) implicam prejuízos à interpretação dos resultados dos testes estatísticos aplicados às coleções. Para as estatísticas apresentadas na Tabela 4.1 utilizou-se técnica de remoção de valores *outliers*.

Tabela 4.1: Estatística da duração dos vídeos do YouTube.

Coleção	Quantidade de vídeos	Duração média (seg)	Desvio padrão (seg)	Coef. de variação
Entretenimento-2011	413524	224,7	182,8	0,81
Esporte-2011	440584	170,5	129,6	0,76
Música-2011	223564	221,5	85,2	0,38

Na Tabela 4.1 são apresentados dados estatísticos da duração dos vídeos de cada coleção. A duração média dos vídeos das três coleções varia de 170,5 a 224,7 segundos. Observando o desvio padrão e o coeficiente de variação, conclui-se que a coleção da categoria entretenimento apresenta a maior dispersão na duração dos vídeos, com coeficiente de variação 0,81, enquanto que, a categoria música é a menos dispersa, com coeficiente de variação 0,38. Ao analisar o histograma na Figura 4.6, percebe-se que a coleção música, de fato, possui maior concentração em torno da média, confirmando as conclusões que se tem a partir do desvio padrão e coeficiente de variação.

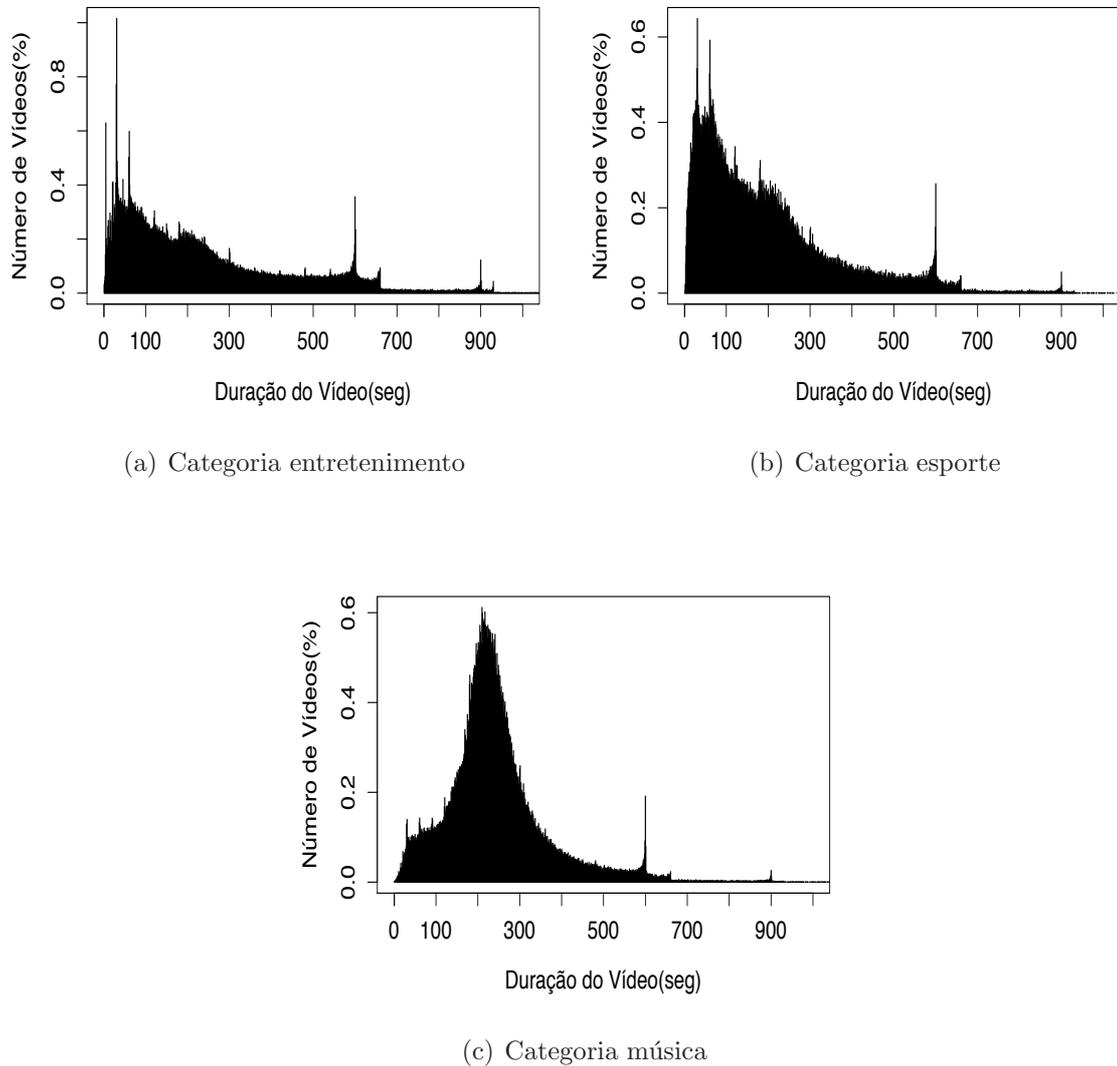


Figura 4.6: Histograma do tempo de duração dos vídeos.

As coleções entretenimento e esporte possuem maior concentração de vídeos até 120 segundos, representando uma taxa de 25,9% e 42,1% do total, respectivamente. Já a coleção música possui maior concentração em torno da média, 221,5 segundos, isso ocorre por haver, na categoria música, muitos vídeos com apenas uma música, e a maioria das músicas têm duração em torno desse valor. Em todas as coleções percebe-se pico em 600 e em 900 segundos, isso é devido ao limite de duração imposto pelo YouTube. Até julho de 2010 era 10 minutos e, desde então, passou-se a aceitar vídeos de 15 minutos.

## 4.3 Considerações Finais do Capítulo 4

Neste capítulo as coleções de vídeos utilizadas nesta pesquisa foram analisadas. Estudou-se o padrão de acesso das coleções do YouTube, a fim de reproduzi-lo nas simulações, assim como utilizar tais conhecimentos para oportunizar melhorias nos sistemas de distribuição de vídeos. Percebeu-se que o padrão de acesso de todas as coleções segue as leis da potência; porém, dos modelos de distribuições analisados, o que mais se ajustou foi a Lognormal, apesar de apresentar divergência para vídeos com alta popularidade, mantendo o conjunto desses vídeos relativamente grande em relação aos acessos do YouTube. Verificou-se a aderência dos modelos analisados com os acessos do YouTube utilizando o teste estatístico não paramétrico de Wilcoxon. O teste rejeitou a hipótese, com nível de significância de 5%, de os acessos registrados nas coleções seguirem os modelos de distribuição estatística analisados.

Para análise de duração dos vídeos identificou-se que em todas as coleções, há, de fato, a predominância de vídeos de curta duração, mais de 99% dos vídeos possuem até 1.000 segundos de duração. Nas coleções entretenimento e esporte há uma maior concentração nos primeiros minutos, com diminuição de acordo com o aumento do tempo. Já a coleção música possui um comportamento diferente, tendo maior concentração em torno de 221,5 segundos, que é a duração aproximada da maioria dos clips de músicas.

## CAPÍTULO 5

# AVALIAÇÃO DO IMPACTO DO *CHURN* NAS POLÍTICAS DE GERENCIAMENTO DE OBJETOS

Em sistemas P2P a disponibilidade de conteúdo está diretamente relacionada com o número de pares conectados simultaneamente, ou seja, o tamanho do sistema. Usuários de aplicações de VoD que são baseadas em sistemas P2P podem experimentar baixa qualidade de experiência (QoE) com a aplicação sofrendo longo atraso e frequente perda de continuidade quando tais sistemas têm tamanho reduzido.

Como apresentado no Capítulo 2, as CDNs foram desenvolvidas para lidar com a demanda de recursos dos provedores de conteúdo multimídia devido a sua alta disponibilidade e pequeno atraso. No entanto, a escalabilidade de tais redes e o seu alto custo de implantação têm dificultado seu uso em larga escala. Assim, sistemas híbridos que exploram as vantagens de ambos, CDN e P2P, têm sido implantados comercialmente (HUANG et al., 2008). Nesses sistemas, pares compartilham os seus recursos, por exemplo, capacidade de armazenamento e largura de banda, durante os períodos de conexão, diminuindo as exigências sobre a infraestrutura de servidores da CDN. Por outro lado, os servidores CDN garantem a disponibilidade de conteúdo quando a contribuição dos pares é limitada pela dinâmica típica do sistema P2P.

Entretanto, essa arquitetura por si não garante a efetiva distribuição do conteúdo sem mecanismos de gerenciamento efetivos. Uma questão a ser investigada é o impacto do *churn*, isto é, os ciclos de entrada e saída dos pares, sobre os mecanismos de gerência do conteúdo distribuído pelo sistema. Esse ciclo pode induzir a replicação descontrolada de dados nos pares reduzindo a capacidade efetiva de armazenamento do sistema.

As principais contribuições apresentadas neste capítulo são duas. Primeiro, avalia-se o aumento da capacidade individual de armazenamento dos pares como uma medida para manter o desempenho das políticas, uma vez que a capacidade de armazenamento do sistema é afetada diretamente pelo *churn*. Com isso espera-se que os pares sejam capazes de armazenar objetos de valor durante uma escala de tempo maior e, assim, melhorem a disponibilidade de conteúdo. Segundo, a replicação de dados é um componente chave para melhorar a disponibilidade do conteúdo em sistemas de distribuição. No entanto, encontrar o equilíbrio entre os dados replicados e não replicados é um importante parâmetro para projetar tais sistemas. Sendo assim, avalia-se o quanto a replicação de dados afeta o desempenho das políticas analisando-se um sistema com e sem replicação.

O restante deste capítulo está organizado da seguinte forma. A Seção 5.1 apresenta o sistema utilizado para distribuir o conteúdo. Na Seção 5.2 apresenta-se procedimento utilizado para reproduzir em simulação os acessos aos vídeos. Na Seção 5.3 são apresentados os modelos de *churn* utilizados para definir a dinâmica dos pares na parte P2P do sistema. Os resultados numéricos coletados a partir de experimentos são apresentados na Seção 5.4. Finalmente, na Seção 5.5 as considerações finais deste capítulo são apresentadas.

## 5.1 A Distribuição de Conteúdo

O sistema considerado nesta pesquisa é um sistema híbrido (CDN-P2P) de distribuição de vídeos sob demanda. Os vídeos distribuídos são de curta duração, tal como os vídeos do YouTube.

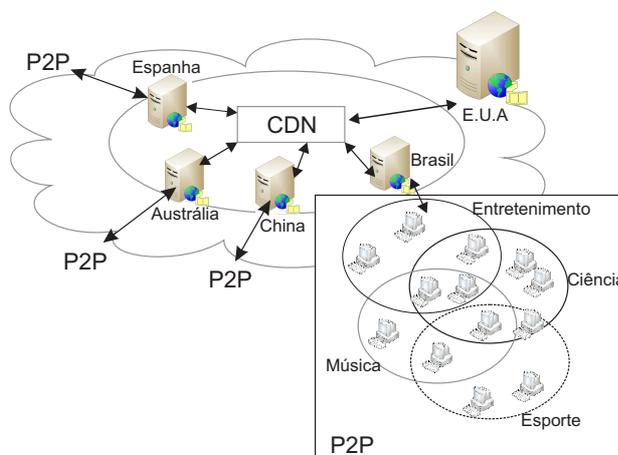


Figura 5.1: A infraestrutura do sistema de distribuição CDN-P2P.

Nesse sistema, os pares da rede P2P auxiliam os servidores da CDN respondendo às requisições, quando possível. Os pares são agrupados em comunidades semânticas que

são construídas em torno de assuntos, tais como, música, esporte e entretenimento, como ilustrado na Figura 5.1. Cada par pode participar de várias comunidades, dependendo do conteúdo acessado pelo mesmo. As redes sobrepostas formadas nesse sistema são baseadas na semântica dos conteúdos, tal como apresentado na Seção 2.3.3.

O sistema apresentado foi estudado utilizando um simulador de eventos discretos, cujo tem sua estrutura ilustrada na Figura 5.2. O arquivo contendo metadados da coleção de vídeos representa os objetos armazenados na CDN. Os pares da rede P2P ao entrarem no sistema realizam requisições, tais requisições são baseadas no padrão de acesso identificado na própria coleção utilizada. Os pares dispõem de certa capacidade de armazenamento reservada para manter alguns vídeos que são gerenciados por uma instância da política de gerenciamento implementada no par. Na rede P2P os pares apresentam uma dinâmica de conexão e desconexão que é caracterizada no simulador pelos modelos de *churn* implementados, isto é, exponencial e cauda longa.

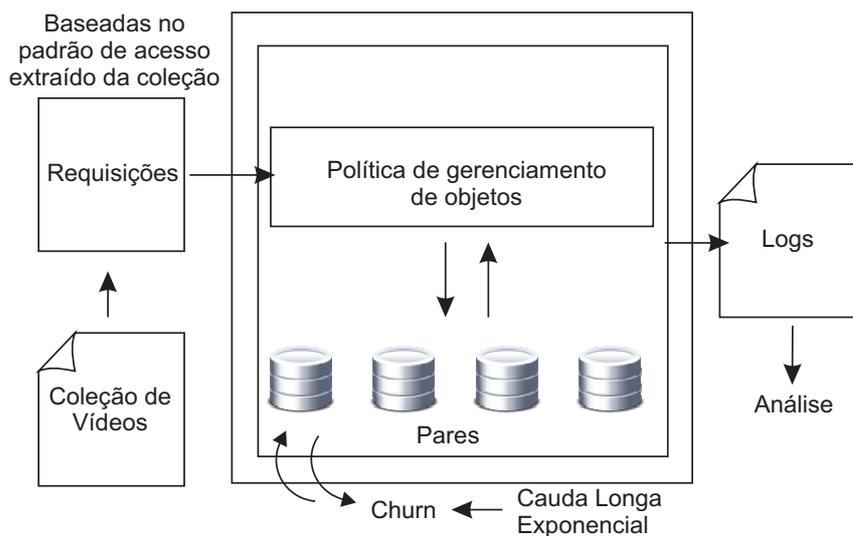


Figura 5.2: Estrutura geral do simulador.

O simulador mantém registro das operações realizadas no sistema. Em uma dada escala de tempo são coletadas tais informações e gravadas em *logs*, de onde são derivadas as análises apresentadas nesta dissertação.

## 5.2 Padrão de Acesso aos Vídeos

Na análise apresentada no Capítulo 4 observa-se que as distribuições estatísticas analisadas não modelam com precisão o padrão de acesso identificado na coleção. Portanto, para simular os acessos à coleção de vídeos no sistema, mantendo o mesmo padrão identificado na coleção, utilizou-se o seguinte procedimento:

1. Todas as requisições são guiadas pela combinação da duração do vídeo e do “tempo de avaliação”, ou seja, o tempo no qual o usuário decide qual vídeo irá assistir;
2. Os acessos foram reproduzidos a partir da função densidade de probabilidade extraída da coleção do YouTube utilizando o método de geração de número aleatório aceitação-rejeição (RIPLEY, 1987);

As suposições feitas nesse procedimento são que os vídeos acessados serão vistos por completos e os usuários irão despende algum tempo procurando vídeos relacionados antes de acessar um novo vídeo.

Os acessos gerados em simulação representam satisfatoriamente o padrão de visualização identificado na coleção do YouTube. O teste de aderência não paramétrico de Wilcoxon com nível de significância de 5% aceita a hipótese de que os valores gerados pela simulação (passo 2 do procedimento) provêm da mesma função de distribuição extraída das coleções do YouTube.

### 5.3 Modelos de *Churn*

Para entender o *churn* e seu impacto no desempenho das políticas de gerenciamento de objetos implementadas na porção P2P do sistema híbrido avaliado empregou-se o modelo de *churn* genérico definido em Yao et al. (2006).

Seja um sistema P2P com  $n$  pares, onde cada par  $i$  está conectado (*UP*) ou desconectado (*DOWN*) em um tempo  $t$ . Esse comportamento pode ser modelado por um processo  $Z_i(t)$ (5.1) para cada par  $i$ :

$$Z_i(t) = \begin{cases} 1 & \text{se o par } i \text{ está } UP \text{ no tempo } t \\ 0 & \text{caso contrário} \end{cases} \quad (5.1)$$

Para cada processo  $Z_i(t)$  a duração de uma sessão *UP*  $\{L_{i,c}\}_{c=1}^{\infty}$  tem uma distribuição associada  $F_i(x)$  e a duração de uma sessão *DOWN*  $\{D_{i,c}\}_{c=1}^{\infty}$  tem outra distribuição associada  $G_i(x)$ , onde  $c$  representa o número do ciclo. A duração das sessões *UP* e *DOWN* de um par  $i$  é dada pelas variáveis randômicas  $L_{i,c} > 0$  e  $D_{i,c} > 0$ , respectivamente. O *churn* do sistema P2P avaliado neste trabalho foi modelado pelas seguintes distribuições:

i) exponencial,

$$f(x; \lambda) = 1 - e^{-\lambda x}$$

com média  $\lambda^{-1}$ ; e *ii*) Pareto,

$$F(x; \beta, \alpha) = 1 - (1 + x/\beta)^\alpha, x > 0, \alpha > 1$$

com média  $\beta/\alpha - 1$ .

Com base no modelo acima, as seguintes suposições são feitas:

1. Para capturar a natureza independente dos pares, o conjunto  $\{Z_i(t)\}_{i=1}^n$  é composto por processos de nascimento e morte mutuamente independente. Assim, os pares se comportam de forma independente uns dos outros e os processos  $\{Z_i(t)\}$  e  $\{Z_j(t)\}$ , para qualquer  $i \neq j$ , são independentes;
2.  $l_i$  é a duração média de uma sessão *UP* e  $d_i$  é a duração média de uma sessão *DOWN*, ambas, são independentes para cada par.

Neste estudo foram considerados 28 minutos, tempo de duração média de uma sessão do YouTube (CHA et al., 2009), para definir  $l_i$ , e 14 minutos para definir  $d_i$ , portanto, espera-se uma comunidade de pares bastante ativa.

Quando o modelo de *churn* exponencial é aplicado, a duração de uma sessão *UP* é definida por uma distribuição exponencial, enquanto que a duração de uma sessão *DOWN* é definida por uma distribuição Pareto. A Figura 5.3 mostra a CCDF das distribuições exponencial (sessão *UP*) e Pareto (sessão *DOWN*) com parâmetros  $l_i$  e  $d_i$  definidos como mencionado anteriormente. Analisando a Figura 5.3(a), observa-se que no caso da distribuição exponencial, após 120 minutos a probabilidade de se escolher valores maiores do que os esperados desaparece. Por outro lado, na distribuição Pareto essa probabilidade ainda existe por uma escala de tempo maior. Assim, pode-se concluir que o padrão de *churn* exponencial expressa o seguinte cenário: os pares permanecem conectados quando entram na comunidade, mas ao longo do tempo vão perdendo o interesse.

No modelo de *churn* cauda longa, a duração das sessões *UP* e *DOWN* são definidas pela distribuição Pareto. Diferentemente do modelo de *churn* exponencial, o modelo de *churn* cauda longa ainda mantém uma probabilidade significativa ao longo do tempo, como mostrado na Figura 5.3(b), indicando que há probabilidade de pares permanecerem conectados por longo período. Assim sendo, esse modelo de *churn* expressa o seguinte cenário: os pares mantêm interesse no conteúdo por uma escala de tempo maior.

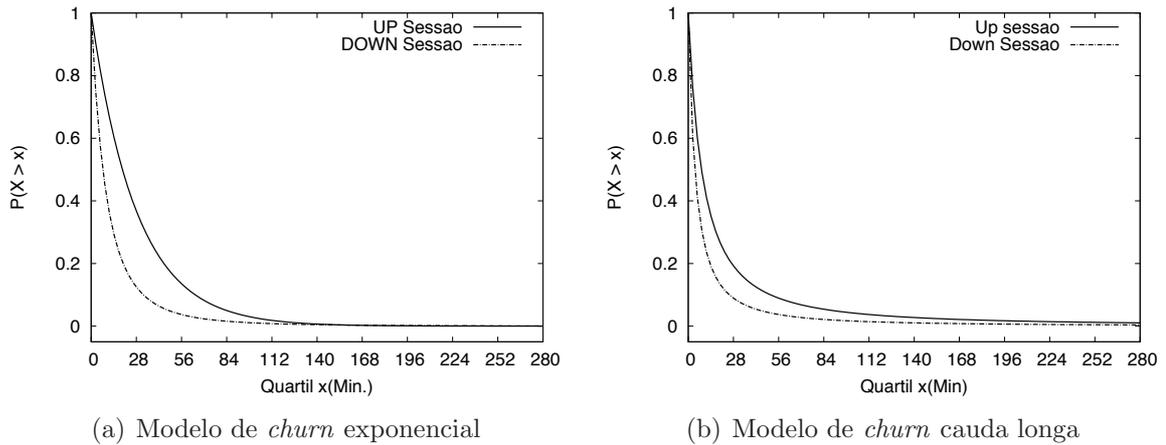


Figura 5.3: CCDF da duração das sessões de UP e DOWN para os modelos de *churn* exponencial e cauda longa.

## 5.4 Resultados Numéricos

Nesta seção avalia-se o impacto dos eventos de entrada e saída de pares, isto é, o *churn*, no desempenho das políticas de gerenciamento de objetos estabelecidas nos pares que auxiliam os servidores da CDN na tarefa de distribuição de conteúdo. Para isso, utilizou-se o sistema híbrido CDN-P2P de distribuição de vídeo (Seção 5.1), e a coleção de vídeo Ent06, analisada na Seção 4.1. Mediu-se a taxa de decréscimo de performance – o quanto o desempenho das políticas de gerenciamento é impactado pelo *churn*. Essa taxa de decréscimo é definida como segue:

$$TA_{Decr} = 1 - \frac{TA_{com}}{TA_{sem}} \quad (5.2)$$

onde  $TA_{com}$  e  $TA_{sem}$  são, respectivamente, a taxa de acerto coletada no sistema com e sem *churn*. Avaliou-se essa taxa de decréscimo considerando-se os dois modelos de *churn*, exponencial e cauda longa.

Como mencionado no Capítulo 3, o desempenho de uma política de gerenciamento de objeto pode ter valores diferentes dependendo da métrica utilizada, por exemplo, a política LRU pode ter uma boa performance pela métrica de taxa de acerto (*hit rate*), no entanto, pode ter baixa performance para a métrica taxa de acerto de *byte* (*byte hit rate*). Dessa forma, neste trabalho estudou-se tanto a taxa de acerto quanto a taxa de acerto de *byte* coletadas pelo sistema com e sem a presença de *churn*. Do ponto de vista quantitativo, as duas obtiveram taxa de decréscimo de performance similar. Assim, é apresentado somente o desempenho do sistema com relação à taxa de decréscimo de acerto, conforme definida pela Equação 5.2.

As cinco políticas de gerenciamento de objetos avaliadas neste estudo foram: POP, LRU, LFU, GDSP e PARTIAL, todas descritas no Capítulo 3. Para as simulações avaliadas neste capítulo, definiu-se a capacidade de armazenamento do sistema em função da duração total da coleção. Cada par tem capacidade  $c_j$ , sendo a capacidade de armazenamento total do sistema definida por:

$$C = \sum_{j=1}^N c_j \quad (5.3)$$

onde  $N$  é o número de pares do sistema. Dessa forma, para avaliar o desempenho das políticas em função do aumento de  $C$ , diferentes cenários de simulação foram definidos, aumentando o valor de  $C$  em até 20% da duração total da coleção.

Na Figura 5.4 apresenta-se o impacto do *churn* em termos da taxa de decréscimo utilizando-se os modelos, exponencial e cauda longa. A taxa de decréscimo, definida pela Equação 5.2, registrada para as políticas estudadas apresenta valores diferentes como descrito a seguir: as políticas sofrem menos impacto do *churn* à medida que o valor de  $C$  aumenta. No entanto, esse aumento não afeta linearmente o desempenho das políticas. Quando o tamanho do sistema é reduzido, ou seja 1%, as políticas GDSP, LFU e POP registram o maior impacto para o modelo de *churn* exponencial, com taxa de decréscimo de 54,7%, 45,3%, 43,7%, respectivamente. Para o modelo de *churn* cauda longa, sofrem maior impacto as políticas LFU e POP, com taxa de decréscimo de 36,4% e 35,2%, respectivamente.

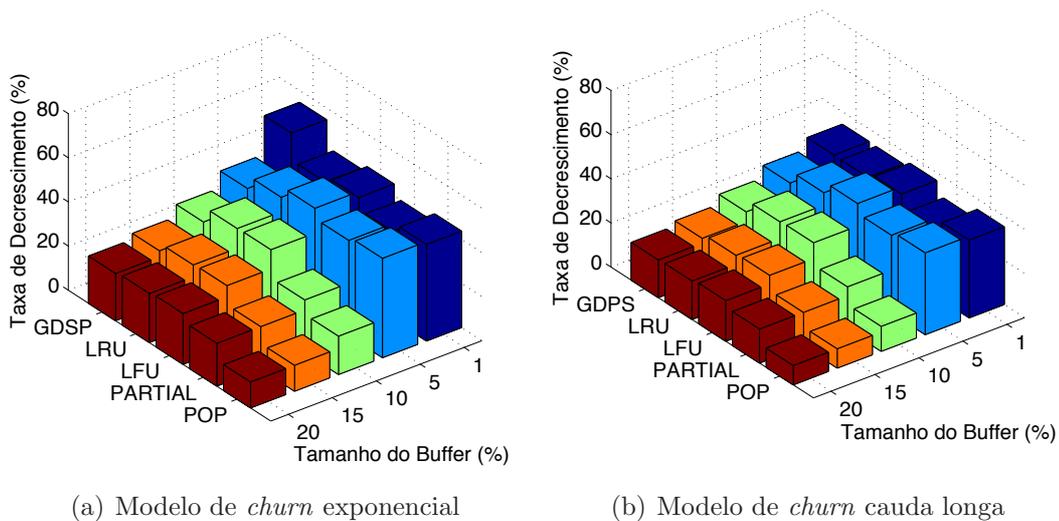


Figura 5.4: Taxa de decréscimo de acerto das políticas de gerenciamento de objetos.

Com o aumento de  $C$  (10%, 15% e 20%) percebe-se que a política POP se destaca

quanto a absorção do impacto do *churn* para ambos os modelos. Especificamente para o tamanho de  $C$  em 20% essa política registra a menor taxa de decréscimo, 11,7% e 8,7% para os modelos de *churn* exponencial e cauda longa, respectivamente. Por outro lado, as políticas LFU, LRU e GDSP sofrem maior impacto, registrando respectivamente taxa de decréscimo de 23,0%, 22,2% e 21,4% para o modelo exponencial e 18,4%, 17,4% e 17,1% para o modelo cauda longa. A quantidade de dados replicados é a principal razão para o pouco impacto sofrido pela política POP, quando o tamanho de  $C$  aumenta, como será mostrado nas avaliações a seguir.

O aumento da capacidade de armazenamento do sistema impacta o desempenho das políticas de gerenciamento de objetos, especificamente para o modelo de *churn* exponencial para o qual as variações nas taxas de decréscimo são mais evidentes. Para esse modelo de *churn*, como apresentado na Seção 5.3, os pares vão perdendo o interesse ao longo do tempo pelo conteúdo disponibilizado pelas comunidades e, por conseguinte, reduzindo o número de pares ativos na comunidade.

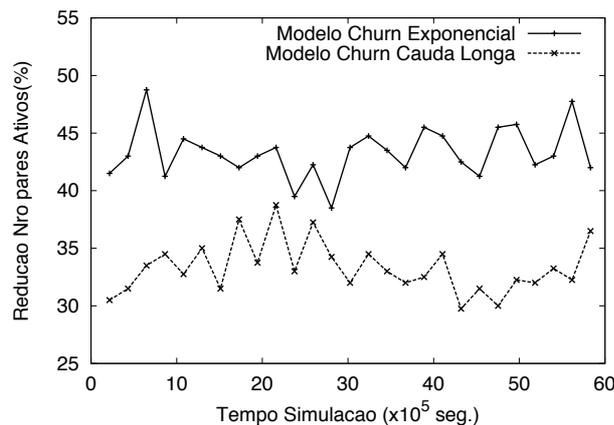


Figura 5.5: A diminuição do número de pares ativos.

Na Figura 5.5 apresenta-se a redução do tamanho do sistema, isto é, a diminuição do número de pares ativos, quando o *churn* do sistema é definido pelos modelos exponencial e cauda longa. Quando o *churn* é definido pelo modelo cauda longa o decréscimo máximo do tamanho do sistema é 39%, enquanto que para o modelo exponencial, esse valor é de 48%. Como observação geral de projeto, sabendo-se que suposições fortes a cerca da capacidade de armazenamento dos pares pode colocar em risco sua implementação, um sistema CDN-P2P deve desenvolver mecanismos para manter por uma escala de tempo maior, pares interessados no conteúdo disponibilizado pelas comunidades.

Cada vez que um par deixa o sistema, o conteúdo popular armazenado nesse torna-se indisponíveis para os demais pares. Quando o par retorna ao sistema, parte do conteúdo

armazenado em seu *buffer* pode ter sido solicitada por outros pares diretamente aos servidores da CDN. Essa dinâmica naturalmente replica dados entre os pares que compartilham interesses e, conseqüentemente, diminui a capacidade nominal de armazenamento do sistema.

Na Figura 5.6 apresenta-se a taxa de replicação de dados no sistema para os dois modelos de *churn* avaliados. Para as políticas LRU, LFU e GDSP, a maior quantidade de dados replicados demanda 20% da capacidade de armazenamento. Essa demanda para as duas outras políticas, PARTIAL e POP, é de aproximadamente 25% e acima de 45%, respectivamente. A quantidade de dados replicados gerida pela política POP se mostrou bem maior em comparação com as políticas LRU, LFU e GDSP. Ou seja, a política POP replicou os dados duas vezes mais que as outras políticas. Isso ocorre devido a POP manter na cache somente objetos com certo número de visualizações, não menos que 10.000, fazendo com que a quantidade de dados replicados cresça rapidamente em função do *churn*.

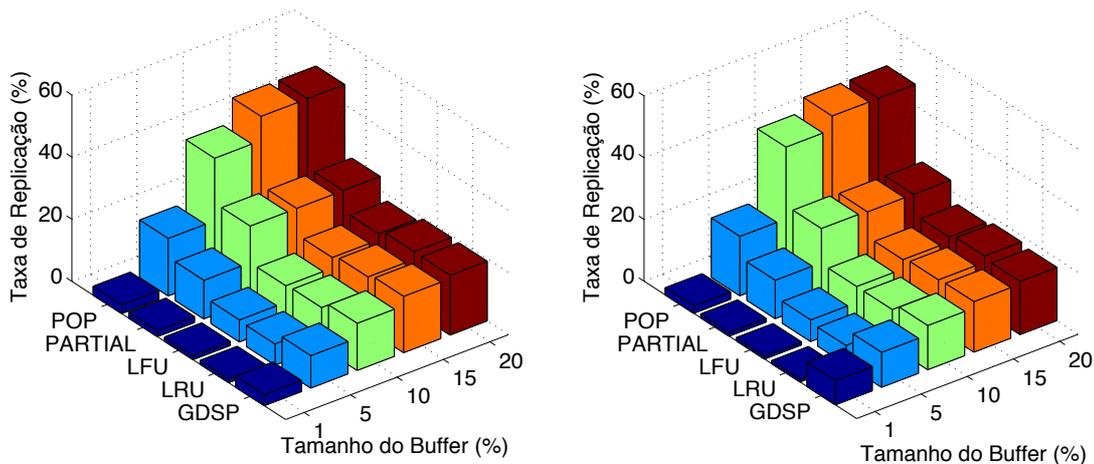
(a) Modelo de *churn* exponencial(b) Modelo de *churn* cauda longa

Figura 5.6: Quantidade de dados replicados manipulados pelas políticas.

Resiliência a falha é uma das principais características de sistemas P2P. Nesse contexto, quando um par falha na prestação de um serviço que está sob sua responsabilidade, outro par deve responder por esse mesmo serviço. Essa propriedade tem conseqüências no sistema de distribuição de vídeos proposto, isto é, as políticas avaliadas têm de lidar com a replicação de dados que ocorre de forma independente nos pares.

Nos estudos anteriores verificou-se que pelo menos 20% dos dados armazenados são réplicas, sendo esse valor superior a 45% quando a política POP é usada. A seguir mostram-se os resultados dos estudos realizados para avaliar o impacto da replicação no desempenho

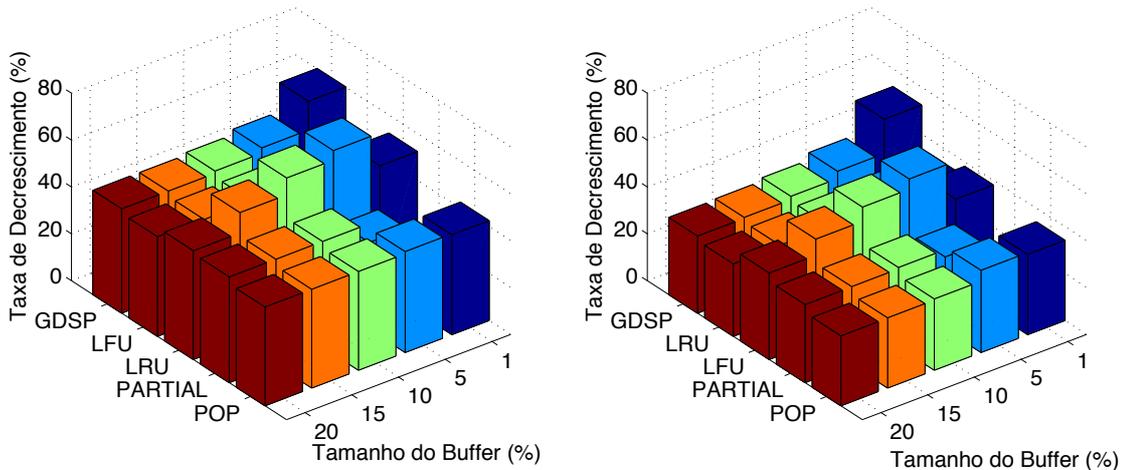
das políticas estudadas. Para avaliar tal impacto, foram realizados experimentos considerando o seguinte procedimento: os pares removem qualquer objeto replicado toda vez que retornam ao sistema.

Nesse novo cenário, a taxa de decréscimo é computada da seguinte forma:

$$TA_{DecrNaorep} = 1 - \frac{TA_{NaorepCom}}{TA_{sem}} \quad (5.4)$$

onde  $TA_{NaorepCom}$  é a taxa de acerto coletada no sistema com *churn* mas sem dados replicados e  $TA_{sem}$  é a taxa de acerto no sistema sem *churn*.

Na Figura 5.7 mostra-se a taxa de decréscimo para todas as políticas avaliadas no contexto anterior. Apesar do aumento da capacidade de armazenamento do sistema, a taxa de decréscimo observada é de aproximadamente 40% quando o *churn* é definido pelo modelo exponencial. Quando o *churn* é definido pelo modelo cauda longa, a taxa de decréscimo é de 20% aproximadamente (Figura 5.7(b)).



(a) Modelo de *churn* exponencial

(b) Modelo de *churn* cauda longa

Figura 5.7: Taxa de decréscimo de acertos das políticas de gerenciamento de objetos em um sistema sem replicação de dados.

Comparando os resultados apresentados na Figura 5.4, o desempenho das políticas sob os dois modelos de *churn* é reduzido em 50%. A partir desses resultados pode-se concluir que: *i*) o aumento da capacidade de armazenamento tem impacto limitado no desempenho das políticas; e *ii*) uma quantidade controlada de réplicas tende a melhorar o desempenho do sistema.

As conclusões anteriores sugerem que a utilização de algoritmo para apoio a replicação

de dados em tais sistemas tende a melhorar o seu desempenho. Dessa forma, quando um par iniciar procedimento de saída do sistema deve utilizar esse algoritmo para propagar informações que serão usadas para tomada de decisão de outros pares durante o processo de substituição de objetos.

## 5.5 Considerações Finais do Capítulo 5

Neste capítulo foi avaliado o impacto do *churn* nas políticas de gerenciamento de objetos implementadas em um sistema de arquitetura híbrida (CDN-P2P). Avaliou-se o aumento da capacidade de armazenamento dos pares como uma ação para manter o desempenho das políticas, uma vez que a capacidade de armazenamento do sistema é afetada pelo *churn*. Embora a diferença de desempenho de uma política diminua com o aumento da capacidade de armazenamento dos pares, esse aumento não impacta proporcionalmente a performance da política.

Além disso, foram realizados estudos para analisar a forma como o grau de replicação impacta o desempenho das políticas de gerenciamento de objetos. Verificou-se que o cenário de pior caso, em termos de disponibilidade de conteúdo, é para um sistema sem replicação de dados. Apesar do aumento da capacidade de armazenamento, não se verificaram melhorias no desempenho das políticas no sistema. Por outro lado, a replicação de dados devido ao *churn* melhora o desempenho das políticas até um determinado patamar. Percebe-se, a partir do desempenho da política POP, que, se a quantidade de réplicas estiver sob controle, a disponibilidade de conteúdo pode ser melhorada.

## CAPÍTULO 6

# POLÍTICAS DE GERENCIAMENTO DE OBJETOS EM SISTEMAS CDN-P2P

A sinergia entre redes P2P e CDN tem permitido a oferta de serviços de vídeos de larga escala. Sistemas híbridos CDN-P2P, que capitalizam vantagens de ambas, têm sido viabilizados. Nesses sistemas, pares compartilham seus recursos diminuindo a demanda sobre a infraestrutura da CDN. Por outro lado, os servidores da CDN garantem a disponibilidade de conteúdo quando as contribuições dos pares são limitadas pelo *churn*, ou quando o conteúdo for inédito aos pares da rede P2P.

Contudo, sistemas híbridos não são, por si só, capazes de garantir escalabilidade e QoE sem os mecanismos apropriados de gerenciamento. Por exemplo, a disponibilidade de conteúdo diminui se os *buffers* nos pares forem mal gerenciados. De fato, o gerenciamento efetivo do conteúdo armazenado nos pares da rede P2P é necessário, uma vez que dados replicados podem dominar a ocupação dos *buffers* dos pares devido ao *churn* como mostrado no Capítulo 5. Consequentemente, identificar objetos valiosos e torná-los disponíveis na rede P2P tende a reduzir a carga de trabalho submetida aos servidores da CDN.

Neste capítulo são propostas e avaliadas quatro políticas de gerenciamento de conteúdo multimídia em redes CDN-P2P. Essas políticas usam metadados que estão associados aos vídeos para melhorar a disponibilidade de conteúdo em sistemas híbridos CDN-P2P. Diferentes de políticas já implementadas, por exemplo, a GDSP, que separa parte da capacidade de armazenamento para manter o histórico de acesso aos dados, as políticas propostas usam dados disponibilizados pelos provedores de conteúdo, através de seus

sistemas de recomendação, para rapidamente identificar padrões de acesso e, assim, valorar os objetos. Adicionalmente, as políticas propostas capturam a influência dos sistemas de recomendação na popularidade das coleções de vídeos distribuída pelo sistema CDN-P2P.

A avaliação das políticas propostas foi realizada usando dados reais, coletados do site YouTube, em uma variedade de cenários de rede com tamanho e dinâmica identificada em sistemas reais de distribuição de conteúdo. Resultados numéricos mostram que sob as responsabilidades das políticas propostas a disponibilidade de conteúdo pode ser melhorada em mais de 70%, comparada à gerência realizada com a política LFU, e supera em mais de 50% a GDSP, uma política conhecida por realizar gerenciamento efetivo de conteúdo.

O restante deste capítulo está organizado da seguinte forma. Na Seção 6.1 apresenta-se a modelagem do problema de gerência de vídeos em sistema CDN-P2P. Na Seção 6.2 descrevem-se as políticas de gerenciamento de objetos. Posteriormente, na Seção 6.3 apresentam-se os resultados numéricos coletados nos experimentos de simulação. Finalmente, na Seção 6.4, as considerações deste capítulo são apresentadas.

## 6.1 Caracterizando a Distribuição de Conteúdo em um Sistema CDN-P2P

Nesta seção apresenta-se a modelagem do problema de distribuição de vídeos de curta duração em um sistema CDN-P2P. Na Figura 6.1(a) ilustra-se a dinâmica no tratamento das requisições de conteúdo, como segue: um par realiza requisição para assistir a um vídeo (passo 1). O servidor por sua vez analisa se os pares da rede P2P possuem o vídeo requisitado, caso positivo, é devolvida uma lista de pares que possuem tal vídeo (passo 2). O par requisitante estabelece conexão com os pares dessa lista e começa a receber o fluxo de vídeo (passo 3). Se o vídeo requisitado não puder ser servido pelos pares, ou por ser inédito ou por alguma falha no sistema, a requisição ao vídeo é atendida pela CDN (passo 4).

A Figura 6.1(b) ilustra a imagem do sistema em um dado tempo  $t$ . Seja  $S$  uma coleção de  $M$  vídeos armazenada nos servidores da CDN. Cada vídeo  $i$  ( $i = 1, \dots, M$ ) da coleção  $S$  possui um tamanho  $s_i$  e um custo  $k_i$  que caracteriza a importância do objeto.

A rede P2P que apoia a distribuição da coleção  $S$  pela CDN possui  $N$  pares, cada um com capacidade  $c_j$ , e tem a sua participação afetada pela dinâmica dos pares. Devido a essa dinâmica, um vídeo  $i$  pode acabar replicado em vários pares. Assim, a variável

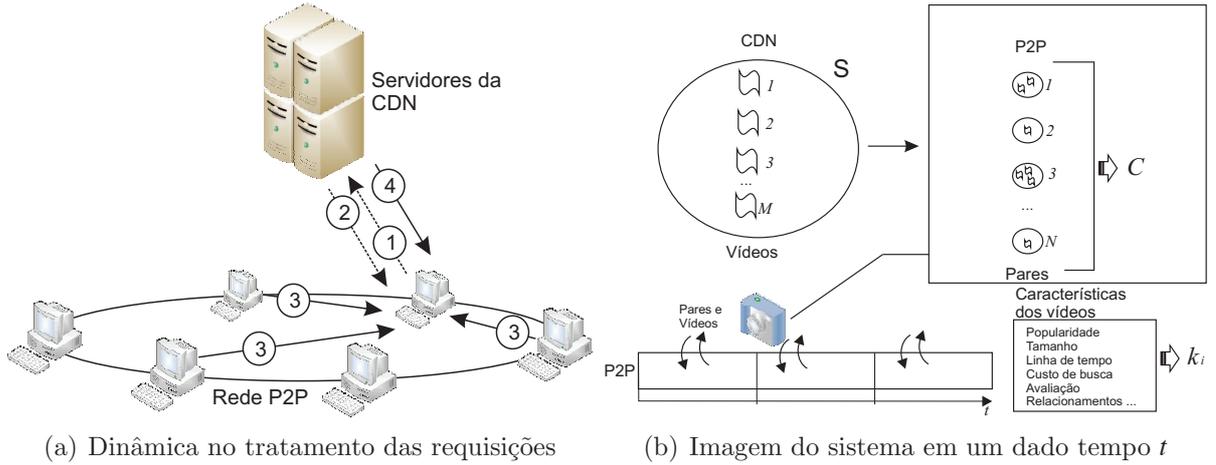


Figura 6.1: Modelagem do problema de gerência de vídeo em sistema CDN-P2P.

binária  $x_{ij}$ , ( $i = 1, \dots, M; j = 1, \dots, N$ ), será igual a 1 (um) se o vídeo  $i$  estiver armazenado no par  $j$ , e será igual a zero caso contrário.

Neste cenário, deseja-se minimizar a carga de trabalho imposta à infraestrutura de servidores, através das requisições realizadas aos vídeos armazenados nessa. Em outras palavras, deseja-se maximizar o conteúdo importante  $z$ , definido pelo custo dos objetos  $k_i$ , na porção P2P do sistema, e dessa forma reduzir, a um número mínimo, as requisições atendidas diretamente pelos servidores. A seguinte formulação expressa esse problema:

$$\begin{aligned}
 \text{Maximizar } z &= \sum_{j=1}^N \sum_{i=1}^M k_i x_{ij} \\
 \text{Sujeito a: } & \sum_{i=1}^M s_i x_{ij} \leq c_j, \quad j = 1, \dots, N \\
 & \sum_{j=1}^N x_{ij} \leq N, \quad i = 1, \dots, M \\
 & \sum_{i=1}^M x_{ij} \leq M, \quad j = 1, \dots, N \\
 & x_{ij} \in \{0, 1\}, \quad j = 1, \dots, N; \quad i = 1, \dots, M
 \end{aligned}$$

A formulação apresentada caracteriza o problema da mochila 0-1 múltipla, descrito em Martello e Toth (1981), e não possui solução ótima em tempo polinomial. A representação da dinâmica dos pares em sua definição, isto é, a conexão e desconexão dos pares, a qualquer instante  $t$  agregam características estocásticas ao problema, ver Figura 6.1(b), o que o torna pelo menos tão difícil quanto o problema descrito.

Dado o problema de gerência de vídeo em um sistema CDN-P2P ser NP-difícil, isto é, não possui solução ótima em tempo polinomial, as políticas de gerenciamento de tais vídeos são heurísticas, que buscam soluções aproximadas para o problema.

## 6.2 Políticas para Gerenciamento de Objetos Baseadas em Custo e em Conteúdo

Nesta seção apresentam-se os algoritmos para gerenciamento de objetos armazenados em um sistema de distribuição de VoD, que usa uma rede de sobreposição P2P para apoiar a tarefa de distribuição, ver Seção 5.1 para mais detalhes. Os algoritmos propostos decidem armazenar ou excluir objetos dos pares usando o valor do objeto, que é calculado por uma função de custo geral  $k_i$ , também chamado de valor principal. Essa função usa informações associadas aos objetos para valorá-los e um mecanismo de envelhecimento  $L$  similar ao mecanismo utilizado por Jin e Bestavros (2000). O Algoritmo 2 descreve a dinâmica de gerenciamento dos objetos no sistema de distribuição.

---

**Algoritmo 2:** Algoritmo para gerenciamento de objetos baseado em custo e conteúdo.

---

```

Entrada: ( $i$ )
Saída:  $k_i$ 
 $L \leftarrow 0.0$ ;
para cada requisição a  $i$  faça
  se  $i$  está na cache então
    // Atualização de objetos
     $k_i \leftarrow L + (v_i + F_i) \times \frac{c_i}{s_i}$ ;
  senão
    se  $s_i \leq \text{cache}$  então
      enquanto não existir espaço suficiente na cache para  $i$  faça
         $L \leftarrow \min[k_q | q \in \text{cache}]$ ;
        remover  $q$  que satisfaça  $k_q = L$ ;
      armazena  $i$ ;
      // Substituição de objetos
       $k_i \leftarrow L + F_i \times \frac{c_i}{s_i}$ ;

```

---

No Algoritmo 2,  $v_i$ ,  $c_i$  e  $s_i$  definem, respectivamente, o total de visualização local, custo associado com o *download* e o tamanho do objeto  $i$ . A função  $c_i$  é dependente dos recursos de rede disponíveis para cada usuário. Neste estudo definiu-se o valor de  $c_i$  igual a 1 para todos os objetos. Essa definição baseou-se no estudo de Cao e Irani (1997), onde se identificou que com esse valor se obtém a melhor taxa de acerto para política baseada

no tamanho do objeto (GDS). A função  $F_i$  é o fator de custo e valora os objetos de acordo com os metadados disponibilizados pelo distribuidor do conteúdo.

O valor de custo geral de um objeto  $i$  é definido por,

$$k_i = L + F_i \times \frac{c_i}{s_i} \quad (6.1)$$

se o objeto não estiver na cache, e por,

$$k_i = L + (v_i + F_i) \times \frac{c_i}{s_i} \quad (6.2)$$

caso esse esteja na cache. Em ambos os casos,  $F_i$  é definido utilizando-se os seguintes dados: o número de visualizações, a data de publicação do conteúdo e a taxa de qualificação. Os dois primeiros dados são coletados pelas ferramentas de gerência do tráfego e o último é gerado pelos usuários que espontaneamente avaliam o conteúdo. O mecanismo de recomendação do YouTube, baseado em tais informações, influencia as escolhas dos usuários (DAVIDSON et al., 2010). A seguir, mostra-se como o fator de custo  $F_i$  é definido para cada dado utilizado.

### 6.2.1 Fator de Custo Baseado em Frequência de Visualizações

Para definir o fator de custo baseado em frequência de visualizações globais, o cálculo de  $k_i$  considera o número de visualizações registrado pelo sistema de distribuição de conteúdo em seu nível global. Assim, o fator  $F_i$  é definido por,

$$F_i = \frac{vg_i}{\max[vg_q \mid q \in \text{cache}]} \quad (6.3)$$

onde  $\max[vg_q \mid q \in \text{cache}]$  determina o maior valor do contador de visualizações globais registrado dos objetos já armazenados na cache e  $vg_i$  é o número de visualização global do objeto  $i$  sendo avaliado.

No sistema de distribuição de VoD estudado (Seção 5.1), os objetos acessados são mantidos na comunidade e requisições futuras para eles são contadas somente pelos pares que armazenam o objeto, isto é, pares da rede P2P. A política de gerenciamento de objetos baseada nesse fator de custo foi chamada de GDSVF, do inglês *Greedy-Dual Size View Frequency-based*.

### 6.2.2 Fator de Custo Baseado em Qualificação

Aos usuários do YouTube é permitido qualificar os vídeos que eles assistem, “gostou” ou “não gostou”. Essa impressão pessoal é usada para posicionar vídeos relacionados (DAVIDSON et al., 2010). Além disso, usuários podem acessar os vídeos mais bem qualificados em diferentes escalas de tempo, isto é, dia, semana, mês e todos os tempos; durante uma sessão de acesso ao YouTube. Devido à relevância da qualificação, definiu-se o fator de custo baseado em qualificação global, isto é,  $F_i$  é dado por,

$$F_i = \frac{qlf_i}{\max[qlf_q \mid q \in \text{cache}]} \quad (6.4)$$

onde  $\max[qlf_q \mid q \in \text{cache}]$  fornece a qualificação máxima registrada dos objetos já armazenados na cache e  $qlf_i$  é a qualificação do objeto  $i$ .

A qualificação feita pelo usuário é considerada constante durante a simulação. Em outras palavras, uma vez que o vídeo é qualificado globalmente ele manterá esse valor e será apropriadamente classificado pela política. Essa política recebeu o nome de GDSR, do inglês *Greedy-Dual Size Rating-based*.

### 6.2.3 Fator de Custo Baseado em Taxa de Visualizações

Esse fator de custo é baseado na idade do vídeo, isto é, o intervalo de tempo decorrido desde a sua publicação, e nas visualizações registradas nesse intervalo. Em Davidson et al. (2010) essas informações são classificadas como sinais de qualidade do vídeo e têm sido usadas pelo sistema de recomendação do YouTube para medir a probabilidade de tal vídeo ser apreciado pelos usuários. Vídeos que possuem alta probabilidade de serem apreciados são candidatos a lista de relacionados de um *feed* de vídeos. Baseado na contagem de visualizações e relevância da idade para classificação define-se a função de custo geral  $k_i$  baseada nas visualizações globais por unidade de tempo  $tv g_i = \frac{vg_i}{idade_i}$ . Dessa forma, o fator  $F_i$  é dado por,

$$F_i = \frac{tv g_i}{\max[tv g_q \mid q \in \text{cache}]} \quad (6.5)$$

onde  $tv g_i$  é a taxa de visualização global do objeto  $i$  e  $\max[tv g_q \mid q \in \text{cache}]$  é a maior taxa entre todos os objetos já armazenados na cache.

Novamente, devido à arquitetura do sistema de distribuição dos vídeos, esses sinais de qualidade do vídeo são modificados pela atividade da comunidade. Especificamente, um vídeo tem o seu contador de visualizações adicionado e a sua idade atualizada durante a sua permanência na rede P2P do sistema de distribuição. Essa política recebeu o nome

de GDSVR, do inglês *Greedy-Dual Size View Rate-based*.

### 6.2.4 O Fator de Custo Adaptativo Baseado em Frequência de Visualização

A efetividade de políticas de gerenciamento de objetos baseadas em sinais de qualidade do vídeo depende da convergência entre os padrões de acesso local, que representa o interesse de uma comunidade por um conteúdo, e o global, que representa o interesse geral pelo conteúdo. Em caso de descompasso entre esses padrões, o uso dos sinais de qualidade colhidos a partir do interesse geral podem penalizar objetos, fazendo com que a política de gerenciamento empregada tenha desempenho ruim.

Para considerar o possível descompasso de interesses introduziram-se modificações na dinâmica da política de gerenciamento de conteúdo, ver Algoritmo 4. Essas modificações identificam se os acessos locais e globais convergem ou divergem e, com base nisso, a política decide qual função geral de custo  $k_i$  será utilizada, com ou sem o fator  $F_i$  (Algoritmo 3).

---

**Algoritmo 3:** Algoritmo para Calcular  $k_i$

---

**Entrada:**  $(i, L, t_s^{max})$

**Saída:**  $(k_i)$

se  $i$  for inédito então

    se  $t_s \geq t_s^{max}$  então

$k_i \leftarrow L + \frac{c_i}{s_i};$

    senão

$k_i \leftarrow L + F_i \times \frac{c_i}{s_i};$

senão

    se  $t_s \geq t_s^{max}$  então

$k_i \leftarrow L + v_i \times \frac{c_i}{s_i};$

    senão

$k_i \leftarrow L + (F_i + v_i) \times \frac{c_p}{s_p};$

---

Nesta política (Algoritmo 4) é utilizado o fator de custo  $F_i$  baseado nas visualizações globais (Equação 6.5), que apresenta a melhor valoração dos objetos em cenário convergente. Ao desconsiderar o fator de custo  $F_i$ , ou seja, identificar descompasso entre os padrões de acesso, essa política define o valor de  $k_i$  utilizando a seguinte equação:

$$k_i = L + v_i \times \frac{c_i}{s_i} \quad (6.6)$$

Essa implementação da política percebe o contexto em que está atuando a partir da comparação entre taxa de sensibilidade  $t_s$  e o limiar máximo de sensibilidade  $t_s^{max}$ . Os vídeos armazenados nos pares são mantidos em uma lista  $P$  de tamanho  $n$ , ordenada de forma decrescente em função do valor de  $k_i$ . Seja  $vg_i^1$  o número de visualizações globais do objeto  $i$  localizado na primeira posição de  $P$ . Seja  $vg_i^n$  o número de visualizações globais do objeto  $i$  localizado na última posição de  $P$ . Nesse cenário, a taxa de sensibilidade calculada para um par  $j$  qualquer é dada por:

$$t_s = \frac{\# \text{ de vezes que } vg_i^1 < vg_i^n}{\# \text{ total de acessos}} \quad (6.7)$$

Na definição de  $t_s$ , supõe-se que uma maior concentração de vídeo com um número reduzido de visualizações globais  $vg_i$ , no início da lista, indica que padrões de acesso local e global divergem. O limiar máximo de sensibilidade  $t_s^{max}$  define quanto descompasso entre os padrões de acesso local e global será aceito pela política. Dessa forma, quando esse descompasso atinge o valor estabelecido em  $t_s^{max}$  a valoração passa a desconsiderar os sinais globais do objeto. O limiar máximo de sensibilidade é um parâmetro dependente da carga de trabalho submetida ao sistema. Nos estudos numéricos apresentados nesta dissertação, utilizou-se um limiar máximo de valor  $t_s^{max} = 0,5$ .

Além de perceber o contexto através da taxa de sensibilidade, nesta nova política implementou-se um mecanismo de bloqueio que impede a substituição de objetos de maior valor por objetos de menor valor, motivada apenas pelo ineditismo. Para o bloqueio é realizado um pré-cálculo de  $k_i$  considerando o valor de  $L = 0$ . Caso o valor pré-calculado de  $k_i$  seja menor que o valor atual de  $L$  a inserção do objeto  $i$  na cache é bloqueada, pois esse objeto tem menos valor que todos os demais que estão na cache.

---

**Algoritmo 4:** Algoritmo para gerenciamento de objetos adaptativo ao contexto.

---

```

 $L \leftarrow 0.0;$ 
 $t_s^{max} \leftarrow 0.5;$ 
para cada requisição ao objeto  $i$  faça
  se  $i$  está na cache então
     $k_i \leftarrow \text{calcular}(i, L, t_s^{max});$ 
  senão
    // Pré-cálculo de  $k_i$ 
     $k_i \leftarrow \text{calcular}(i, 0, t_s^{max});$ 
    // Bloqueia vídeos com  $k_i$  menor ou igual a  $L$ 
    se  $(k_i \geq L)$  e  $(s_i \leq \text{cache})$  então
      enquanto não houver espaço livre na cache para  $i$  faça
         $L \leftarrow \min[k_q \mid q \in \text{cache}];$ 
        remover  $q$  que satisfaça  $k_q = L;$ 
      armazena  $i;$ 
       $k_i \leftarrow \text{calcular}(i, L, t_s^{max});$ 

```

---

O Algoritmo 4 mostra as modificações implementadas no Algoritmo 2, quando o fator de custo adaptativo é usado para valorar os objetos. A dinâmica dessa política ao manter um fator de sensibilidade para perceber a tendência entre os padrões de acesso faz com que os objetos sejam valorados de acordo com o contexto. Devido a essa característica, tal política recebeu o nome de GDSA, do inglês *Greedy-Dual Size Adaptive*.

## 6.3 Resultados Numéricos

Para avaliar a efetividade das políticas propostas, experimentos numéricos foram conduzidos usando a política LFU como o limiar mínimo de desempenho, em termo de taxa de acerto, esperado para o sistema de distribuição apresentado na Seção 5.1. As políticas propostas são comparadas com a GDSP, que é uma política baseada em custo cujo desempenho na gerência de objetos obteve destaque na literatura (JIN; BESTAVROS, 2000). Ambas, LFU e GDSP, estão descritas no Capítulo 3.

O desempenho da política LRU, em termos de taxa de acerto, também foi avaliado como limiar mínimo de desempenho, mas como seu desempenho foi pior que o registrado pela LFU, para todas as cargas de trabalho no cenário de rede considerado, os resultados referentes a essa política não são apresentados nesta dissertação.

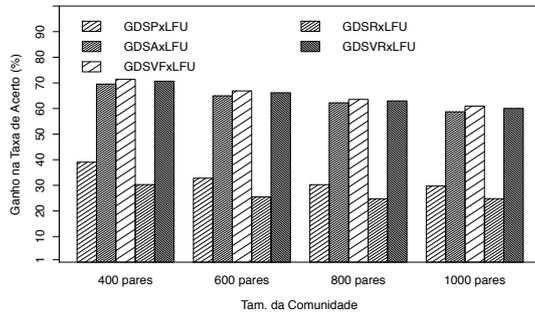
Para a realização dos experimentos numéricos, o sistema de distribuição apresentado na Seção 5.1 foi ajustado para servir vídeos das seguintes categorias registrados no site do YouTube: entretenimento, esporte e música. Esses vídeos foram coletados de acordo com o procedimento descrito na Seção 4.2. Uma vez que os vídeos sementes usados pelo coletor foram obtidos no nível país da CDN, usada pelo YouTube, considerou-se que o padrão de acesso nos vídeos acessados expressa a preferência regional dentro da infraestrutura de distribuição. As redes sobrepostas da parte P2P do sistema são formadas em torno das categorias de vídeos e seus tamanhos variam de 400 a 1000 pares, fazendo sua capacidade de armazenamento variar de 3% a 8% de toda a coleção de vídeos. Os acessos aos vídeos no sistema foram gerados utilizando o procedimento descrito na Seção 5.2.

O desempenho das políticas propostas foi avaliado utilizando os dois modelos de *churn* estudados no Capítulo 5, exponencial e cauda longa, aplicados em dois cenários de padrão de acesso, um no qual os acessos locais e globais convergem e outro no qual divergem. Utilizaram-se esses dois cenários para avaliar, em cenário que os acessos não convergem, o desempenho das políticas que se valem dos acessos globais, para valorar os objetos, tal como, as baseadas em frequência de visualização (GDSVF) e em taxa de visualização (GDSVR).

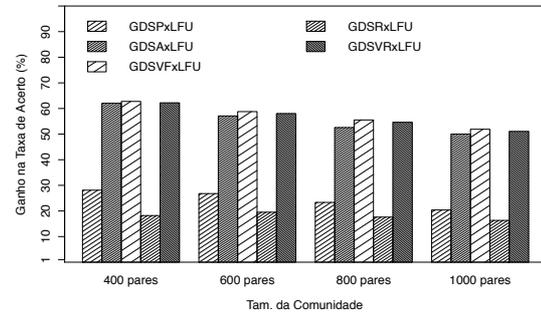
Essas políticas são mais efetivas quando os acessos de uma comunidade tendem ao acesso global, caso contrário, tais acessos têm influência negativa na determinação da importância de um objeto para a comunidade. Para avaliar o impacto da divergência entre os padrões de acesso local e global, utilizou-se o pior caso, no qual os vídeos das coleções foram classificados a partir da contagem das visualizações em ordem reversa ao padrão de acesso global, isto é, o vídeo com a menor probabilidade de acesso global é o de maior probabilidade de acesso local.

No cenário em que os padrões de acesso local e global convergem (Figura 6.2), observa-se que as políticas baseadas em frequência de visualização (GDSVF), em taxa de visualização (GDSVR) e adaptativa (GDSA) apresentam os melhores desempenhos para a comunidade esporte, com ganhos respectivos de 73,5%, 72,8% e 70,0% para o modelo de *churn* exponencial e 67,8%, 67,3% e 66,4% para o modelo de *churn* cauda longa. Essas políticas mantiveram bom desempenho para as três categorias de vídeos analisadas. As políticas baseadas em popularidade (GDSP) e baseada em qualificação (GDSR) obtiveram melhor desempenho também para a comunidade esporte, com ganhos respectivos de 50,3% e 46,9% para o modelo de *churn* exponencial e 44,5% e 41,1% para o modelo cauda longa. No entanto, não apresentaram ganho para a comunidade música. Para esta comunidade as políticas GDSVR, GDSVF e GDSA superaram a política GDSP respectivamente em

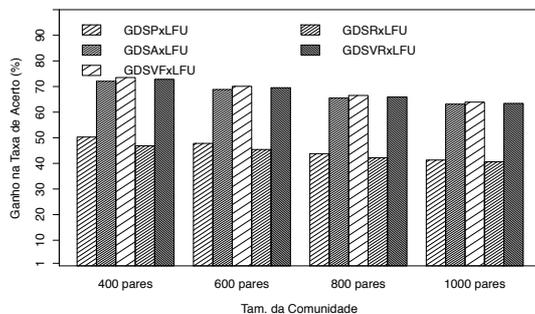
70,1%, 69,3% e 54,9% para o modelo de *churn* exponencial e 66,2%, 65,8% e 52,0% para o modelo de *churn* cauda longa.



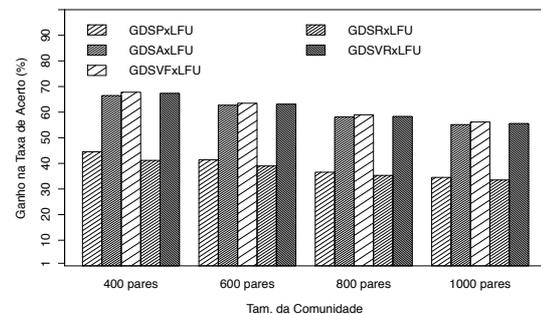
(a) Comunidade entretenimento com modelo de *churn* exponencial



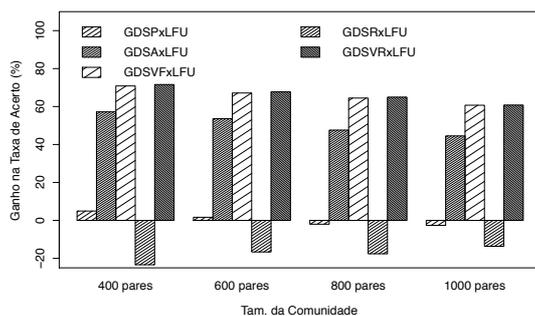
(b) Comunidade entretenimento com modelo de *churn* cauda longa



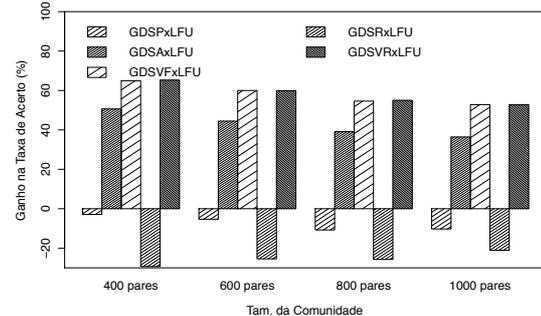
(c) Comunidade esporte com modelo de *churn* exponencial



(d) Comunidade esporte com modelo de *churn* cauda longa



(e) Comunidade música com modelo de *churn* exponencial



(f) Comunidade música com modelo de *churn* cauda longa

Figura 6.2: Ganho sobre a política LFU para o cenário convergente entre padrões de acesso local e global.

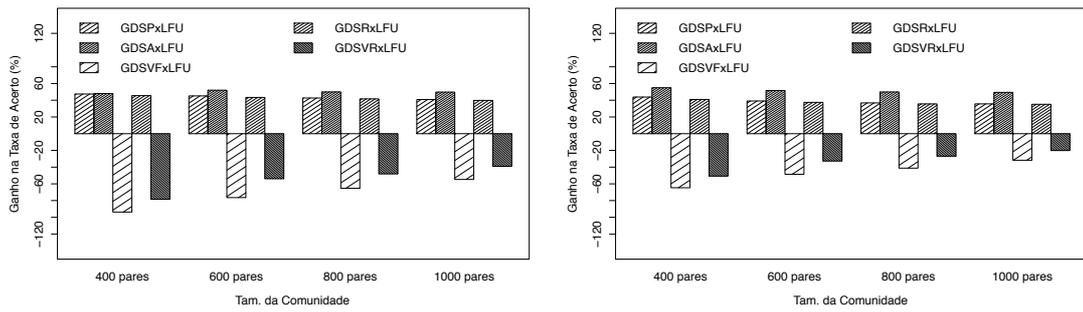
Uma vez que todas as quatro políticas avaliadas utilizam o tamanho do objeto para compor o custo geral dos objetos, avaliou-se cada coleção para determinar qual foi o papel da duração do vídeo na valorização dos objetos distribuídos pelas comunidades entretenimento, esporte e música. No Capítulo 4 apresenta-se a análise da duração dos vídeos,

onde identifica-se que a coleção da categoria música tem a duração dos vídeos concentrada em torno de 221,5 segundos e com menor variabilidade de tamanho, coeficiente de variação 0,38. Em outras palavras, o tamanho do objeto contribui substancialmente para a valoração dos objetos das comunidades entretenimento e esporte, mas tem contribuição limitada na valoração de objetos da comunidade música. A Figura 6.2.(e) e a Figura 6.2.(f) sugerem essa contribuição com o desempenho mínimo, isto é, o desempenho da política LFU sendo igual ou maior que o desempenho das políticas GDSP e GDSR.

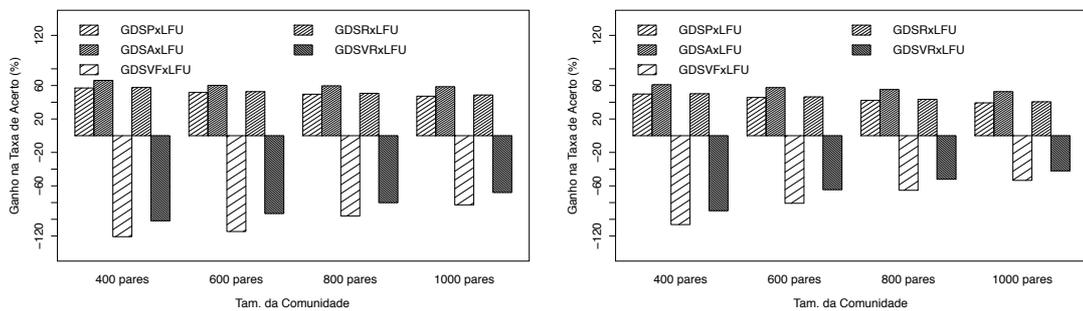
Como identificado no Capítulo 5, o aumento da capacidade de armazenamento do sistema, representado pela quantidade de pares, diminui o ganho no desempenho de uma política. Por isso, o ganho é maior quando o sistema possui 400 pares e diminui a medida que se aumenta a quantidade de pares no sistema. Ao comparar o desempenho das políticas para os dois modelos de *churn* percebe-se que o ganho é maior para o modelo exponencial, isso ocorre devido ao modelo exponencial manter em média menor quantidade de pares ativos que o modelo cauda longa.

A política baseada em qualificação (GDSR) não apresentou nenhuma melhoria em relação à GDSP para o cenário de convergência entre os padrões de acesso local e global (Figura 6.2). Isso ocorre pela dificuldade do conteúdo em agradar todos os visitantes; vídeos com um grande número de visualizações possuem baixa probabilidade de apresentarem qualificação máxima. Um pequeno número de avaliações negativas diminuirá a taxa de qualificação para menos que cinco, o valor máximo de qualificação. Para as três coleções, a maioria dos vídeos com máxima qualificação possuem um pequeno número de visualizações e o número de vídeos com muitas visualizações e qualificação máxima é bem pequeno. Como a qualificação compõe o custo geral dos objetos na política GDSR, objetos com poucos acessos, porém com qualificação máxima, possuem um custo maior, sendo priorizados durante a gerência da cache e, assim, impactando negativamente o desempenho dessa política.

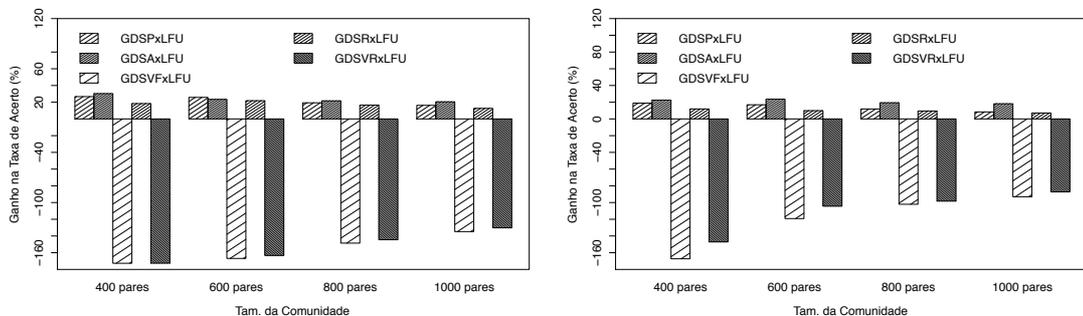
Na Figura 6.3 é apresentado o desempenho das políticas de gerenciamento de objetos para o cenário no qual há divergência entre os padrões de acesso local e global. Observa-se que as políticas GDSA, GDSR e GDSP obtiveram os melhores resultados para a comunidade esporte, com ganhos respectivos de 66,1%, 57,7% e 57,1% para o modelo de *churn* exponencial e de 61,0%, 50,2% e 49,8% para o modelo de *churn* cauda longa. Essas políticas também mantiveram os melhores ganhos para as comunidades entretenimento e música. As políticas baseadas em frequência de visualização (GDSVF) e taxa de visualização (GDSVR) não apresentaram ganhos em relação à LFU.



(a) Comunidade entretenimento com modelo de *churn* exponencial (b) Comunidade entretenimento com modelo de *churn* cauda longa



(c) Comunidade esporte com modelo de *churn* exponencial (d) Comunidade esporte com modelo de *churn* cauda longa



(e) Comunidade música com modelo de *churn* exponencial (f) Comunidade música com modelo de *churn* cauda longa

Figura 6.3: Ganho sobre a política LFU para o cenário divergente entre padrões de acesso local e global.

A perda evidente na taxa de acerto das políticas GDSVF e GDSVR em relação a LFU ocorre pelo fato do mecanismo de valoração dos vídeos para essas políticas ser baseado na contagem de visualizações globais. Vídeos com alta probabilidade de acesso local são menos valorados, visto que esses têm poucas visualizações globais. Apesar dessas políticas agregarem as visualizações locais a partir do segundo acesso ao vídeo, que funcionaria como um capturador do padrão de acesso local, a renovação dos objetos com pouca prioridade ocorrem com muita frequência, dificultando assim, a sensibilidade ao padrão de acesso

local.

As políticas GDSA, GDSR e GDSP têm bom desempenho em relação a LFU, tendo a GDSA apresentado o melhor desempenho dentre elas. O mecanismo de valoração da política GDSP captura a tendência de popularidade no nível local, por isso não sofre impacto das visualizações globais. A política GDSR aproveita-se do engajamento dos membros da comunidade que avaliam os vídeos de mais importância para eles. De fato, no cenário em que a coleção é acessada em ordem inversa, os vídeos com maior probabilidade de acesso local possuem pouco acesso global, porém, são bem qualificados, a maioria com qualificação máxima. Sugerindo que os membros de pequenas comunidades, isto é, aqueles interessados em conteúdos específicos, investem algum tempo qualificando o conteúdo acessado.

Para a política GDSA o desempenho se destaca devido essa possuir um mecanismo dinâmico de sensibilidade entre os padrões de acessos local e global. Esse mecanismo ao identificar a divergência entre os padrões permite a política deixar de utilizar a função baseada na taxa de visualizações globais para utilizar apenas as visualizações locais para valorar os vídeos. Essa política ainda conta com a contribuição do mecanismo de bloqueio de vídeos definidos como de baixo valor para a comunidade.

## 6.4 Considerações Finais do Capítulo 6

Estudos têm demonstrado que a influência do *churn* nos mecanismos de gerenciamento de conteúdo, na porção P2P de um sistema de distribuição de conteúdo híbrido (CDN-P2P), reduz a performance geral desse sistema. Neste capítulo estudou-se a eficácia das políticas baseadas em sinais associados aos vídeos disponíveis no sistema de distribuição de VoD, como forma de melhorar a disponibilidade de conteúdo em um sistema CDN-P2P. Quatro políticas de gerenciamento de objetos (GDSVF, GDSVR, GDSR e GDSA) foram propostas e avaliadas usando dados reais oriundos das coleções de vídeos do YouTube.

Das políticas avaliadas, a GDSVF, a GDSVR e a GDSA obtiveram os melhores resultados quando há convergência entre os padrões de acesso global e local, melhorando a disponibilidade de conteúdo em mais de 70% quando comparadas a política LFU e superando a política GDSP em mais de 50%. Para o cenário de divergência entre os acessos locais e globais, as políticas GDSA e GDSR obtiveram ganhos em relação a LFU, tendo a GDSA alcançado o melhor ganho 66,1%. A política GDSA apresentou ótimo desempenho, superando a política GDSP em todos os cenários.

## CAPÍTULO 7

## CONSIDERAÇÕES FINAIS

Aplicações para distribuição de fluxo de vídeo sob demanda baseadas na arquitetura cliente-servidor têm problemas de escalabilidade. Soluções que utilizam CDN diminuem a latência e melhoram a disponibilidade dos conteúdos, mas têm alto custo financeiro para instalação e manutenção. Soluções baseadas em redes P2P têm sido discutidas, uma vez que são naturalmente escaláveis e robusta. No entanto, a efetividade dos serviços depende do engajamento dos pares; caso o sistema apresente poucos pares ativos, a disponibilidade de conteúdo é baixa. Neste contexto, surgiram os sistemas de distribuição híbridos CDN-P2P, que capitalizam as vantagens de ambas as arquiteturas. Porém, por si só, sistemas híbridos não resolvem o problema se os conteúdos distribuídos não forem bem gerenciados, de tal forma que essa gerência aumente a disponibilidade de conteúdos e, por conseguinte, minimize a carga de trabalho nos servidores.

Para o contexto de VoD abordado neste trabalho, no qual a parte P2P do sistema de distribuição possui *churn*, ou seja, os pares entram e saem do sistema aleatoriamente, políticas de gerência de objetos tradicionais não se mostram efetivas. Pois, essas foram propostas para contextos diferentes dos abordados nesta dissertação. Através de análise de dados reais coletados do site mais popular de compartilhamento de vídeo, o YouTube, identificaram-se estatísticas importantes, quanto ao padrão de acesso e duração dos vídeos. Utilizou-se de tais conhecimentos para oportunizar melhorias nos sistemas de distribuição de vídeos.

Cientes da presença do *churn* em um sistema P2P, estudou-se o impacto desse nas políticas de gerenciamento de objetos implementadas em um sistema de arquitetura híbrida (CDN-P2P). Avaliou-se o aumento da capacidade de armazenamento dos pares como uma

ação para manter o desempenho das políticas, uma vez que a capacidade de armazenamento do sistema é afetada pelo *churn*. Embora o ganho no desempenho de uma política diminua com o aumento da capacidade de armazenamento dos pares, esse aumento não impacta proporcionalmente a performance da política. Avaliou-se também o quanto a replicação de dados afeta o desempenho das políticas no sistema. Percebeu-se que o pior desempenho ocorre quando não há replicação de dados. Por outro lado, constatou-se que a replicação melhora o desempenho das políticas até certo nível, necessitando de uma gerência mais efetiva para obter maior disponibilidade do conteúdo.

Finalmente, estudou-se a efetividade das políticas baseadas em sinais associados aos vídeos disponíveis em sistema de distribuição de VoD, como forma de melhorar a disponibilidade de conteúdo no sistema avaliado. Quatro políticas de gerenciamento de objetos (GDSVF, GDSVR, GDSR e GDSA) foram propostas e avaliadas em simulação usando dados reais oriundos das coleções de vídeos do YouTube. Quando há convergência entre os padrões de acesso, local e global, as políticas GDSVF, GDSVR e GDSA obtêm os melhores desempenhos, com ganhos sobre a política LFU de 73,5%, 72,8% e 70,0%, respectivamente. Para o cenário que os padrões divergem, a política GDSA obteve o melhor desempenho, com ganho de 66,1% sobre a LFU. Nos estudos realizados, percebeu-se que políticas baseadas em sinais, estes associados aos vídeos disponível nos provedores de conteúdos, obtêm os melhores desempenhos para o cenário de convergência entre os padrões de acesso. Considerando qualquer cenário, a política GDSA alcançou os melhores resultados, superando a GDSP para ambos os cenários.

## 7.1 **Trabalhos Futuros**

Como trabalhos futuros, relacionados aos padrões de acesso, pode-se utilizar outras distribuições estatísticas para modelar o comportamento das visualizações, uma vez que as distribuições analisadas não modelaram com boa precisão os acessos. Em relação ao estudo do impacto do *churn* no desempenho das políticas, pretende-se avaliar mecanismos para controlar a quantidade de dados replicados no sistema e investigar se mecanismos de incentivo a participação dos pares, desenvolvidos para outros sistemas de distribuição de conteúdo, podem ser aplicados em um sistema híbrido (CDN-P2P).

Em continuidade ao estudo da efetividade dos sinais associados aos vídeos no desempenho de políticas de gerenciamento de objetos, pretende-se estender a avaliação para a influência das relações criadas com outros vídeos, utilizando-se, para isso, métricas de redes sociais para identificar vídeos mais valorosos em uma comunidade.

---

## REFERÊNCIAS

ABRAMS, M. et al. *Caching Proxies: Limitations and Potentials*. Blacksburg, VA, USA, 1995.

AKAMAI, T. *Akamai faster forward*. 2012. Disponível em: <<http://www.akamai.com/>>. Acesso em: 09 fev. 2012.

ANNAPUREDDY, S. et al. Is high-quality vod feasible using p2p swarming? In: *Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA: ACM, 2007. (WWW '07), p. 903–912. ISBN 978-1-59593-654-7.

ARLITT, M. et al. Evaluating content management techniques for web proxy caches. *SIGMETRICS Perform. Eval. Rev.*, ACM, New York, NY, USA, v. 27, p. 3–11, mar. 2000. ISSN 0163-5999.

ARLITT, M.; FRIEDRICH, R.; JIN, T. Performance evaluation of web proxy cache replacement policies. *Perform. Eval.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 39, p. 149–164, fev. 2000. ISSN 0166-5316.

BAUER, D. F. Constructing confidence sets using rank statistics. *Journal of the American Statistical Association*, v. 67, n. 339, p. 687–690, set. 1972.

BEZERRA, A. et al. A content-oriented web cache poli. In: *Communications, 2009. LATINCOM '09. IEEE Latin-American Conference on*. [S.l.: s.n.], 2009. p. 1–6.

CALZAROSSA, M.; GELENBE, E. (Ed.). *Performance Tools and Applications to Networked Systems, Revised Tutorial Lectures [from MASCOTS 2003]*, v. 2965 de *Lecture Notes in Computer Science*, (Lecture Notes in Computer Science, v. 2965). [S.l.]: Springer, 2004. ISBN 3-540-21945-5.

CAO, P.; IRANI, S. Cost-aware www proxy caching algorithms. In: *Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems*. Berkeley, CA, USA: USENIX Association, 1997. p. 18–18.

- CASTRO, M. et al. Splitstream: high-bandwidth multicast in cooperative environments. In: *Proceedings of the nineteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2003. (SOSP '03), p. 298–313. ISBN 1-58113-757-5.
- CHA, M. et al. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2007. (IMC '07), p. 1–14. ISBN 978-1-59593-908-1.
- CHA, M. et al. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Trans. Netw.*, IEEE Press, Piscataway, NJ, USA, v. 17, p. 1357–1370, out. 2009. ISSN 1063-6692.
- CHENG, X.; LIU, J. Netteube: Exploring social networks for peer-to-peer short video sharing. *INFOCOM*, 2009.
- CHERKASOVA, L. *Improving WWW Proxies Performance with Greedy-Dual-Size-Frequency Caching Policy*. [S.l.], 1998.
- CORNUEJOLS, G.; FISHER, M. L.; NEMHAUSER, G. L. Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms. *Management Science*, v. 23, n. 8, p. 789–810, 1977.
- CRESPO, A.; GARCIA-MOLINA, H. Semantic overlay networks for p2p systems. In: MORO, G.; BERGAMASCHI, S.; ABERER, K. (Ed.). *Agents and Peer-to-Peer Computing*. [S.l.]: Springer Berlin / Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3601). p. 1–13.
- DAILYMOTION. *DailyMotion Homepage*. 2012. Disponível em: <<http://www.dailymotion.com>>. Acesso em: 09 dez. 2011.
- DAVIDSON, J. et al. The youtube video recommendation system. In: *Proceedings of the fourth ACM conference on Recommender systems*. New York, NY, USA: ACM, 2010. (RecSys '10), p. 293–296. ISBN 978-1-60558-906-0.
- EDGECAST, N. *CDN Solutions*. 2012. Disponível em: <<http://www.edgecast.com/>>. Acesso em: 05 mar 2012.
- FUJITA, N. et al. Coarse-grain replica management strategies for dynamic replication of web contents. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 45, p. 19–34, May 2004. ISSN 1389-1286.
- GOOGLE. *YouTube APIs and Tools*. 2012. Disponível em: <<http://code.google.com/intl/en/apis/youtube/overview.html>>. Acesso em: 09 dez. 2011.
- HAREESH, K.; MANJIAH, D. H. Design issues and challenges of peer-to-peer video on demand system. In: *First International Conference on Artificial Intelligence, Soft Computing and Applications*. [S.l.]: Technology, 2011. (AIAA-2011, 3), p. 57–67. ISBN 978-1-921987-01-4.

- HEFEEDA, M.; SALEH, O. Traffic modeling and proportional partial caching for peer-to-peer systems. *IEEE/ACM Trans. Netw.*, IEEE Press, Piscataway, NJ, USA, v. 16, p. 1447–1460, dez. 2008. ISSN 1063-6692.
- HUANG, C. et al. Understanding hybrid cdn-p2p: why limelight needs its own red swoosh. In: *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. New York, NY, USA: ACM, 2008. (NOSSDAV '08), p. 75–80. ISBN 978-1-60558-157-6.
- JANNOTTI, J. et al. Overcast: Reliable multicasting with an overlay network. In: *Usenix OSDI Symposium 2000*. [S.l.: s.n.], 2000. p. 197–212.
- JIANG, W. et al. A new proxy scheme for large-scale p2p vod system. In: *Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on*. [S.l.: s.n.], 2008. v. 1, p. 512–518.
- JIN, S.; BESTAVROS, A. Popularity-aware greedy dual-size web proxy caching algorithms. In: *Proceedings of the The 20th International Conference on Distributed Computing Systems ( ICDCS 2000)*. Washington, DC, USA: IEEE Computer Society, 2000. (ICDCS '00), p. 254–. ISBN 0-7695-0601-1.
- JOHNSON, K. L. et al. The measured performance of content distribution networks. *Computer Communications*, p. 202–206, 2001.
- KARLSSON, M.; KARAMANOLIS, C. Choosing replica placement heuristics for wide-area systems. In: *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on*. [S.l.: s.n.], 2004. p. 350–359. ISSN 1063-6927.
- KOSTIĆ, D. et al. Bullet: high bandwidth data dissemination using an overlay mesh. In: *Proceedings of the nineteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2003. (SOSP '03), p. 282–297. ISBN 1-58113-757-5.
- KULKARNI, V.; DEVETSIKIOTIS, M. Communication timescales, structure and popularity: Using social network metrics for youtube-like multimedia content distribution. In: *Communications (ICC), 2010 IEEE International Conference on*. [S.l.: s.n.], 2010. p. 1–5. ISSN 1550-3607.
- LEE, S. H.; KIM, P.-J.; JEONG, H. Statistical properties of sampled networks. *Phys. Rev. E*, American Physical Society, v. 73, p. 016102, Jan 2006.
- LI, B. et al. On the optimal placement of web proxies in the internet. In: *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. [S.l.: s.n.], 1999. v. 3, p. 1282–1290 vol.3.
- LIU, Y.; GUO, Y.; LIANG, C. A survey on peer-to-peer video streaming systems. *Peer-to-Peer Networking and Applications*, v. 1, n. 1, p. 18–28, mar 2008. ISSN 1936-6450.
- MARTELLO, S.; TOTH, P. Heuristic algorithms for the multiple knapsack problem. *Computing*, Springer Wien, v. 27, p. 93–112, 1981. ISSN 0010-485X. 10.1007/BF02243544.

- MIR, N.; NATARAJA, M.; RAVIKRISHNAN, S. A performance evaluation study of video-on-demand traffic over ip networks. In: *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*. [S.l.: s.n.], 2011. p. 757–762.
- MIRCHANDANI, P. B.; FRANCIS, R. L. *Discrete Location Theory*. Hardcover: Wiley, 1990. (Wiley Series in Discrete Mathematics and Optimizat). ISBN 978-0-471-89233-5.
- MONITORING, W. *YouTube Facts & Figures (history & statistics)*. maio 2010. SITEIMPULSE. Disponível em: <<http://www.website-monitoring.com/blog/2010/05/>>. Acesso em: 16 fev. 2012.
- PALLIS, G.; VAKALI, A. Insight and perspectives for content delivery networks. *Commun. ACM*, ACM, New York, NY, USA, v. 49, p. 101–106, jan. 2006. ISSN 0001-0782.
- PASSARELLA, A. A survey on content-centric technologies for the current internet: Cdn and p2p solutions. *Computer Communications*, v. 35, n. 1, p. 1–32, 2012. ISSN 0140-3664.
- PATHAN, M.; BUYYA, R.; VAKALI, A. Content delivery networks: State of the art, insights, and imperatives. In: BUYYA, R.; PATHAN, M.; VAKALI, A. (Ed.). *Content Delivery Networks*. [S.l.]: Springer Berlin Heidelberg, 2008, (Lecture Notes in Electrical Engineering, v. 9). p. 3–32. ISBN 978-3-540-77887-5. 10.1007/978-3-540-77887-5\_1.
- PAUL, S.; PAN, J.; JAIN, R. Architectures for the future networks and the next generation internet: A survey. *Computer Communications*, v. 34, n. 1, p. 2–42, 2011. ISSN 0140-3664.
- PELTOTALO, J. et al. Peer-to-peer streaming technology survey. In: *Proceedings of the Seventh International Conference on Networking*. Washington, DC, USA: IEEE Computer Society, 2008. p. 342–350. ISBN 978-0-7695-3106-9.
- PITKOW, J. E.; RECKER, M. M. A simple yet robust caching algorithm based on dynamic access patterns. In: *In Proceedings of the Second International WWW Conference*. [S.l.: s.n.], 1994. p. 1039–1046.
- PODLIPNIG, S.; BÖSZÖRMENYI, L. A survey of web cache replacement strategies. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 35, p. 374–398, dez. 2003. ISSN 0360-0300.
- PPLIVE. *PPLive Homepage*. 2012. Disponível em: <<http://www.pplive.com>>. Acesso em: 01 out. 2010.
- PPSTREAM. *PPStream Homepage*. 2012. Disponível em: <<http://www.ppstream.com>>. Acesso em: 01 dez. 2010.
- QIU, L.; PADMANABHAN, V.; VOELKER, G. On the placement of web server replicas. In: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. [S.l.: s.n.], 2001. v. 3, p. 1587–1596 vol.3.

- R Development Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2010. Disponível em: <<http://www.R-project.org/>>. Acesso em: 01 jul 2011.
- RIPLEY, B. D. *Stochastic Simulation*. USA: Wiley, 1987. (Wiley Series in Probability and Statistics). ISBN 0-471-818884-4.
- VAKALI, A.; PALLIS, G. Content delivery networks: status and trends. *Internet Computing, IEEE*, v. 7, n. 6, p. 68–74, nov.-dec. 2003. ISSN 1089-7801.
- VAZIRANI, V. V. *Approximation Algorithms*. 1st. ed. Hardcover: Springer-Verlag, 2001. ISBN 978-3-540-65367-7.
- VLAVIANOS, A.; ILIOFOTOU, M.; FALOUTSOS, M. Bitos: Enhancing bittorrent for supporting streaming applications. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, p. 1–6, apr. 2006. ISSN 0743-166X.
- VU, Q. H. et al. *Peer-to-Peer Computing: principles and applications*. New York, USA: Springer Berlin Heidelberg, 2010. ISBN 978-3-642-03513-5.
- WANG, J. A survey of web caching schemes for the internet. *ACM SIGCOMM Computer Communication Review*, v. 29, n. 5, p. 36, out. 1999. ISSN 01464833.
- XIA, Z. et al. A field-based semantic p2p overlay network. *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, p. 1–4, oct. 2008.
- XU, D. et al. Analysis of a cdn-p2p hybrid architecture for cost-effective streaming media distribution. *Multimedia Systems*, Springer Berlin / Heidelberg, v. 11, p. 383–399, 2006. ISSN 0942-4962. 10.1007/s00530-006-0015-3.
- YAO, Z. et al. Modeling heterogeneous user churn and local resilience of unstructured p2p networks. In: *Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2006. p. 32–41. ISBN 1-4244-0593-9.
- YEE, T. W. The VGAM package. *R News*, v. 8, n. 2, p. 28–39, out. 2008.
- YOUNG, N. The k-server dual and loose competitiveness for paging. *Algorithmica*, Springer New York, v. 11, p. 525–541, 1994. ISSN 0178-4617. 10.1007/BF01189992.
- YOUTUBE. *Youtube statistic page*. 2012. Disponível em: <[http://www.youtube.com/t/press\\_statistics](http://www.youtube.com/t/press_statistics)>. Acesso em: 06 mar. 2012.