Cálculo da Distribuição da Banda para Servidores de Vídeo *

Carlo Kleber da S. Rodrigues[1] **, Rosa Maria Meri Leão**[1]

[1]Universidade Federal do Rio de Janeiro,
COPPE - Programa de Engenharia de Sistemas e Computação
Caixa Postal 68511, Rio de Janeiro, RJ 21941-972, Brasil

{kleber,rosam}@land.ufrj.br

Abstract. *Several bandwidth sharing techniques have been proposed to provide more scalability to video servers. The patching technique has several advantages such as simplicity of implementation and efficiency in terms of average bandwidth savings. However, the average bandwidth does not always provide an accurate estimation in order to guarantee a given level of QoS to the clients. We thus propose an analytical model to calculate the distribution of the number of streams generated by the patching technique, that is, the bandwidth distribution. Moreover, we investigate the use of this distribution to allocate the bandwidth in order to provide a given level of QoS to the clients, to estimate the QoS when some system parameter values change, and to dimension the server.*

Resumo. *Diversas técnicas de compartilhamento de banda têm sido propostas para aumentar a escalabilidade de servidores de vídeo. A técnica patching tem sido usada por sua simplicidade e eficiência em termos de requisitos de banda média. No entanto, o conhecimento da banda média nem sempre permite alocar banda aos clientes com o objetivo de prover uma certa qualidade. Neste trabalho é então proposto um modelo para obtenção da distribuição do número de fluxos gerados pela técnica patching, ou seja, a distribuição da banda. Além disso, mostramos a utilização desta distribuição para reservar banda de forma a prover uma determinada qualidade, prever a qualidade oferecida caso ocorra uma mudança nos valores dos parâmetros do sistema, e dimensionar o servidor.*

1. Introdução

Aplicações como ensino à distância e vídeo sob demanda têm sido alvo de estudos recentes na literatura. Uma maneira simples de atender os clientes destas aplicações é através do envio de um fluxo *unicast* para cada requisição. Como a banda do servidor é um recurso limitado, o uso de técnicas de compartilhamento de recursos é indispensável para prover maior escalabilidade ao servidor.

Pesquisas anteriores mostram que é possível obter reduções significativas dos valores de banda média do servidor através do emprego de técnicas de compartilhamento de banda. Estas técnicas podem ser do tipo orientada a requisições de clientes ou técnicas de difusão periódica. As técnicas orientadas a requisições (por exemplo, [1, 34, 11, 44, 24, 25, 26, 5, 35, 16, 18]) são reativas no sentido de que transmitem dados em resposta às solicitações dos clientes. Os dados são transmitidos em *multicast* e compartilhados por um grupo de clientes. Elas provêem serviço imediato e economizam o uso de banda por evitar transmissões redundantes de dados. As técnicas de difusão

*Este trabalho é parcialmente financiado pelo CNPq.

periódica (por exemplo, [3, 32, 47, 33, 36, 37, 41, 42, 43, 49]) são baseadas na transmissão periódica de segmentos do objeto, e garantem uma latência de serviço dentro de um intervalo de tempo estabelecido. Mais precisamente, a idéia é dividir o objeto em segmentos e transmití-los periodicamente em canais separados do servidor. Enquanto o cliente exibe um determinado segmento, existe a garantia de que o próximo segmento é recebido em tempo hábil para exibição contínua.

Estas técnicas são usualmente avaliadas em acordo com os valores médios de requisitos de banda associados. Pesquisas anteriores [28, 13, 44, 24, 27, 25, 45] apresentam estudos sobre a métrica banda média que é definida como a banda necessária para o servidor atender às requisições de clientes para um dado objeto. No entanto, a banda média pode não fornecer uma estimativa precisa para dimensionar o servidor. Isto é evidenciado em cenários onde a banda real requerida possui significativa variância.

Este trabalho volta-se para a técnica de compartilhamento *patching* [34]. Esta técnica é do tipo orientada a requisições e tem sido alvo de trabalhos bastante recentes como [31, 8, 30, 29, 48, 23, 40]. A técnica *patching* é simples e, para determinados valores de taxa de chegada de requisições, possui requisitos de banda semelhantes aos requisitos de outras técnicas mais sofisticadas da literatura como [25, 5].

Neste trabalho derivamos então a distribuição do número de fluxos concorrentes, ou seja, a banda, para a técnica de compartilhamento *patching*. Mostramos que a banda pode ser modelada através da distribuição binomial com parâmetros calculados em função da taxa de chegada de requisições e da duração dos objetos. Examinamos tanto o caso do servidor de único objeto como o caso do servidor de múltiplos objetos. Por fim, ilustramos o uso da distribuição obtida para dimensionar um servidor e reservar a banda com o objetivo de prover uma certa qualidade de serviço aos clientes.

O restante deste trabalho está organizado conforme mostrado a seguir. A Seção 2 apresenta o cenário de análise e conceitos básicos necessários para melhor compreensão deste trabalho. A Seção 3 é dedicada à apresentação do modelo para cálculo da distribuição da banda. Resultados analíticos e de simulação são apresentados na Seção 4. Na Seção 5, discutimos os trabalhos relacionados. Por último, as conclusões e os trabalhos futuros constituem a Seção 6.

2. Conceitos Básicos

Considere um servidor de vídeo e um grupo de clientes recebendo dados através da Internet deste servidor. Suponha que os clientes sempre requisitam o início do objeto e assistem ao mesmo até o final sem interrupções (acesso seqüencial), o *buffer* do cliente pode armazenar pelo menos metade do objeto requisitado, e a banda do cliente corresponde a duas vezes a taxa de exibição do objeto requisitado. Por último, a técnica *patching* é utilizada pelo servidor.

A técnica *patching* opera conforme explicado a seguir. O servidor escalona um fluxo *multicast* do objeto inteiro no momento da chegada do primeiro cliente, ou seja, da chegada da primeira requisição. Os clientes seguintes, que requisitam o mesmo objeto e chegam dentro de um limiar de tempo, denominado de *janela*, escutam o fluxo *multicast* e obtém a parte inicial do objeto, denominada de *patch*, através de fluxos *unicast* individuais. Os clientes que chegam após o término da janela iniciam uma nova janela e o processo é reiniciado. A Tabela 1 traz uma síntese dos principais parâmetros utilizados em nosso desenvolvimento.

Em [28, 10] foram propostos modelos bastante semelhantes para estimativa da banda média para a técnica *patching*. Nestes modelos foi suposto que as chegadas de

Símbolo	Definição
λ_j	taxa de chegada de requisições para o objeto O_j
D_j	duração total do objeto O_j, medida em unidades de tempo
W_j	janela para o objeto O_j, medida em unidades de tempo
N_j	número médio de requisições para o objeto O_j que chegam no intervalo D_j. Usualmente denotado de popularidade e computado por $N_j = \lambda_j D_j$
B_j	banda média do servidor para transmitir o objeto O_j usando a técnica *patching*, em unidades da taxa de transmissão do objeto O_j
d_j	unidade em que o objeto O_j é dividido para análise
T_j	duração total do objeto O_j, medida em unidades d_j
w_j	janela para o objeto O_j, medida em unidades d_j
u	número total de objetos no servidor

Tabela 1: Principais parâmetros.

requisições obedecem a um processo de Poisson de taxa λ_j para o objeto O_j. A banda média do servidor para transmissão do objeto O_j, medida em unidades da taxa de exibição do objeto, foi mostrada como sendo dada por:

$$B_j = \frac{D_j + \frac{\lambda_j W_j^2}{2}}{W_j + \frac{1}{\lambda_j}} \ , \tag{1}$$

onde D_j e W_j são a duração total e a janela do objeto O_j, respectivamente. D_j e W_j são expressos na mesma unidade de tempo.

O denominador da Equação 1 é o tempo médio entre as transmissões *multicast* consecutivas, isto é, a duração da janela mais o tempo médio até a chegada de uma nova requisição. O numerador corresponde a soma da duração média dos *patches* com a duração do filme inteiro.

Diferenciando-se a Equação 1 em relação a W_j e igualando o resultado a zero, obtemos o valor ótimo da janela:

$$W_{j,otimo} = \frac{\sqrt{2N_j+1}-1}{\lambda_j} \tag{2}$$

Substituindo este último resultado na Equação 1, obtemos a banda média do servidor para a técnica *patching* quando é usada a janela ótima:

$$B_{j,otimizada} = \sqrt{2N_j + 1} - 1 \ , \tag{3}$$

onde $N_j = \lambda_j D_j$ é o número médio de requisições de clientes que chegam durante o período de tempo D_j. N_j é comumente chamado de popularidade do objeto. Por simplicidade, chamaremos $W_{j,otimo}$ e $B_{j,otimizada}$ de W_j e B_j, respectivamente.

Quando a popularidade N_j é alta, as técnicas de compartilhamento do tipo difusão periódica são mais eficientes, em termos de economia de banda, que as técnicas orientadas a requisições de clientes, e quando ela é baixa ou média, as técnicas orientadas a requisições de clientes é que são mais eficientes [32]. É razoável então termos uma forma de classificar os objetos de acordo com sua popularidade N_j para podermos escolher a técnica de compartilhamento mais eficiente para a transmissão do objeto [28, 32]. Esta classificação é feita usualmente da seguinte forma [26]: para objetos de popularidade alta (isto é, $N \geq 100$), usamos técnicas do tipo difusão periódica; para objetos de popularidade baixa (isto é, $N < 10$) e, mais especialmente, para objetos de popularidade média (isto é, $10 \leq N < 100$), usamos técnicas do tipo orientada a requisições. Assim, como *patching* é uma técnica orientada a requisições, nossas análises neste trabalho se restringem principalmente a valores de N_j no intervalo de 10 a 100.

3. Cálculo da distribuição do número de fluxos concorrentes

Nesta seção apresentamos o modelo para o cálculo da distribuição do número de fluxos concorrentes gerados pela técnica de *patching*, ou seja, os requisitos de banda para esta técnica. Inicialmente obtemos o número de fluxos concorrentes considerando um único objeto e, em seguida, mostramos como o resultado pode ser facilmente estendido para um conjunto de objetos armazenados no servidor.

3.1. Distribuição do número de fluxos concorrentes para um objeto

Considere um servidor de vídeo e um objeto O_j, onde $j \in \{1, \ldots, u\}$ e u é o número total de objetos neste servidor. Suponha que O_j é dividido em unidades de comprimento d_j. Por exemplo, um objeto de duas horas pode ser dividido em 20 unidades de seis minutos cada, ou dividido em 100 unidades de 72 segundos cada. Definiremos então $T_j = \frac{D_j}{d_j}$, ou seja, T_j é igual ao número de unidades d_j em que o objeto O_j foi dividido. De forma similar, a janela w_j será expressa em função de d_j também, ou seja, $w_j = \frac{W_j}{d_j}$. Uma unidade d_j que tenha associada pelo menos uma chegada iniciará um novo fluxo de dados, isto é, um fluxo *multicast* do objeto inteiro ou um fluxo de parte do objeto (*patch*), no início da próxima unidade. Assuma ainda que as chegadas das requisições podem ser representadas por um processo de Poisson de taxa λ_j.

Seja $N(t)$ um processo estocástico que representa o número de janelas de *patching* iniciadas no intervalo $(0, t)$. $N(t)$ é um processo de renovação cujos pontos de renovação são os instantes em que o servidor escalona um fluxo *multicast* do objeto completo (ou seja, início de uma janela). Um ciclo de $N(t)$ tem duração média de $w_j + \frac{1}{\lambda_j}$, e portanto é neste intervalo que faremos nossa análise. Note que o intervalo $w_j + \frac{1}{\lambda_j}$ pode ser reescrito como $w_j + \frac{T_j}{N_j}$. Doravante usaremos esta última expressão para representar este intervalo, assim como a Equação 4 a seguir para representar a janela ótima de *patching*. Esta é uma forma alternativa da Equação 2 desde que $\frac{1}{\lambda_j} = \frac{T_j}{N_j}$.

$$w_j = \frac{T_j(\sqrt{2N_j + 1} - 1)}{N_j} \tag{4}$$

Nosso modelo pressupõe a divisão do objeto e do tempo em unidades de comprimento d_j. Portanto, o intervalo de duração $w_j + \frac{T_j}{N_j}$ pode ser representado como $[1, w_j + \frac{T_j}{N_j}]$. Examinaremos cada uma das suas $w_j + \frac{T_j}{N_j}$ unidades para determinarmos a distribuição do número de fluxos concorrentes em cada uma dessas unidades. Uma vez que tenhamos esse resultado, determinaremos então uma distribuição para representar o número de fluxos concorrentes em todo o intervalo. Como O_j pode representar qualquer objeto do servidor, daqui para frente omitiremos o índice j para simplificar a notação.

Considere a Figura 1. Nela são apresentados os fluxos *multicast* para transmissão do objeto completo no caso em que T é três vezes maior que $w + \frac{T}{N}$. Os fluxos de *patch* são omitidos para simplificar a figura. Este cenário pode ser visto como uma sucessão de intervalos de tamanho $w + \frac{T}{N}$. Note que os fluxos iniciados em um intervalo podem influenciar o número de fluxos concorrentes no intervalo seguinte. Isto ocorre pois nem os fluxos *multicast* nem os fluxos de *patch* são transmitidos na sua totalidade no intervalo em que são iniciados.

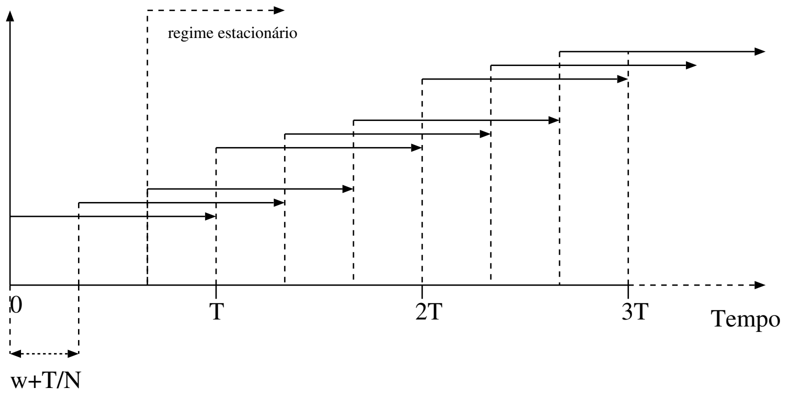


Figura 1: Fluxos *multicast* para transmissão do objeto completo

Analisando o intervalo $[1, w + \frac{T}{N}]$, podemos determinar quantas unidades deste intervalo transmitem fluxos que foram iniciados no intervalo anterior. Assuma que houve pelo menos uma chegada na última unidade do intervalo anterior. O fluxo de *patch* iniciado nesta unidade é o maior possível e, assim, irá se prolongar pelo maior número possível de unidades do intervalo seguinte. A Figura 2 ilustra esta situação. Desta forma subdividimos o intervalo $[1, w + \frac{T}{N}]$ em dois subintervalos:$[1, w - \frac{T}{N}]$ e $[w - \frac{T}{N} + 1, w + \frac{T}{N}]$. No primeiro subintervalo podemos ter fluxos iniciados no intervalo anterior, enquanto que no segundo subintervalo não. Note que quanto maior é o valor $\frac{T}{N}$, em relação ao valor w, menos unidades do intervalo seguinte terão fluxos que foram iniciados no intervalo anterior. No limite, quando $w = \frac{T}{N}$ (veja Equação 4), não há nenhum fluxo que seja iniciado em um intervalo e continue até o intervalo seguinte.

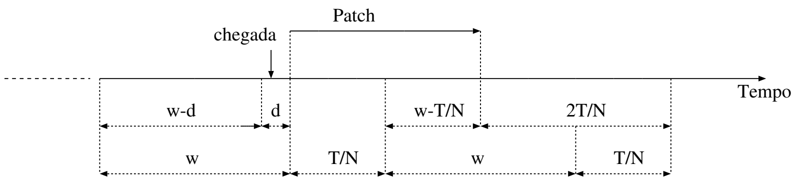


Figura 2: Fluxos transmitidos em mais de um intervalo.

Examinemos o número possível de fluxos concorrentes em cada unidade dos dois subintervalos definidos acima. O número de fluxos concorrentes em cada unidade será aproximado por uma constante mais uma variável aleatória. A constante é devida aos fluxos *multicast* e, através da formulação de Little [38], pode ser computada pela razão $\frac{T}{w+\frac{T}{N}}$. A Figura 1 ilustra um cenário onde esta constante é igual a três. A variável aleatória é devida aos fluxos de *patch* e portanto está diretamente associada ao processo de chegada de requisições.

Determinemos então a distribuição da variável aleatória em questão. Seja X_i esta variável aleatória, isto é, X_i denota o número de fluxos concorrentes de *patch* na i-ésima unidade do intervalo $[1, w + \frac{T}{N}]$, onde $i = 1, ..., w + \frac{T}{N}$. O Teorema 1 determina a distribuição de X_i. Por restrição de espaço, a prova deste teorema é apresentada apenas em uma versão mais extensa deste trabalho [21].

Teorema 1 *X_i tem distribuição binomial com parâmetros $p = 1 - e^{-\frac{N}{T}}$ e $n = n_i$, $i = 1, \ldots, w + \frac{T}{N}$. O valor n_i é calculado conforme descrito a seguir:*

1. *Caso 1: $w = \frac{T}{N}$, então:*

$$n_i = \begin{cases} \lceil \frac{i-1}{2} \rceil & \textit{, se } i \in [1, w] \\ \\ \lceil \frac{2w-(i-1)}{2} \rceil & \textit{, se } i \in [w+1, w+\frac{T}{N}] \end{cases} , \quad (5)$$

2. *Caso 2:* $w > \frac{T}{N}$*, então:*
   - $i \in [1, w - \frac{T}{N}]$*, então:*

$$n_i = \begin{cases} \lceil \frac{w-\frac{T}{N}}{2} \rceil + 1 & \textit{, se } (w - \frac{T}{N}) \textit{ e } i \textit{ são pares} \\ \\ \lceil \frac{w-\frac{T}{N}}{2} \rceil & \textit{, caso contrário} \end{cases} , \quad (6)$$

   - $i \in [w - \frac{T}{N} + 1, w]$*, então:*

$$n_i = \lceil \frac{i-1}{2} \rceil \quad (7)$$

   - $i \in [w + 1, w + \frac{T}{N}]$*, então:*

$$n_i = \lceil \frac{2w - (i-1)}{2} \rceil \quad (8)$$

Do Teorema 1, temos que o número de fluxos concorrentes de *patch* na i-ésima unidade do intervalo $[1, w + \frac{T}{N}]$ é representado por uma variável aleatória binomial X_i, $i = 1, \ldots, w+\frac{T}{N}$. As variáveis X_i possuem o mesmo parâmetro p, no entanto o parâmetro n_i pode variar dependendo do valor de i.

O próximo passo é determinarmos uma aproximação para a distribuição da variável aleatória que representa o número de fluxos concorrentes de *patch* no intervalo $[1, w + \frac{T}{N}]$ como um todo. Ao observamos os valores dos parâmetros n_i para cada uma das unidades do intervalo, constatamos que a variabilidade deste parâmetro é pequena para os diversos cenários considerados. Além disso, para a maioria das unidades, n_i oscila entre dois valores apenas. Por exemplo, na Figura 3(a), o parâmetro n_i varia no intervalo $[13, 17]$, sendo na maioria das unidades do intervalo igual a 13 ou 14. Já no exemplo da Figura 3(b), o parâmetro n_i varia no intervalo $[61, 66]$, sendo na maioria das unidades do intervalo igual a 61 ou 62.

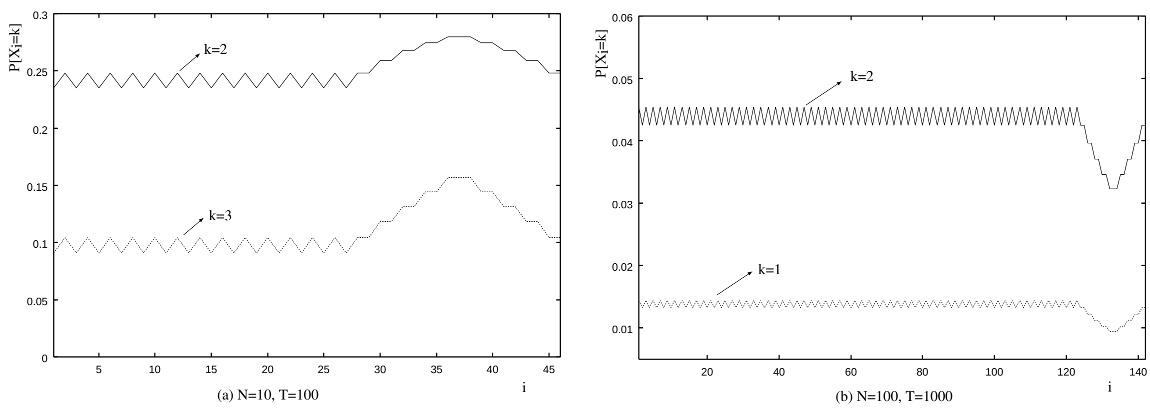


Figura 3: $P[X_i = k]$ **para cada uma das unidades do intervalo** $[1, w + T/N]$**.**

Como foi observado que o parâmetro n_i possui pouca variabilidade para diferentes valores de i, definiremos a variável aleatória binomial X que representa o número de fluxos concorrentes de *patch* no intervalo $[1, w + \frac{T}{N}]$ como um todo, com parâmetro $n = m$, sendo m igual a esperança de n_i. X possui portanto parâmetros $p = 1 - e^{-\frac{N}{T}}$ e $m = \lceil \frac{\sum_{i=1}^{w+T/N} n_i}{w+T/N} \rceil$, e a seguinte função distribuição:

$$F_X(k) = \sum_{j=0}^{k} \binom{m}{j} (1 - e^{-\frac{N}{T}})^j (e^{-\frac{N}{T}})^{m-j} \quad , 0 \le k \le m \tag{9}$$

O número total de fluxos concorrentes é a soma dos fluxos concorrentes para transmissão dos *patches* com os fluxos concorrentes para transmissão do objeto completo. O número de fluxos concorrentes para transmissão do objeto completo, conforme já vimos, é igual a um valor constante dado por $\frac{T}{w+\frac{T}{N}}$. Portanto, a distribuição do número total de fluxos concorrentes é a distribuição da variável aleatória X deslocada para a direita do valor $\frac{T}{w+\frac{T}{N}}$. Definiremos então, X_{total} como a variável aleatória que representa o total de fluxos concorrentes gerados pela técnica de *patching*. A distribuição de X_{total} é igual a:

$$P(X_{total} \le k) = \begin{cases} 0 & , 0 \le k \le v - 1 \\ \sum_{j=v}^{k} \binom{m}{l} (1 - e^{-\frac{N}{T}})^l (e^{-\frac{N}{T}})^{m-l} & , v \le k \le m + v \end{cases} \tag{10}$$

onde $k \in \{0, 1, 2, \ldots\}$ denota o número de fluxos concorrentes, $v = \frac{T}{w+\frac{T}{N}}$, e $l = j - v$.

A Figura 4 ilustra a função probabilidade de massa e a função distribuição complementar da variável aleatória X_{total}. Neste exemplo, temos que o número de fluxos concorrentes para transmissão do objeto completo é igual a 7, portanto a distribuição está deslocada deste valor.

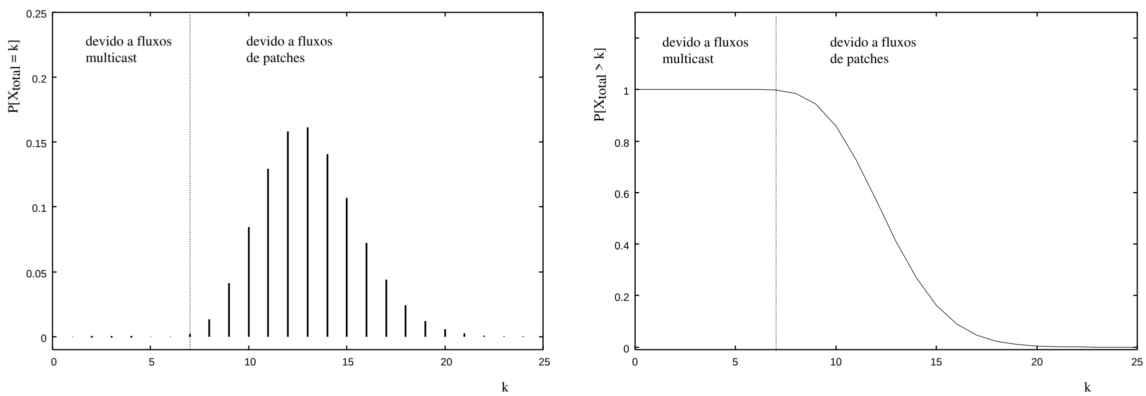


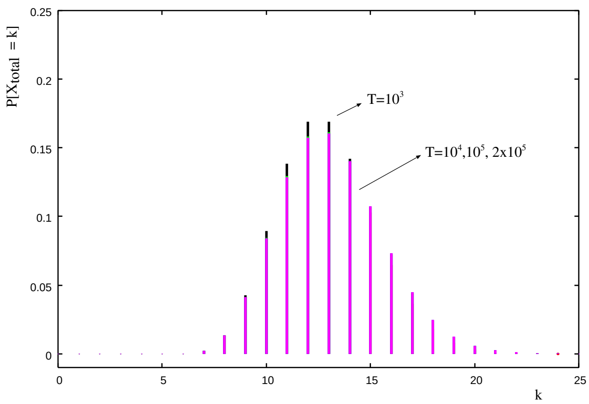
Figura 4: Distribuição do número de fluxos concorrentes

Um dos parâmetros que deve ser definido para o modelo é o tamanho d das unidades em que o objeto será dividido. Note que quanto menor for o valor de d mais próximo o modelo será da técnica de *patching* e maior será o valor de T, pois $T = \frac{D}{d}$. Para determinarmos o valor ideal de T, procedemos como mostrado a seguir. Iremos comparar a banda média (B_{media}), calculada a partir da esperança da variável aleatória X (veja Equação 11), com a banda média (B_j), computada a partir da Equação 3, e avaliar a diferença entre os dois valores usando o conceito de erro relativo. O erro relativo é definido como $E_r = \frac{|B_j - B_{media}|}{B_j}$. O objetivo deste experimento é identificarmos um valor de T tal que os erros relativos obtidos possam ser considerados não significativos.

Na Figura 5 plotamos os valores de erro relativo considerando as popularidades $N = 10, 25, 40, 75$ e 100. Está ilustrada também a função probabilidade de massa (PMF) do número de fluxos concorrentes para diferentes valores de T e $N = 100$. Desta figura, notamos, como já esperado, que quanto maior é o valor de T, menores são os valores dos erros relativos. Podemos observar que, a partir de $T = 10^4$, o erro relativo não apresenta uma diminuição significativa. Com relação a PMF do número de fluxos concorrentes, a partir também de $T = 10^4$, podemos notar que as curvas são muito semelhantes. Portanto, para N no intervalo de 10 a 100, podemos considerar 10^4 como um limite inferior para o parâmetro T.

$$B_{media} = m(1 - e^{-\frac{N}{T}}) + \frac{T}{w+\frac{T}{N}} \ , \tag{11}$$

onde a primeira parcela da soma é a média da variável aleatória X, e a segunda parcela é o total de fluxos concorrentes para transmissão do objeto completo.



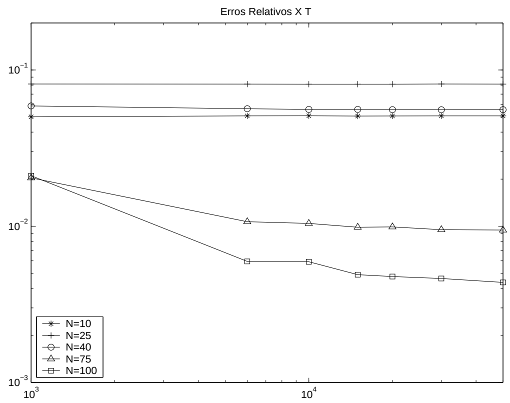


Figura 5: Estudo do valor de T

3.2. Distribuição do número de fluxos concorrentes para múltiplos objetos

Um servidor usualmente possui mais de um objeto. Isto motiva a derivação da distribuição do número de fluxos concorrentes para o caso em que existem chegadas de requisições para vários objetos. A distribuição do número de fluxos concorrentes do servidor considerando múltiplos objetos pode ser útil para o seu dimensionamento.

Na seção anterior vimos que as distribuições de X e X_{total}, respectivamente, dependem de dois parâmetros: N e T. Na análise a seguir, consideraremos T constante e igual ao limite inferior computado no final daquela seção, isto é, $T = 10^4$. Com isso, temos então que as distribuições de X e X_{total}, respectivamente, passam a depender exclusivamente de N.

Seja Y_N a variável aleatória que denota o número de fluxos concorrentes de *patch* devido a todos os objetos de popularidade N, $N \in \{1, 2, \ldots, N_{max}\}$. Da consideração do último parágrafo, temos então que Y_N é uma soma de variáveis aleatórias binomiais idênticas. Mais precisamente, Y_N é uma variável aleatória binomial com parâmetros $p = 1 - e^{-\frac{N}{T}}$ e $n = num_N \times \lceil \frac{\sum_{i=1}^{w+T/N} n_i}{w+T/N} \rceil$, onde num_N denota o número total de objetos de popularidade N no servidor [46].

Seja Y_{total} a variável aleatória que denota o número total de fluxos concorrentes de *patch* devido à transmissão de todos os objetos no servidor. Y_{total} é portanto a soma de N_{max} variáveis binomiais de diferentes parâmetros. Como não existe uma solução fechada para a distribuição desta soma, a distribuição de Y_{total} pode então ser calculada a partir da convolução das funções probabilidade de massa de Y_N, $N \in \{1, 2, \ldots, N_{max}\}$. Esta convolução pode ser eficientemente implementada usando a Transformada Rápida de Fourier [9].

Observe que Y_{total} denota o número de fluxos concorrentes devido a apenas fluxos concorrentes de *patch*. É necessário somar a contribuição constante de fluxos concorrentes *multicast* devido a transmissão do objeto completo. O valor que deve ser somado é igual a $\sum_{N=1}^{N_{max}} \frac{num_N T_N}{w_N + \frac{T_N}{N}}$, onde w_N e T_N são a janela e a duração, respectivamente, dos objetos com popularidade N, $N \in \{1, 2, \ldots, N_{max}\}$. A variável aleatória Y_{total} deslocada para a direita do valor constante definido acima representa o número de fluxos concorrentes necessários para a transmissão de todos os objetos armazenados no servidor.

4. Resultados

Esta seção apresenta os principais resultados obtidos durante a elaboração deste trabalho. Lembramos que, como a técnica *patching* é do tipo orientada a requisições de clientes, nossas análises se restringem principalmente a valores de popularidade N no intervalo de 10 a 100.

Como citado na seção 3.2, a distribuição do número de fluxos concorrentes obtida para múltiplos objetos pode ser usada objetivando o dimensionamento do servidor para notadamente alcançar uma determinada qualidade. Já a distribuição do número de fluxos concorrentes gerados por um objeto O_j pode ser usada para reservar a banda dinamicamente para os clientes recuperando um determinado objeto O_j. Esta reserva é baseada na distribuição do número de fluxos concorrentes gerados por O_j e naturalmente também na qualidade que deseja-se fornecer para aquele grupo de clientes. Iremos então na primeira parte desta seção estudar o comportamento das distribuições obtidas neste trabalho e mostrar como elas podem ser usadas para atingir os objetivos citados acima.

Em seguida realizaremos estudos do comportamento da distribuição do número de fluxos concorrentes gerados por um objeto considerando diferentes valores de janela para uma mesma popularidade. Estudos da literatura obtiveram um valor ótimo para a janela de *patching* em função do valor de N e da taxa de chegada de clientes λ (veja Equação 4). Uma das dificuldades de implementação da técnica é a estimativa precisa e em tempo real do valor de λ (e conseqüentemente do valor de N) para cálculo da janela ótima em um instante de tempo t. Portanto iremos investigar qual o comportamento da distribuição quando são atribuídos valores diferentes do valor ótimo para a janela.

4.1. Distribuição do número de fluxos concorrentes

A Figura 6 ilustra a função probabilidade de massa (PMF) do número de fluxos concorrentes para transmissão de um objeto considerando dois valores de popularidade. A partir desta figura, podemos estimar o número de fluxos concorrentes necessários e sua respectiva probabilidade de ocorrência.

A Figura 7 mostra a distribuição complementar do número de fluxos concorrentes. Nesta figura encontram-se resultados analíticos e de simulação (linhas sólidas e pontilhadas, respectivamente). A simulação foi realizada usando a ferramenta Tangram-II [12, 22]. Os resultados da simulação têm intervalos de confiança de 95% que estão dentro do limite de 5% dos valores reportados. Para os valores de popularidade considerados, veja que os resultados analíticos estão bem próximos dos resultados de simulação. Observe ainda que quanto maior é a popularidade, mais próximas são as curvas. Isto advém do fato de que quanto maior é a popularidade, menor é o número de distribuições binomiais com parâmetros n distintos no intervalo $w + \frac{T}{N}$. Portanto melhor é a aproximação da distribuição do número de fluxos concorrentes por uma variável aleatória binomial cujo parâmetro n é a média dos respectivos parâmetros n das distintas binomiais. A Figura 8 ilustra o número de distribuições binomiais distintas para $T = 10^4$. Para uma popularidade $N = 100$, temos 51 distribuições binomiais distintas, sendo que poderíamos ter

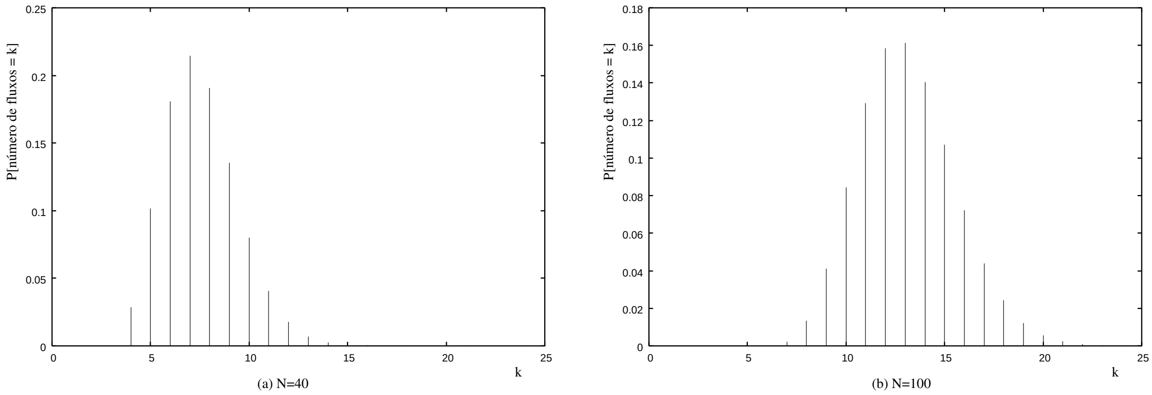


Figura 6: PMF do número de fluxos concorrentes: (a) N = 40 e (b) N = 100

até 1318 distribuições distintas; já para uma popularidade $N = 500$, temos apenas 11 distribuições distintas, sendo que poderíamos ter até 613 distribuições distintas.

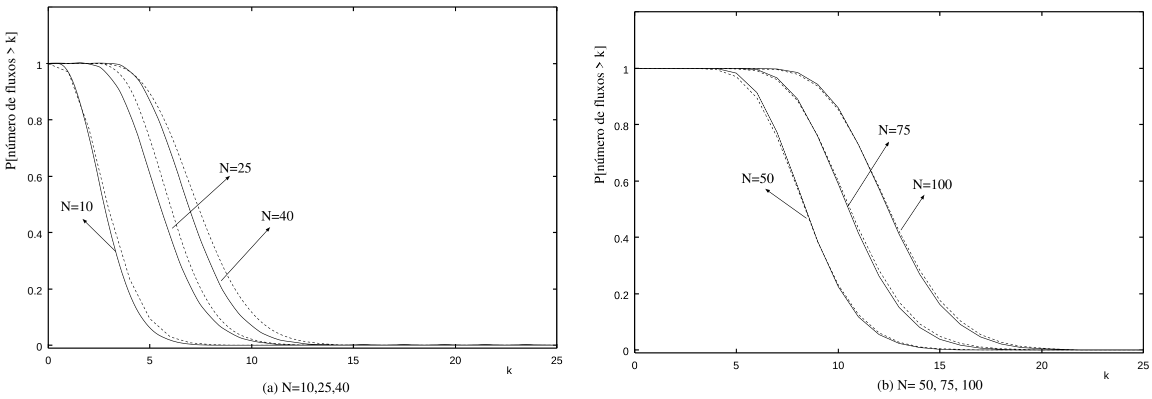


Figura 7: Distribuição do número de fluxos concorrentes: (a) N = 10, 25, 40 e (b) N = 50, 75, 100.

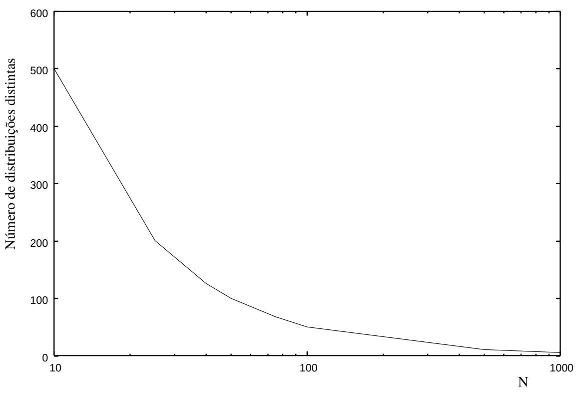


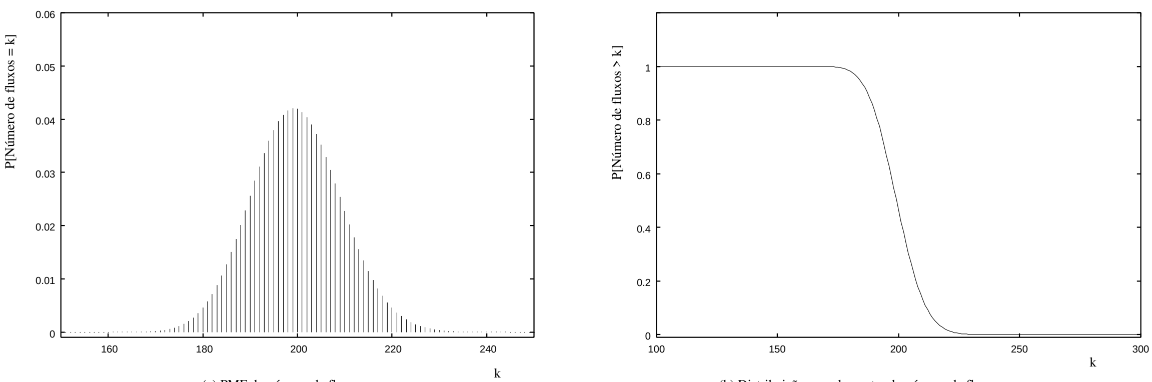
Figura 8: Número de distribuições binomiais distintas para $T = 10^4$.

A distribuição do número de fluxos concorrentes permite uma estimativa da banda do servidor para cada um dos objetos armazenados. Suponha que o objetivo seja reservar um certo número de canais k de forma que $P[X_{total} > k] = 10^{-3}$, ou seja, queremos que a probabilidade do número de fluxos concorrentes ultrapassar o valor k seja pequena. O valor de k que satisfaz a probabilidade acima pode ser facilmente obtido através da Tabela 2 para $N = 100$. Nesta tabela, na primeira coluna temos alguns valores estimados para a banda, na segunda coluna temos a probabilidade do número de fluxos concorrentes ultrapassar os valores estimados, e na terceira coluna temos o incremento necessário da banda média (calculada através da Equação 3) para que esta seja igual a banda estimada. Por exemplo, se toleramos uma probabilidade de exceder o valor estimado não maior que 4.6e-02, então precisamos de um aumento de 30.76% em relação a banda média. Através deste exemplo, podemos concluir que se a reserva for realizada baseada na banda média, existe uma probabilidade alta (da ordem de 0.4) de que sejam necessários mais do que 13 canais para transmissão do objeto para os clientes.

Banda estimada	P[**Núm. fluxos** $>$ **Banda estimada**]	**Incremento necessário**
13	4.1e-01	-
14	2.7e-01	7.69%
15	1.6e-01	15.38%
16	8.9e-02	23.08%
17	4.6e-02	30.76%
18	2.2e-02	38.46%
19	9.6e-03	46.15%
20	4.0e-03	53.85%
21	1.5e-03	61.54%

Tabela 2: Banda estimada para $N = 100$

Após a análise dos requisitos de banda para um objeto, passamos ao estudo dos requisitos de banda para um conjunto de objetos. Neste caso, um dos objetivos é dimensionar o servidor para obter uma determinada qualidade. Consideramos um cenário com 20 objetos sendo transmitidos. A distribuição Zipf [50], com *skew factor* 0.271 [39, 19, 2, 20], foi utilizada para determinar o número de objetos de cada popularidade (veja a Tabela 3). Na Figura 9(a), plotamos a função probabilidade de massa do número de fluxos concorrentes necessários para transmissão de múltiplos objetos. A partir desta figura vemos que o número de fluxos concorrentes ativos está aproximadamente no intervalo de 170 a 230. Estes números dão uma idéia da utilização da banda em um servidor de pequeno porte.



(a) PMF do número de fluxos

(b) Distribuição complementar do número de fluxos

Figura 9: PMF e distribuição do número de fluxos concorrentes para transmissão de múltiplos objetos

Na Figura 9(b) temos a distribuição complementar do número de fluxos concorrentes para transmissão de múltiplos objetos. A Tabela 4 apresenta a banda estimada para o servidor baseada na distribuição da Figura 9. Nesta tabela, na primeira coluna temos alguns valores estimados para a banda, na segunda coluna temos a probabilidade do número de fluxos concorrentes ultrapassar os valores estimados, e na terceira coluna temos o incremento necessário da banda média para que esta seja igual a banda estimada. A banda

N	Num. de objetos
10	2
25	2
40	2
50	3
75	4
100	7

Tabela 3: Cenário do servidor de múltiplos objetos

média foi calculada através da soma das bandas médias obtidas para cada uma das popularidades. Percebemos que com um aumento de 15.05% em relação a banda média, temos uma probabilidade de exceder a banda estimada igual a 8.6e-05. Comparando-se os valores obtidos para a Tabela 4 com aqueles obtidos para a Tabela 2, podemos notar que o incremento da banda com relação a banda média é muito maior para o caso de um único objeto para obter-se uma mesma probabilidade. Isto advém do fato de que a distribuição do número de fluxos concorrentes para transmitir múltiplos objetos possui menor coeficiente de variação do que a distribuição do número de fluxos concorrentes para transmissão de um único objeto. Portanto, para o caso de múltiplos objetos, o valor da banda média pode fornecer uma estimativa mais próxima do valor necessário para obter-se uma determinada qualidade. Note, porém, que no exemplo apresentado, se a banda média for usada para dimensionar o servidor, a probabilidade do número de fluxos exceder a banda é igual a 0.24.

Banda estimada	P[Núm. fluxos $>$ Banda estimada]	Incremento necessário
206	2.4e-01	-
212	9.2e-02	2.83%
216	4.2e-02	4.85%
223	1.0e-02	8.25%
227	2.5e-03	10.19%
237	8.6e-05	15.05%
248	9.1e-07	20.39%
258	6.9e-09	25.24%

Tabela 4: Banda estimada para um servidor de múltiplos objetos

4.2. Estudo do comportamento da distribuição para diferentes valores de janela

Considere o seguinte cenário. Um servidor está estimando dinamicamente o valor de λ para definição do valor da janela a ser usada. Suponha que inicialmente o valor obtido para N seja 75 e a janela seja definida como igual a janela ótima. A banda necessária para que P[Número de fluxos $> k$] seja da ordem de 10^{-2} é igual a 17 (veja Figura 11). Considere que a taxa de chegada de clientes teve uma alteração porém esta mudança ainda não foi detectada pelo servidor. Logo o servidor continuará operando com a janela definida para $N = 75$ e com a reserva de banda calculada para esta popularidade. Iremos avaliar então qual o impacto para o servidor e os clientes da não atualização do parâmetro λ. Dois aspectos precisam ser avaliados:(i) qual o impacto para o servidor de operar com uma janela de *patching* diferente da janela ótima ? Quais são os requisitos de banda para esta nova janela ? (ii) a reserva de banda feita considerando a popularidade de $N = 75$ garantirá que qualidade para o cliente ?

Em relação ao primeiro aspecto temos a análise a seguir. A Figura 10 ilustra a distribuição do número de fluxos concorrentes X_{total}, para $N = 50$ e $N = 100$, considerando uma variação da janela 20% acima e 20% abaixo das janelas ótimas para esses valores de popularidade. Podemos notar que as distribuições são muito semelhantes. Com este resultado podemos garantir que se o servidor não detectar a mudança na taxa de chegada e conseqüentemente não atualizar o valor da janela, a banda necessária é muito semelhante àquela definida para a janela ótima. Ou seja, a janela definida para o *patching* pode variar dentro de certos limites que esta variação não altera os requisitos de banda.

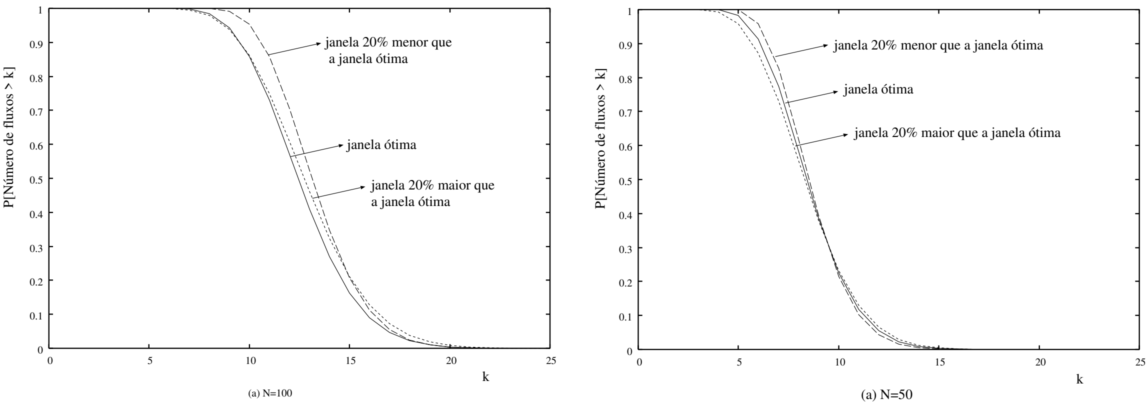


Figura 10: Distribuição do número de fluxos concorrentes para $N = 50$ e $N = 100$, e janelas variando em um intervalo de 20% em torno da janela ótima.

Agora passaremos a análise do segundo aspecto. Se a reserva de banda for realizada para uma dada popularidade e a taxa de chegada aumentar (diminuir) o que podemos notar através da Figura 11 é que a probabilidade do número de fluxos concorrentes serem superiores à banda reservada irá aumentar (diminuir). Suponha inicialmente $N = 75$ e uma reserva de banda igual a 17 canais. Neste caso temos $P[X_{total} > 17]$ =6.8e-03. Caso a taxa de chegada diminua de forma a que o novo valor de N seja 50 e o servidor não detecte esta alteração, teremos que $P[X_{total} > 17] =$ 1.1e-04. Neste caso a probabilidade de que seja necessário um número maior de canais além dos já reservados para este grupo de clientes irá diminuir. O que aumenta a qualidade oferecida para os clientes. No caso da taxa de chegada aumentar, e obtermos $N = 100$, temos que $P[X_{total} > 17] =$ 7.2e-02. A qualidade neste caso diminui.

Resumindo, o cálculo da distribuição do número de fluxos concorrentes permite que seja feita uma reserva de banda para um dado objeto ou um grupo de clientes com o objetivo de garantir uma certa qualidade. É possível também prever a qualidade que será oferecida para os clientes caso a taxa de chegada de requisições se altere e o servidor não detecte rapidamente esta mudança. Um outro resultado importante está relacionado ao valor da janela. Através dos resultados obtidos, foi constatado que alterações em torno de 20% do valor da janela ótima não alteram de forma significativa a distribuição do número de fluxos concorrentes.

5. Trabalhos Relacionados

Tam et al [4] foram os primeiros a formalmente examinarem a distribuição da banda de servidores de vídeo sob demanda. O trabalho deles direcionou-se para as seguintes três técnicas de compartilhamento do tipo orientada a requisições de clientes: Dynamic Fibonacci Tree [5, 6], Dyadic [35], e ERMT (Earliest Reachable Merge Target) [25]. Um estudo comparativo entre estas técnicas foi a principal contribuição. Contudo, nenhum resultado analítico foi apresentado naquela oportunidade, apenas simulações foram realizadas para consecução dos resultados.

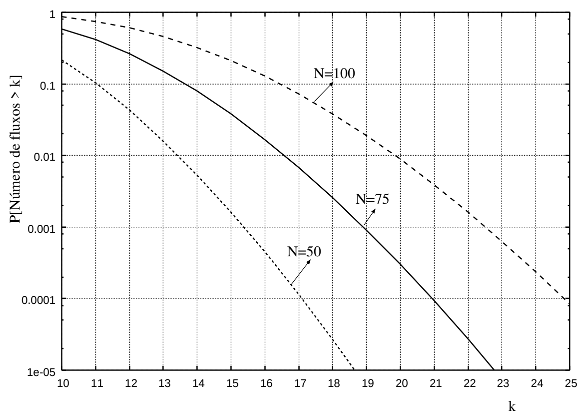


Figura 11: Distribuição do número de fluxos concorrentes para diferentes valores de N e mesma janela w.

Wong et al [15, 18] utilizaram a noção de razão competitiva [7] para estudar alguns problemas relacionados à análise e ao projeto de técnicas de compartilhamento em sistemas de vídeo sob demanda. Eles consideraram a métrica *banda máxima* para a estimativa da razão competitiva. A razão competitiva foi calculada a partir da razão entre o número máximo de fluxos concorrentes usados para transmissão de um objeto, considerando o emprego de uma determinada técnica, e o número máximo de fluxos concorrentes usados por um escalonamento ótimo off-line. Vemos então que, embora tenha sido derivado um limite superior para a utilização da banda, não há nenhuma informação a respeito da sua distribuição.

Por último, Souza e Silva et al [23] propuseram um modelo analítico para computar a distribuição da capacidade para transmitir todos os fluxos de *patch* iniciados em uma janela ótima (veja Equação 2) da técnica *patching*, e a partir disso a distribuiução da capacidade total foi inferida. Entretanto, como um fluxo de *patch* não necessariamente se extingue na janela em que ele foi iniciado, o modelo então proposto nem sempre fornece uma distribuição exata da capacidade total requerida. Além disso, nenhum resultado foi desenvolvido com relação ao caso do servidor de múltiplos objetos.

Portanto, nosso trabalho é, salvo engano, o primeiro a analiticamente derivar a distribuição da banda de um servidor de vídeo sob demanda, considerando o emprego da técnica *patching* tanto para o caso do servidor de único objeto, como também para o caso do servidor de múltiplos objetos.

6. Conclusões e Trabalhos Futuros

Neste trabalho, apresentamos um modelo que permite o cálculo da distribuição do número de fluxos (requisitos de banda) para um servidor de vídeo que usa a técnica de compartilhamento *patching*. Obtivemos a distribuição do número de fluxos tanto para o caso de um objeto, como para o caso de múltiplos objetos.

No caso de um objeto, mostramos que a distribuição do número de fluxos pode ser aproximada por uma binomial com parâmetros que dependem da popularidade do objeto. Usando simulações, obtivemos erros relativos bastante satisfatórios e observamos que, quanto maior é a popularidade do objeto, mais preciso é o modelo. A distribuição do número de fluxos para múltiplos objetos é uma soma de distribuições binomiais que pode ser calculada eficientemente através da Transformada Rápida de Fourier.

O uso das distribuições foi ilustrado através da elaboração de alguns cenários. Mostramos que é possível realizar uma reserva de banda para um dado objeto, ou para um grupo de clientes recuperando um objeto, com o objetivo de garantir uma certa qualidade. Esta reserva é feita baseada na distribuição do número de fluxos concorrentes. É possível também prever a qualidade que será oferecida para os clientes caso a taxa de chegada de requisições se altere e o servidor não detecte rapidamente esta mudança. Um outro resultado importante está relacionado ao valor da janela. Através dos resultados obtidos, foi constatado que alterações em torno de 20% do valor da janela ótima não alteram de forma significativa a distribuição do número de fluxos concorrentes.

Como trabalhos em andamento e futuros, apontamos três direções. A primeira consiste em derivar a distribuição da banda do servidor considerando outras técnicas orientadas a requisições de clientes e baseadas em esquemas de *merge* hierárquicos como [25, 26, 5, 35, 14, 16, 15, 18, 17]. Alguns resultados preliminares estão em [21]. A segunda diz respeito a estimativa da banda média e distribuição da banda do servidor considerando acesso não seqüencial do cliente. E a terceira refere-se a influência sobre a distribuição da banda de um processo de chegadas de requisições diferente de Poisson.

Referências

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. On optimal piggyback merging policies for video-on-demand systems. In *Proc. 1996 ACM SIGMETRICS Conf. On Measurement and Modeling of Computer Systems*, pages 200–209, 1996.

[2] J. M. Almeida, J. Krueger, D. L. Eager, and M. K. Vernon. Analysis of educational media server workloads. In *Proc. 11th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'01)*, pages 21–30, June 2001.

[3] K. C. Almeroth and M. H. Ammar. The use of multicast delivery to provide a scalable and interactive video-on-demand service. *IEEE Journal on Selected Areas in Communications*, 14(5):1110–1122, August 1996.

[4] A. Bar-Noy, G. Goshi, R. E. Ladner, and K. Tam. Comparison of stream merging algorithms for media-on-demand. In *Proc. of MMCN 2002*, San Jose, CA, January 2002.

[5] A. Bar-Noy and R. E. Ladner. Competitive on-line stream merging algorithms for media-on-demand. In *Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2001.

[6] A. Bar-Noy and R. E. Ladner. Efficient algorithms for optimal stream merging for media-on-demand. *SIAM Journal on Computing*, 33(5):1011–1034, 2004.

[7] A. Borodin and R. El-Yaniv. *On-line computation and competitive analysis*. Cambridge University Press, The Pitt Building, Trumpington Street, Cambridge, United Kingdom, 1998.

[8] M. K. Bradshaw, B. Wang, S. Sen, L. Gao, J. Kurose, P. Shenoy, and D. Towsley. Periodic broadcast and patching services - implementation, measurement and analysis in an internet streaming video testbed. *ACM Multimedia Journal SI on Multimedia Distribution*, 2003.

[9] E. O. Brigham. *The Fast Fourier Transform and its applications*. Prentice-Hall, Inc., New Jersey, 1988.

[10] Y. Cai, K. Hua, and K. Vu. Optimizing Patching Performance. In *Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, 1999.

[11] Y. Cai and K. A. Hua. An efficient bandwidth-sharing technique for true video on demand systems. In *Proc. of the 7th ACM Int'l Multimedia Conference (ACM Multimedia'99)*, pages 211–214, November 1999.

[12] R.M.L.R. Carmo, L.R. de Carvalho, E. de Souza e Silva, M.C. Diniz, and R.R. Muntz. Performance/Availability Modeling with the TANGRAM-II Modeling Environment. *Performance Evaluation*, 33:45–65, 1998.

[13] Y. Chai, K. Hua, and K. Vu. Optimizing patching performance. In *Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, January 1999.

[14] Wun-Tat Chan, Tak-Wah Lam, Hing-Fung Ting, and Wai-Ha Wong. A 5-competitive on-line scheduler for merging video streams. In *Proc. of the 15th Int'l Parallel and Distributed Processing Symposium*, 2001.

[15] Wun-Tat Chan, Tak-Wah Lam, Hing-Fung Ting, and Wai-Ha Wong. Competitive analysis of on-line stream merging algorithms. In *Proc. of the 27th Int'l Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 188–200, 2002.

[16] Wun-Tat Chan, Tak-Wah Lam, Hing-Fung Ting, and Wai-Ha Wong. A unified analysis of hot video schedulers. In *Proc. of the 34th ACM Symposium on Theory of Computing (STOC)*, pages 179–188, 2002.

[17] Wun-Tat Chan, Tak-Wah Lam, Hing-Fung Ting, and Wai-Ha Wong. On-line stream merging in a general setting. *Theoretical Computer Science*, 296(1):27–46, 2003.

[18] Wun-Tat Chan, Tak-Wah Lam, Hing-Fung Ting, and Wai-Ha Wong. On-line stream merging, max span, and min coverage. In *Proc. of the 5th Conference on Algorithms and Complexity (CIAC)*, pages 337–346, 2003.

[19] A. L. Chervenak. *Tertiary storage: An evaluation of new applications*. PhD thesis, CSD, University of California at Berkeley, 1994.

[20] M. Chesire, A. Wolman, G. Voelker, and H. Levy. Measurement and analysis of a streaming-media workload. In *USENIX Symposium on Internet Technologies and Systems (USITS)*, 2001.

[21] Carlo K. da S. Rodrigues and Rosa M. M. Leão. Bandwidth usage distribution of VoD servers using patching. Technical report, Departamento de Engenharia de Sistemas e Computação, UFRJ/LAND, Julho 2004.

[22] E. de Souza e Silva and R.M.M. Leão. Modelagem e Análise de Redes com o Conjunto de Ferramentas TANGRAM II. In *XXI Simpósio Brasileiro de Redes de Computadores(Workshop de Ferramentas)*, pages 897–904, Natal, Brasil, Maio 2003.

[23] E. de Souza e Silva, Rosa M.M. Leão, and Morganna C. Diniz. The Required Capacity Distribution for the Patching Bandwidth Sharing Technique. In *Proc. of IEEE ICC2004*, May 2004.

[24] D. Eager, M. Vernon, and J. Zahorjan. Minimizing bandwidth requirements for on-demand data delivery. In *Proc. 5th Int'l Workshop on Multimedia Information Systems (MIS'99)*, October 1999.

[25] D. Eager, M. Vernon, and J. Zahorjan. Optimal and efficient merging schedules for video-on-demand servers. In *Proc. 7th ACM Int'l Multimedia Conference (ACM Multimedia'99)*, pages 80–87, November 1999.

[26] D. Eager, M. Vernon, and J. Zahorjan. Bandwidth skimming: A technique for cost-effective video-on-demand. In *Proc. 2000 Multimedia Computing and Networking*, January 2000.

[27] D. L. Eager, M. C. Ferris, and M. Vernon. Optimized regional caching for on-demand data delivery. In *Proc. ISET/SPIE Conf. Multimedia Computing and Networking (MMCN'99)*, pages 301–316, January 1999.

[28] L. Gao and D. Towsley. Supplying instantaneous video-on-demand services using controlled multicast. In *Proc. IEEE Multimedia Computing Systems'99*, June 1999.

[29] C. Griwodz, M. Liepert, M. Zink, and R. Steinmetz. Tune to lambda patching. *ACM SIGMETRICS Performance Evaluation Review*, 27(4):20–26, 2000.

[30] D. Guan and S. Yu. A two-level patching scheme for video-on-demand delivery. *IEEE Transactions on Broadcasting*, 50(1):11–15, March 2004.

[31] Y. Guo, K. Suh, J. Kurose, and D. Towsley. P2cast: peer-to-peer patching scheme for VoD service. In *Proc. of the 12th World Wide Web Conference (WWW-03)*, May 2003.

[32] Ailan Hu. Video-on-demand broadcasting protocols: A comprehensive study. In *Proc. IEEE Infocom*, pages 508–517, 2001.

[33] K. Hua and S. Sheu. Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. In *Proc. ACM SIGCOMM*, September 1997.

[34] K. A. Hua, Y. Cai, and S. Sheu. Patching: A multicast technique for true video-on-demand services. In *Proc. 6th ACM Int'l Multimedia Conf. (ACM MULTIMEDIA'98)*, pages 191–200, 1998.

[35] E. G. Coffman Jr., P. Jelenkovic, and P. Momcilovic. Provably efficient stream merging. In *Proc. of 6th Int'l Workshop on Web caching and Content Distribution*, June 2001.

[36] L. Juhn and L. Tseng. Fast data broadcasting and receiving scheme for popular video service. *IEEE Transactions on Broadcasting*, 44(1):100–105, Mar 1998.

[37] L. Juhn and L. Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Transactions on Broadcasting*, 43(3):268–271, Sept 1997.

[38] Leonard Kleinrock. *Queueing Systems*, volume 1: Theory. John Wiley & Sons, New York, 1975.

[39] Huadong Ma and K. G. Shin. Multicast video-on-demand services. *ACM SIGCOMM Computer Communication Review*, 32(1), 2002.

[40] Bernardo C. M. Netto. Patching interativo: Um novo método de compartilhamneto de recursos para transmissão de vídeo com alta interatividade. Master's thesis, UFRJ, Departamento de Engenharia de Sistemas e Computação/LAND, Fevereiro 2004.

[41] J.-F. Paris, S. W. Carter, and D. D. E. Long. A low bandwidth broadcasting protocol for video on demand. In *Proc. of IEEE Int'l Conference on Computer Communications and Networks (IC3N'98)*, 1998.

[42] J.-F. Paris, S. W. Carter, and D. D. E. Long. A hybrid broadcasting protocol for video on demand. In *Proc. 1999 Multimedia Computing Networking Conference (MMCN'99)*, January 1999.

[43] J.-F. Paris, S. W. Carter, and D. D. E. Long. A simple low-bandwidth broadcasting protocol for video on demand. In *Proc. 8th Int'l Conference on Computer Communications and Networks (IC3N'99)*, October 1999.

[44] S. Sen, L. Gao, J. Rexford, and D. Towsley. Optimal patching schemes for efficient multimedia streaming. In *Proc. 9th Int'l Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'99)*, June 1999.

[45] H. Tan, D. Eager, M. Vernon, and H. Guo. Quality of service evaluations of multicast streaming protocols. In *Proc. ACM SIMETRICS 2002. Int'l Conference on Measurement and Modeling of Computer Systems*, June 2002.

[46] Kishor S. Trivedi. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley & Sons, Inc., second edition, New York, 2002.

[47] S. Viswanathan and T. Imielinski. Pyramid broadcasting for video-on-demand service. In *Proc. of the SPIE Multimedia Computing and Networking Conference*, pages 66–77, February 1995.

[48] Y. W. Wong and Jack Y. B. Lee. Recursive Patching - An efficient technique for multicast video streaming. In *Proc. 5th International Conference on Enterprise Information Systems (ICEIS) 2003*, pages 23–26, Angers, France, April 2003.

[49] R.M. Zafalão, N.L.S. Fonseca, and C.C. de Souza. The Polyharmonic Broadcasting Protocol subject to Bandwidth Constraints. In *Brazilian Computer Networks Conference*, pages 397–410, 2003. In Portuguese.

[50] G. K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, MA, 1949.