

CÁLCULO DE REQUISITOS DE CAPACIDADE PARA A TÉCNICA *PATCHING* DE COMPARTILHAMENTO DE RECURSOS

Edmundo de Souza e Silva, Rosa M. M. Leão e Morganna C. Diniz

Abstract - In the past few years several bandwidth sharing techniques for multimedia servers have been proposed in the literature in order to improve their scalability. One of these techniques is *patching*. The main advantages of *patching* are: (i) the simplicity of implementation and (ii) efficiency. In other words, considering a video with popularity varying from low to medium, the *patching* scheme requires, in the average, the same transmission bandwidth as other more sophisticated techniques. Recently, a simple expression to estimate the average bandwidth required by the *patching* technique has been obtained in the literature. However, averages values do not always provide an accurate estimation of the requirement bandwidth. We propose an analytical model to compute the distribution of the bandwidth required by the *patching* technique. We investigate the behavior of the distribution when we vary the size of the *patching* window. We also analyze the QoS obtained when the average bandwidth is used to dimension the server.

Keywords: *patching*, resource sharing techniques, multimedia server.

Resumo - Diversas técnicas de compartilhamento de recursos de servidores multimídia tem sido propostas na literatura. Dentre as propostas, a técnica de *patching* oferece como principais vantagens: a simplicidade de implementação; e requisitos de banda média na mesma ordem de grandeza de outras técnicas para filmes com popularidade variando de baixa a média. A capacidade média de transmissão para a técnica de *patching* pode ser estimada por uma expressão simples, obtida recentemente na literatura. No entanto, nem sempre valores médios fornecem a informação adequada para o dimensionamento do servidor. Neste trabalho é proposto um modelo analítico que permite calcular a distribuição dos requisitos de capacidade para a técnica de *patching*. Com os resultados obtidos, investigou-se a sensibilidade da função distribuição da capacidade necessária ao método de *patching* em relação ao tamanho da janela desta técnica. Investigou-se também a adequação ou não da capacidade média para fornecer a QoS desejada.

Palavras-chave: *patching*, técnicas de compartilhamento de recursos, servidor multimídia.

1. INTRODUÇÃO

O desenvolvimento de novas tecnologias como redes de alta velocidade, comunicação sem fio, técnicas de compressão de dados e CPU's com alta capacidade de processamento possibilitou a criação de infra-estruturas capazes de suportar uma grande variedade de aplicações multimídia. Dentre essas aplicações podemos citar, vídeo e áudio sob demanda, ambientes de realidade virtual, bibliotecas digitais, jogos eletrônicos, entre outras.

Uma característica comum das aplicações multimídia é que estas possuem requisitos de tempo estritos, ou seja, os dados devem ser tocados no receptor obedecendo a um escalonamento com pouca margem de flexibilidade. Os requisitos em termos de capacidade de transmissão e armazenamento também são bastante altos. Portanto, o projeto dessas aplicações no cenário da Internet torna-se um grande desafio.

O projeto de servidores multimídia tem sido amplamente discutido na literatura. Um dos aspectos considerados com frequência é o estudo de técnicas para o compartilhamento da capacidade de transmissão do servidor. Considere um servidor que atende 1500 clientes cada um recebendo um filme com taxa média de 1.5 Mbps. A banda do canal de saída deste servidor para suportar estes clientes deve ser igual a 2.25 Gbps. O alto requisito de banda pode tornar o custo deste servidor proibitivo.

Várias técnicas tem sido propostas na literatura para o compartilhamento de um mesmo fluxo por diversos clientes tais como [1]: *batching* [2], *stream tapping* [3], *patching* [4, 5], *controlled multicast* [6] e *hierarchical stream merging* (HSM) [7, 8]. Nessas técnicas os clientes fazem pedidos para o servidor e este envia o fluxo para os clientes. Na técnica de *batching* clientes que chegam em uma determinada janela de tempo formam um grupo que irá compartilhar a única transmissão do fluxo iniciada imediatamente após o instante final da janela. A desvantagem desta técnica é a latência introduzida para os clientes. As técnicas de *stream tapping*, *patching* e *controlled multicast* são similares e foram introduzidas de forma a minimizar a latência introduzida pela técnica de *batching*. Uma janela é definida e todos os clientes que chegam durante esta janela recebem duas transmissões simultâneas: uma corresponde ao filme completo (esta é a transmissão que iniciou a janela) e a outra corresponde a parte do filme que o cliente perdeu (o *patching*). Desta forma cada um dos clientes da janela é agrupado com o cliente que iniciou a janela em um instante de tempo que é igual ao tempo da sua chegada somado a duração do *patching*. Na técnica de HSM os clientes são agrupados hierarquicamente. A principal diferença desta técnica em relação ao *patching* é que um cliente pode ser agrupado com qualquer outro e não so-

Edmundo de Souza e Silva é professor da COPPE/PESC e do Departamento de Ciência da Computação/IM da UFRJ, Rosa M. M. Leão é professora da COPPE/PESC da UFRJ e Morganna C. Diniz é professora das Faculdades IBMEC/RJ, Rio de Janeiro, RJ, Brasil. E-mails: {edmundo, rosam, morganna}@land.ufrj.br.

mente com aquele que iniciou a janela. Uma outra classe de técnicas de compartilhamento de recursos são as denominadas de *broadcast* periódico [9, 10, 11, 12] onde o servidor envia os blocos de dados periodicamente em seus canais de saída.

Dentre as técnicas orientadas aos pedidos dos clientes a técnica de *patching* possui algumas vantagens: não introduz o retardo adicional existente na técnica de *batching* e é bem menos complexa que a técnica HSM. Em [13] foi feito um estudo visando analisar os requisitos de capacidade de diversas técnicas de compartilhamento de recursos. A técnica de *patching* apresentou requisitos semelhantes aos da técnica HSM para filmes com popularidade baixa e média. Por esses motivos esta técnica está sendo atualmente implementada no servidor RIO desenvolvido no âmbito do projeto COMIT [14, 15].

Estudos da literatura obtiveram a capacidade média necessária para atender clientes requisitando um certo filme, em função da popularidade do filme, quando a técnica de *patching* é usada. Expressões simples para a capacidade média de transmissão e para a janela ótima da técnica foram determinadas em função da taxa de chegada de clientes e do tempo de exibição do filme [5, 6, 13]. Embora a capacidade média seja útil em uma primeira etapa do dimensionamento dos recursos, nem sempre valores médios fornecem a informação adequada para tal dimensionamento. Isso é evidente, por exemplo, se a demanda por capacidade variar muito em torno do valor médio. Neste caso uma demanda maior que a média irá impactar negativamente na QoS fornecida aos usuários. (Por exemplo, usuários não atendidos por falta de banda ou retardos inaceitáveis na entrega de blocos do fluxo sendo transmitido.) É importante então tentar obter métricas tais como momentos ou mesmo a distribuição dos requisitos por banda. O foco deste trabalho é o de tentar responder perguntas tais como: Qual a probabilidade da demanda por banda ser maior que x vezes a demanda média? A demanda por recursos implementando a janela ótima obtida com a banda média fornecerá a qualidade de serviço almejada?

Foi desenvolvido um modelo analítico que fornece como resultado a probabilidade da capacidade total requisitada pela técnica de *patching* ultrapassar um dado valor. Em outras palavras foi obtido um algoritmo recursivo para calcular a distribuição do número total de bytes requisitados pelos clientes a cada ciclo da técnica e, a partir dessa métrica, inferir a distribuição da capacidade total necessária. Uma outra medida obtida foi o intervalo entre a ocorrência de ciclos onde a duração da soma dos *patches* é maior que um certo valor. Com os resultados obtidos, investigamos a sensibilidade da função distribuição da capacidade do método em relação a janela, e também a adequação ou não da capacidade média para fornecer a QoS desejada.

Na Seção 2 é feita uma breve descrição da técnica de *patching*. O modelo e as medidas de interesse estão definidos na Seção 3 e o algoritmo proposto para o cálculo da distribuição da capacidade é apresentado na Seção 4. Alguns resultados são apresentados na Seção 5 e a Seção 6 conclui o artigo.

2. A TÉCNICA DE PATCHING

Para entender o modelo proposto é importante detalhar o funcionamento da técnica de *patching*. Nesta técnica é definida uma janela w para um determinado fluxo f que pode representar a transmissão de um filme. A janela é o intervalo mínimo entre o instante inicial de duas transmissões completas e sucessivas do fluxo f . O primeiro pedido de um cliente para o fluxo f inicia a janela e a transmissão *multicast* de f . Todos os clientes que chegam dentro da janela irão receber o fluxo iniciado pelo primeiro cliente e armazená-lo em um *buffer*. Para obter a parte inicial do fluxo f , chamada de *patch*, que já foi transmitida pelo servidor, é estabelecido um outro canal entre o cliente e o servidor. O *patch* tem a duração igual ao tempo entre a chegada do primeiro cliente da janela e o tempo da chegada do cliente que iniciou o *patch*. Os dados que chegam através deste segundo canal são imediatamente apresentados para o cliente. Quando os dados recebidos através do *patch* terminam, o cliente começa a consumir os dados armazenados no *buffer*. Portanto, os clientes devem ser capazes de armazenar os dados recebidos do fluxo principal e do *patch* assim como de receber dados em dois canais simultaneamente.

A Figura 1 ilustra o mecanismo de *patching*. Suponha que três pedidos chegam dentro da primeira janela w . O pedido r_0 inicia a transmissão multicast do fluxo f , r_1 inicia a transmissão do *patch* de duração $(t_1 - t_0)$ para o segundo cliente e r_2 inicia a transmissão do *patch* de duração $(t_2 - t_0)$ para o terceiro cliente. O pedido r_3 inicia uma nova janela e a partir deste momento o procedimento se repete.

O tamanho da janela w pode melhorar o desempenho da técnica. Se w é muito grande, a maior parte da capacidade do servidor é usada para o envio de *patches*, por outro lado se w é muito pequena, não haverá compartilhamento de fluxos. Modelos foram propostos em [5, 6] visando o cálculo da janela que minimiza os requisitos de capacidade média do servidor e em [13] é feito um estudo comparativo da capacidade de diversas técnicas de compartilhamento de recursos.

Nestes modelos é suposto que as chegadas dos pedidos dos clientes obedecem a um processo de Poisson com taxa λ para o fluxo f . Logo pode ser mostrado que os requisitos de capacidade do servidor são

$$C = \frac{T + (\lambda w^2)/2}{w + 1/\lambda}, \quad (1)$$

sendo T o tamanho total do fluxo f e w a janela do *patching*.

Diferenciando-se a expressão (1) em relação a w , obtém-se o valor da janela ótima que é dado por

$$w_o = \frac{\sqrt{2N + 1} - 1}{N}, \quad (2)$$

sendo $N = \lambda T$.

Substituindo-se o valor da janela ótima em (1), temos que os requisitos de capacidade média do servidor são dados por

$$C_{w_o} = \sqrt{2N + 1} - 1. \quad (3)$$

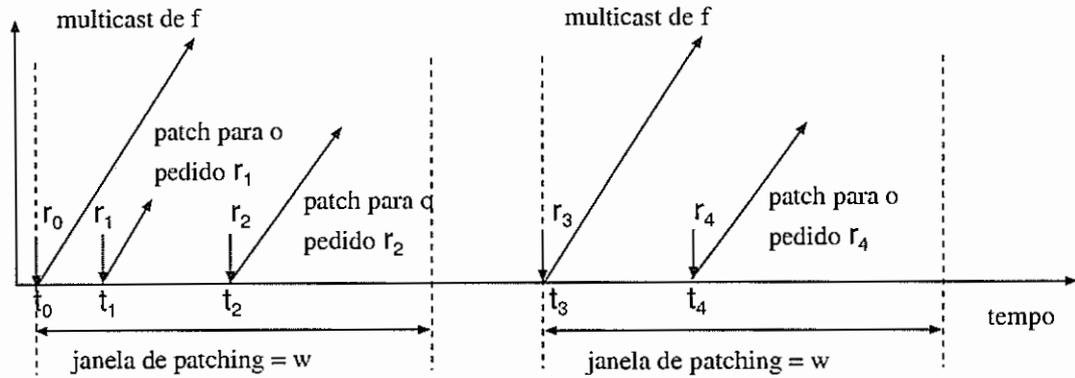


Figure 1. A técnica de patching.

3. O MODELO

3.1 PRELIMINARES

Para que o algoritmo proposto neste trabalho possa ser entendido, é necessário que se faça uma breve revisão sobre combinação linear de estatísticas de ordem.

Sejam V_1, V_2, \dots, V_n n variáveis aleatórias independentes e identicamente distribuídas (iid) cuja função distribuição cumulativa $F_Y(y)$ é contínua e estritamente crescente. Definimos as variáveis aleatórias $V_{(1)}, V_{(2)}, \dots, V_{(n)}$ por

$$V_{(i)} = a \text{ } i\text{-ésima menor entre as variáveis } V_1, V_2, \dots, V_n.$$

Obviamente, $V_{(1)} \leq V_{(2)} \leq \dots \leq V_{(n)}$. $V_{(i)}$ é denominada a estatística de i -ésima ordem da amostra (V_1, V_2, \dots, V_n) e $(V_{(1)}, V_{(2)}, \dots, V_{(n)})$ é denominado o conjunto de estatísticas de ordem de tamanho n associado a amostra (V_1, V_2, \dots, V_n) .

Em particular, sejam $U_1(t), \dots, U_n(t)$ variáveis aleatórias uniformes iid no intervalo $(0, t)$, e suponha que $U_{(0)}(t) = 0$ e $U_{(n+1)}(t) = t$. Como a variável aleatória $U_{(j)}(t)$ associada ao intervalo $(0, t)$ tem a mesma distribuição de $tU_{(j)}(1)$ associada ao intervalo $(0, 1)$, sem perda de generalidade, podemos supor que $t = 1$. Para simplificar a notação consideremos $U_{(j)} \stackrel{\text{def}}{=} U_{(j)}(1)$ no restante do artigo. A Figura 2 mostra um exemplo de estatísticas de ordem com $n = 5$.

A combinação linear de estatísticas de ordem é definida por

$$G(n) = \sum_{j=1}^{n+1} a_j U_{(j)}, \quad (4)$$

sendo cada a_j um número real.

Uma propriedade importante para o modelo a ser desenvolvido é a relação de estatísticas de ordem com a ocorrência de eventos de um processo de Poisson. Sejam $\tau_1, \tau_2, \dots, \tau_n$ os instantes de tempo onde eventos de um processo de Poisson ocorrem no intervalo $(0, t)$ condicionado a existência de n eventos em $(0, t)$. Então, as variáveis $\tau_1, \tau_2, \dots, \tau_n$ são distribuídas como as estatísticas de ordem de n variáveis aleatórias uniformes em $(0, t)$ [16].

3.2 O TEMPO TOTAL DOS PATCHES

Os modelos abordados na Seção 2 consideram os instantes de chegada dos pedidos dos clientes distribuídos como um

processo de Poisson. Faremos essa mesma suposição, e seja λ a taxa de chegada de clientes. Considere uma janela de patching de duração w , iniciada no tempo τ_0 pelo primeiro pedido de um cliente após o término da janela anterior a τ_0 . Sejam $\tau_1, \tau_2, \dots, \tau_n$ os instantes das chegadas dos pedidos subsequentes de clientes no intervalo $(0, w)$, condicionado a terem sido feitos n pedidos no intervalo (O intervalo é aberto e portanto os n pedidos não incluem aquele do cliente que chegou em τ_0 e deu início a nova janela de patching.) Pela identificação do processo de Poisson com estatísticas de ordem feita acima, e como w é um valor fixo (igual a janela da técnica de patching), fica claro que os instantes dos pedidos são distribuídos de acordo com as estatísticas de ordem de um conjunto de n variáveis aleatórias uniformes em $(0, w)$.

A Figura 3 ilustra o início e fim do período de uma janela. Na figura um período de distribuição de filme foi iniciado por um pedido de cliente. Imediatamente após essa primeira requisição de fluxo inicia-se um intervalo de duração igual a w . Durante $(0, w)$ três outras requisições chegam. A distribuição de τ_1 é igual a de $U_{(1)}$, a de τ_2 é igual a de $U_{(2)}$, etc. Observando ainda a Figura 3 podemos notar que a duração do i -ésimo patch enviado para $(i + 1)$ -ésimo cliente é igual a $U_{(i)}$. Portanto, a soma dos patches no intervalo $(0, w)$ condicionada a n pedidos no intervalo é igual a soma das estatísticas de ordem de n variáveis aleatórias uniformes iid subtraindo o valor w da janela (ver equação 4). Em outras palavras, a distribuição dos requisitos de capacidade relativos aos patches pode ser obtida a partir da distribuição de uma combinação linear de estatísticas de ordem de variáveis aleatórias uniformes iid.

Na seção seguinte indicaremos como calcular algumas medidas de interesse baseando-se na combinação linear de estatísticas de ordem e em teoria de renovação. Para simplificar a notação omitiremos das definições abaixo a dependência do valor da janela w .

3.3 MEDIDAS DE INTERESSE

Na Seção 3.2 foi mostrado que o tempo total dos patches necessários ao atendimento dos clientes em uma janela, condicionado a existência de n pedidos, é igual a soma das estatísticas de ordem de um conjunto de n variáveis aleatórias iid. Seja C a taxa de transmissão de um filme (Assumimos constante, por exemplo 1.5 Mbps para um filme codificado

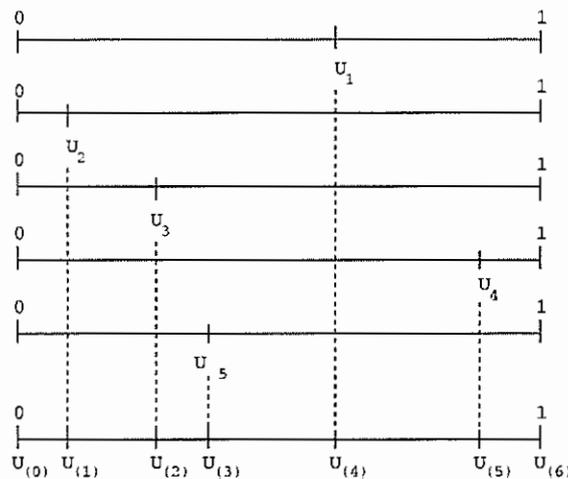


Figure 2. Um conjunto de estatísticas de ordem uniforme em (0, 1).

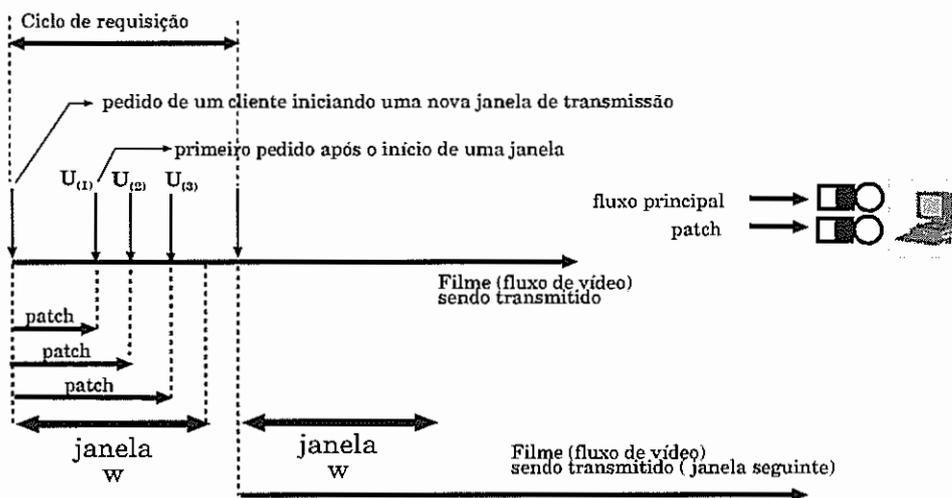


Figure 3. Um período de *patching* (ou período de distribuição de filme).

em MPEG2.) Considere um *ciclo de requisição* aquele entre dois pedidos de clientes que iniciam um novo fluxo completo de transmissão (ver Figura 3), e seja $X(t)$ o processo estocástico que indica o total de bits requisitados a cada ciclo em $(0, t)$ (igual ao total de tempo do filme requisitado mais os *patches*). $X(t)$ é então um processo de renovação. Seja o k -ésimo ciclo dado pelo intervalo (η_{k-1}, η_k) , Δ_k o comprimento deste intervalo, Ψ_k a soma de todos os *patches* no intervalo. Suponha que a cada ciclo uma recompensa seja ganha, cujo valor será indicado abaixo. Seja R_k a recompensa ganha no k -ésimo ciclo, e independente do tamanho do intervalo, e $R(t) = \sum_{k=1}^{N(t)} R_k$, sendo $N(t)$ o número de ciclos em $(0, t)$. Dos resultados da teoria de renovação com recompensas (ver, por exemplo, Teorema 3.6.1 em [17]):

$$\lim_{t \rightarrow \infty} \frac{R(t)}{t} \text{ com probabilidade 1} = \frac{E[R_k]}{E[\Delta_k]} \text{ (para qualquer } k) = \frac{E[R]}{E[\Delta]} \quad (5)$$

Defina uma variável aleatória indicadora I_k :

$$I_k = \begin{cases} 1 & \text{se } \Psi_k > v \\ 0 & \text{caso contrário.} \end{cases}$$

Igualando $R_k = I_k$, temos como consequência que, como já foi mostrado na seção anterior, os R_k são iids. Então pela equação (5) a taxa limite $\sigma(v)$ do número de intervalos com valor do *patch* maior que v será dada por

$$\sigma(v) = \frac{E[I]}{E[\Delta]} = \frac{P[\Psi > v]}{w + 1/\lambda} \quad (6)$$

A equação (6) indica que, a cada $1/\sigma(v)$ um *patch* de tamanho maior que v ocorre, ou no mínimo vC bits de *patch* tem que ser transmitidos, além do fluxo (filme) iniciado no ciclo.

Defina uma variável aleatória Θ_k igual a duração de um ciclo cuja soma dos *patches* requisitados neste ciclo seja maior que v . Seja a probabilidade $P_{\Theta}(v)$ do sistema encontrar-se em um intervalo (η_{k-1}, η_k) qualquer tal que $\Psi_k > v$. Ainda da teoria de renovação,

$$P_{\Theta}(v) = \frac{E[\Theta_k]}{E[\Delta_k]} = \frac{P[\Psi_k > v] \times E[\Delta_k]}{E[\Delta_k]} = P[\Psi > v] \quad (7)$$

Em outras palavras, a probabilidade de um ciclo qualquer requerer mais do que vC bits de *patch* é obtida por (7) acima.

Um ciclo então, requisita no mínimo $(T + v)C$ bits com probabilidade $P_{\Theta}(v)$. Um servidor de vídeo transmite blocos de informação de tamanho fixo aos seus clientes, por exemplo de tamanho 512 K bytes, como é usado no servidor RIO [18, 19]. Supondo a taxa de $C = 1.5$ Mbps, blocos são enviados a intervalos de 2.7 segundos, isto é, de comprimento uma a duas ordens de grandeza menor em relação ao tamanho médio dos ciclos (que em geral são na ordem de poucos a vários minutos). Portanto, os eventos de transmissão de blocos podem ser considerados como um fluxo contínuo em relação aos eventos de início de ciclo. É importante notar também que, a cada ciclo, estão sendo transmitidos os blocos relativos a filmes iniciados em outros ciclos além dos gerados no próprio ciclo. Seja C_f taxa limite (quando $t \rightarrow \infty$) para transmitir os filmes gerados a cada ciclo. Então,

$$C_f = \frac{T}{w + 1/\lambda} C,$$

sendo T o tempo de transmissão de um filme.

Como $P_{\Theta}(v)$ é a probabilidade de um ciclo gerar mais do que v bits de patch, e supondo que em média os blocos gerados num ciclo são transmitidos no mesmo ciclo (hipótese conservadora), então a capacidade adicional em um ciclo C_p é tal que

$$C_p > \frac{v}{w + 1/\lambda} C$$

com probabilidade $P_{\Theta}(v)$, e a capacidade total é $C_T > C_f + C_p$ com probabilidade $P_{\Theta}(v)$.

As medidas de interesse acima podem então ser obtidas de $P[\Psi > v]$, calculado a partir da soma das estatísticas de ordem conforme mostrado anteriormente. Na seção seguinte detalharemos o algoritmo usado para calcular as medidas de interesse baseado nos resultados de [20] para o cálculo da distribuição de uma combinação linear de estatísticas de ordem.

4. O ALGORITMO

Na seção anterior foi mostrado como calcular a probabilidade limite (quando $t \rightarrow \infty$) $P_{\Theta}(v)$ da capacidade total C_T requisitada ser maior que um certo valor, e também a taxa $\sigma(v)$. Estas medidas estão em função do cálculo da combinação linear de n estatísticas de ordem uniformes $G(n)$. Nesta seção mostramos como obter $G(n)$ e então as medidas de interesse.

Seja $Y_j = U_{(j)} - U_{(j-1)}$, $j = 1, \dots, n + 1$. Os Y_j são variáveis aleatórias intercambiáveis [17]. Substituindo a fórmula $U_{(j)} = \sum_{i=1}^j Y_i$ em (4) e reorganizando os somatórios temos

$$G(n) = \sum_{j=1}^{n+1} d_j Y_j, \quad (8)$$

sendo $d_j = a_j + \dots + a_{n+1}$, $j = 1, \dots, n + 1$.

No caso do modelo tratado neste trabalho, a variável de interesse é

$$G'(n) = \sum_{j=1}^{n+1} d_j Y_j - 1, \quad (9)$$

e ainda $a_1 = \dots = a_{n+1} = 1$, logo $d_1 \geq \dots \geq d_{n+1} \geq 0$. Note que, neste caso,

$$G'(n) = \sum_{j=1}^{n+1} U_{(j)} - 1 = \sum_{j=1}^n U_{(j)} = \sum_{j=1}^{n+1} (n+1-j) Y_j. \quad (10)$$

Da definição de d_j acima, podemos observar que $d_j \neq d_i$ se $j \neq i$, e cada intervalo Y_j está associado a uma constante d_j diferente das outras constantes d_i , $i \neq j$.

Definição 1

Seja o vetor $\mathbf{k}(m, n+1) = \langle k_1, \dots, k_{n+1} \rangle$ de dimensão $n+1$, onde qualquer entrada k_i assume apenas um dos valores 0 ou 1, com m entradas consecutivas iguais a 1, e o restante das entradas iguais a zero.

Da definição acima, $\|\mathbf{k}(m, n+1)\|_1 = m$, isto é, a norma um de $\mathbf{k}(m, n+1)$ é m , $1 \leq m \leq n+1$.

Definição 2

Sejam α e β dois índices do vetor $\mathbf{k}(m, n+1)$ tal que $1 \leq \alpha, \beta \leq n+1$, α é o menor índice satisfazendo $k_{\alpha} = 1$ (portanto $k_{\alpha-1} = 0$ se $k_1 = 0$ ou $\alpha = 1$ caso contrário); e β é o maior índice satisfazendo $k_{\beta} = 1$.

Note que, da definição 2, $\beta = \alpha + m - 1$. O Teorema 1 abaixo descreve um algoritmo recursivo e eficiente para o cálculo de $P_{\Theta}(v)$.

Teorema 1

$$P_{\Theta}(v) = \sum_{n=0}^{\infty} \left(\frac{(\lambda w)^n}{n!} e^{-\lambda w} \right) \Upsilon(v, n) \quad (11)$$

sendo $\Upsilon(v, n) = \Theta(v, \mathbf{k}(n+1, n+1))$. $\Theta(v, \mathbf{k}(n+1, n+1))$ é calculado pela seguinte recursão, para qualquer valor de $n \geq 0$, $1 \leq m \leq n+1$:

$$\begin{aligned} \Theta(v, \mathbf{k}(m, n+1)) = & \quad (12) \\ & \left(\frac{(n+1-\alpha)-v}{m-1} \right) \Theta(v, \mathbf{k}(m, n+1) - \mathbf{1}_{\beta}) + \\ & \left(\frac{v-(n+1-\beta)}{m-1} \right) \Theta(v, \mathbf{k}(m, n+1) - \mathbf{1}_{\alpha}). \end{aligned}$$

Sendo que as condições iniciais para a recursão são

$$\Theta(v, \mathbf{k}(m, n+1)) = \begin{cases} 0 & \text{se } n+1-\alpha \leq v \\ 1 & \text{se } n+1-\beta > v. \end{cases} \quad (13)$$

Prova:

Na Seção 3 foi mostrado que, condicionado a n pedidos no intervalo $(0, w)$, a soma dos patches é igual a soma de n variáveis aleatórias uniformes iid naquele intervalo. Essa soma tem distribuição complementar igual a $\Upsilon(v, n)$, e é igual a $G'(n)$ atribuindo valores apropriados para os d_j conforme indicado acima. A equação (11) é obtida simplesmente descondicionando no número de pedidos em $(0, w)$. Como o número de pedidos dos clientes é dado por uma distribuição Poisson com taxa λ , o descondicionamento é trivial.

Em [20], Diniz, de Souza e Silva e Gaiil obtiveram uma recursão eficiente para o cálculo da combinação linear de n estatísticas de ordem de variáveis aleatórias uniformes iid. A recursão dada pela equação (11) é obtida particularizando os valores da recursão de [20] para $a_1 = \dots = a_{n+1} = 1$, e fazendo as devidas simplificações. \square

Observações:

- A soma infinita da equação (11) pode ser facilmente truncada a um valor N de forma a que o resultado esteja dentro de uma tolerância dada. Seja $\epsilon(N)$ o valor do erro cometido quando a série é truncada a N .

$$\epsilon(N) \leq 1 - \sum_{n=0}^N \frac{(\lambda w)^n}{n!} e^{-\lambda w}.$$

Por exemplo, se $\epsilon(N) < 10^{-3}$ e $\lambda w = 1$ então $N = 5$; caso $\lambda w = 10$, então $N = 21$. Para valores grandes de λw a distribuição Poisson pode ser aproximada pela distribuição normal com parâmetro $(\lambda w, \lambda w)$. Então:

$$N \approx \lambda w + z_\epsilon \sqrt{\lambda w}$$

sendo que z_ϵ é tal que $P[\Omega > z_\epsilon] \leq \epsilon$ e Ω é normal $(0, 1)$.

- Os valores de $\Upsilon(v, n)$ são independentes de λ e w . Portanto, os mesmos valores $\Upsilon(v, n)$ podem ser usados para diversos tamanhos de janela de *patch* e taxas de chegada de pedidos dos cliente, reduzindo os custos computacionais quando o objetivo for calcular as métricas definidas acima, variando λ e w .

O custo computacional para calcular $P_\Theta(v)$ depende do valor N de corte da série infinita de (11) (o que depende da tolerância usada), e do cálculo dos termos $\Theta(v, k(m, n+1))$. Para uma determinada tolerância $\epsilon(N)$ os valores $\Theta(v, k(n+1, n+1))$ para $0 \leq n \leq N$ precisam ser calculados.

Considere inicialmente o custo de armazenamento. Para implementar a recursão precisa-se de apenas um vetor de dimensão $N+1$ e de duas variáveis inteiras (os índices) $(\alpha$ e $\beta)$. A cada iteração da recursão os valores dos índices são alterados.

Seja agora o número de operações. Para calcular um determinado valor $\Theta(v, k(m, n))$, são necessárias apenas duas multiplicações. É fácil ver que a recursão pode ser modelada como uma árvore binária. Seja um dado nó $\Theta(v, k(m, n+1))$. O ramo da direita é $\Theta(v, k(m, n+1) - 1_\beta)$, e o ramo da esquerda é $\Theta(v, k(m, n+1) - 1_\alpha)$. É importante observar que um determinado nó pode possuir mais de um nó-pai. Por exemplo, o vetor $k = \langle 0, 1, 0 \rangle$ é filho de $k = \langle 1, 1, 0 \rangle$ e de $k = \langle 0, 1, 1 \rangle$. Como os nós do nível m da árvore são tais que $\|k(m, n+1)\|_1 = m$ os vetores correspondentes a cada nó tem o mesmo número de entradas iguais a zero, variando apenas os índices α e β correspondentes. Então o número de nós no nível m da árvore é igual a $n - m + 1$. Em consequência, para se obter todos os nós de um nível m são necessárias no máximo $2(n - m + 1)$ multiplicações. Naturalmente, quanto maior a altura da árvore, maior o custo da recursão. No pior caso, temos que a altura da árvore é igual a N e portanto, o número total de multiplicações para o cálculo de uma determinada árvore é $O(N^2)$.

Note que o problema exige o cálculo de $\Theta(v, k(n+1, n+1))$ para $n = 0, \dots, N$ (N é o ponto de corte da equação (1)). A princípio seria preciso calcular $N+1$ árvores. Entretanto, é possível mostrar que apenas a árvore com N níveis precisa ser obtida.

Seja o vetor $k'(m, n+1)$ cujo índice β' é igual a $n+1$, isto é as entradas k'_1, \dots, k'_{n+1-m} são iguais a zero. Seja o vetor $k^*(m, l)$, $l < n+1$ com índice $\beta^* = l$. (β' e β^* são definidos para k' e k^* da mesma forma que β foi para k .) Portanto, $k_1^* = \dots = k_{l-m}^* = 0$ e $k_{l-m+1}^* = \dots = k_l^* = 1$.

Não é difícil mostrar que $\Theta(v, k^*(m, l)) = \Theta(v, k'(m, n+1))$. Logo, é suficiente para se obter $P_\Theta(v)$ calcular apenas valores correspondentes a árvore que tem como raiz $\Theta(v, k(N+1, N+1))$. Em consequência, o custo total do algoritmo, em relação ao número de multiplicações é $O(N^2)$. N é em geral pequeno, como ilustrado acima. Além do mais relembramos que os $\Upsilon(v, n)$ independem dos valores de λ e w .

5. RESULTADOS

Para que os resultados obtidos neste trabalho sejam independentes da taxa de transmissão de um filme, definimos a medida *capacidade relativa* como $C_R = C_T/C$, sendo que C_T foi definido na Seção 3.3 e C é igual a capacidade necessária para a transmissão de um filme considerada constante (por exemplo igual a 1.5Mbps).

Nos modelos apresentados na Seção 2, foram obtidos valores para a janela ótima e a respectiva capacidade média C_{wo} para a técnica de *patching*. Realizamos alguns experimentos com os seguintes objetivos: (i) estudar a sensibilidade da janela de *patching*, ou seja, verificar como se comporta a distribuição da capacidade relativa quando variamos o valor desta janela; (ii) obter a probabilidade da capacidade relativa ser maior do que a média C_{wo} ; (iii) obter os valores de capacidade relativa que satisfazem ao requisito: $P[C_R > c_r]$ é igual a um dado limite γ .

Consideramos um filme de duração $T = 100$ minutos e λ variando de 0.1 a 5 clientes por minuto. Como $\mathcal{N} = \lambda T$, temos $\mathcal{N} = [10, 500]$ clientes que podem chegar em média durante a duração de um filme. Estes valores foram escolhidos pois estudos da literatura (por exemplo, [13]) mostram que para valores de $\mathcal{N} > 100$ existem outras técnicas cujos requisitos de capacidade média são inferiores aos requisitos da técnica de *patching*.

5.1 ESTUDO DA VARIAÇÃO DA JANELA

Consideramos $\mathcal{N} = 10, 25, 50, 100, 500$. Os valores de janela ótima e respectiva capacidade calculados segundo as equações (2) e (3), estão listados na Tabela 1. A unidade de capacidade média é o número de canais onde cada canal possui capacidade para transmitir um fluxo (por exemplo, 1.5Mbps para filmes codificados em MPEG-2).

\mathcal{N}	Janela ótima (min)	Capacidade média (N^2 de canais)
10	35.82	3.58
25	24.56	6.14
50	18.09	9.05
100	13.17	13.17
500	6.13	30.64

Table 1. Requisitos de capacidade para a janela ótima

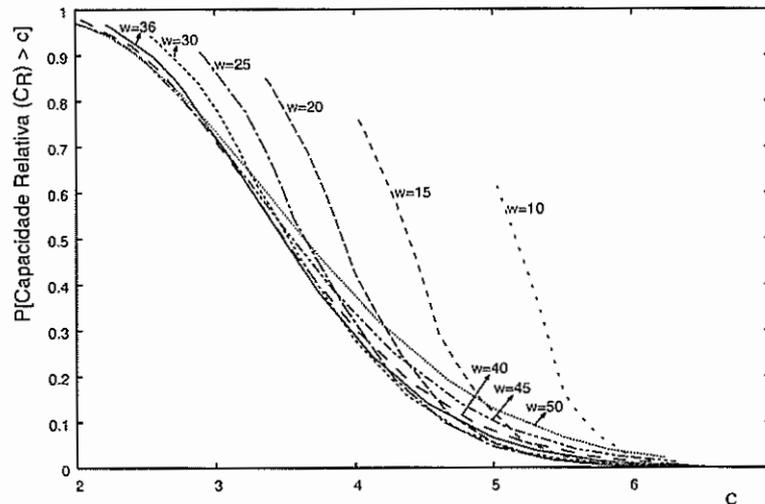


Figure 4. Distribuição da capacidade relativa para $\mathcal{N} = 10$.

A Figura 4 apresenta a distribuição da capacidade relativa para $\mathcal{N} = 10$ considerando diversos valores de janela. Observamos que para valores de janela entre 25 e 36 minutos, as probabilidades não diferem de mais de uma ordem de grandeza. Podemos notar também que a probabilidade da capacidade relativa ser maior que a capacidade calculada para a janela ótima (ou seja 3.58) é aproximadamente 0.5 para $w = [25, 36]$. Isto indica que o valor de C_{wo} não satisfaz os requisitos de capacidade de um determinado ciclo de requisição em 50% dos ciclos observados, isto é, 50% dos ciclos de requisição geram mais tráfego que a média.

No gráfico da Figura 5 mostramos que para $w = 20$ pode-se obter uma probabilidade uma ordem de grandeza inferior à probabilidade obtida para a janela ótima para certos valores de capacidade relativa C_R . Suponha que o objetivo seja dimensionar um servidor para que a probabilidade de C_R ultrapassar um certo valor seja inferior a um dado limite, por exemplo 10^{-4} . Com uma janela de $w = 20$ e $C_R = 6.5$ esta condição é satisfeita. Já para $w = 36$ (janela "ótima"), a capacidade relativa necessária é igual a 7.7. Este resultado mostra que nem sempre o valor da janela ótima é o que resulta em maior economia de capacidade relativa necessária a satisfazer a QoS desejada.

Na Figura 6 temos a distribuição da capacidade para $\mathcal{N} = 50$. Podemos observar que para valores de janela variando entre 15 e 20 minutos temos que as probabilidades para um dado valor de capacidade relativa são muito próximas considerando valores de probabilidade até 10^{-6} . Neste caso, podemos observar que a escolha da janela ótima resultaria em maior economia de capacidade relativa.

Verificamos que a probabilidade da capacidade relativa ser maior do que C_{wo} (igual a 9.05) é de aproximadamente 0.7 para $w = [15, 20]$, indicando que existe uma alta probabilidade dos requisitos de capacidade relativa serem superiores aos requisitos estimados pelo parâmetro C_{wo} .

Para o caso de $\mathcal{N} = 100$, podemos observar através da Figura 7 que somente valores de janela de 13 e 15 minutos apresentam distribuição de capacidade relativa semelhante.

Através dos resultados obtidos com os experimentos descritos acima, podemos concluir que a medida que \mathcal{N} aumenta uma pequena variação na escolha do valor da janela pode re-

sultar em uma grande diferença na função distribuição da capacidade relativa. Ou seja a sensibilidade do valor da janela de *patching* aumenta com o valor de \mathcal{N} . Por exemplo para $\mathcal{N} \leq 25$, a janela pode variar em um intervalo de aproximadamente 10 minutos, já para $\mathcal{N} = 100$ este intervalo é de aproximadamente 3 minutos sem que a função distribuição tenha alterações significativas.

5.2 ESTUDO DA CAPACIDADE RELATIVA NECESSÁRIA PARA ATENDER REQUISITOS DE QoS

Suponha que o objetivo seja dimensionar um servidor de forma a que o seguinte requisito seja atendido: a probabilidade da capacidade relativa ser maior que um dado limite é igual um certo valor, ou seja $P[C_R > c_r] = \gamma$. Logo, é necessário obter os valores de c_r que satisfazem esta condição. Obtivemos estes valores de c_r para dois valores de γ . Foi ainda obtida $P[C_R > C_{wo}]$, isto é a probabilidade de que a capacidade média C_{wo} , calculada a partir de (3), seja ultrapassada. Consideramos $\mathcal{N} = [10, 500]$ e usamos os valores de janela ótima.

A Figura 8 ilustra os resultados. Podemos observar que as probabilidades obtidas para a capacidade C_{wo} são bastante altas, indicando que C_{wo} não é um parâmetro adequado para dimensionar o servidor se o objetivo é a obtenção de uma probabilidade menor do que 10^{-1} . É possível notar que para obter-se valores de probabilidade da ordem de 10^{-2} a capacidade relativa deve ser 40% maior que a capacidade C_{wo} . E para que a probabilidade seja da ordem de 10^{-3} o acréscimo varia de 100% a 50% com o aumento do valor de \mathcal{N} .

Uma outra medida que pode indicar a qualidade de serviço que o servidor pode oferecer é $\sigma(v)$ definida na equação (6). Podemos dizer que a cada $1/\sigma(v)$ um *patch* de tamanho maior que v ocorre.

A Figura 9 ilustra o comportamento da variável $\sigma(v)$. Consideramos $\mathcal{N} = 50$ e $w = 18$ minutos. Para estes valores, temos que $C_{wo} = 9.05$ que é o valor inicial do eixo das abscissas do gráfico. No eixo das abscissas representamos as capacidades relativas calculadas para tamanhos de *patches* maiores do que o tamanho médio do *patch* obtido

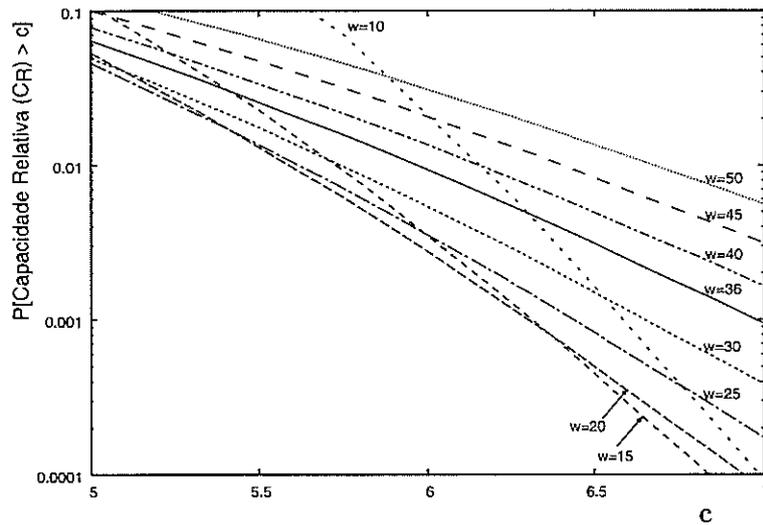


Figure 5. Distribuição da capacidade relativa para $\mathcal{N} = 10$.

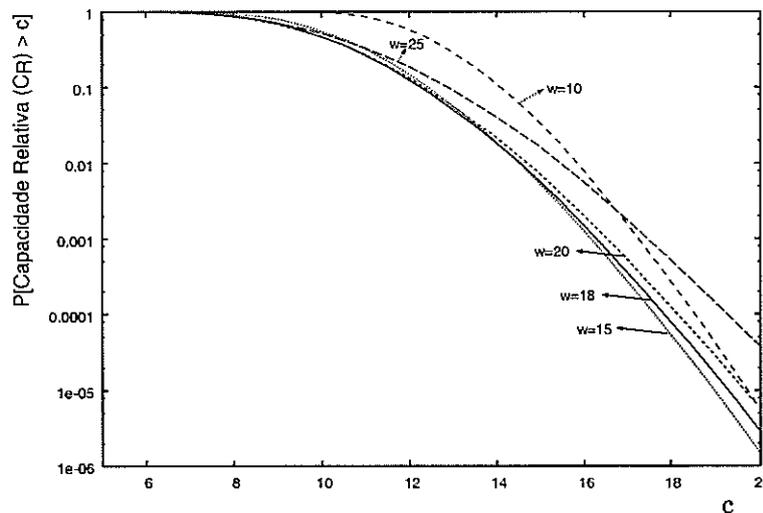


Figure 6. Distribuição da capacidade relativa para $\mathcal{N} = 50$.

para $w = 18$ (janela ótima). Desta forma poderemos observar qual a frequência de ocorrência de *patches* com duração superior ao *patch* médio estimado com o valor da janela ótima. No eixo das ordenadas está representada a variável $1/\sigma(v)$ normalizada pela duração de um filme (100 minutos).

O gráfico da Figura 9 indica que, em um intervalo de 100 minutos (igual a duração de um filme), ocorrerão ciclos de requisição cujo valor de C_R está no intervalo $[9, 11]$. Ou seja, ocorrerão ciclos cuja demanda por capacidade é 20% maior do que C_{wo} . Observando o gráfico para outros valores de capacidade, podemos notar que, em média, a cada 10 transmissões completas de um filme, ocorrerão ciclos cuja capacidade relativa é 55% maior que C_{wo} . É evidente então o impacto adverso na QoS se o valor médio C_{wo} for usado no dimensionamento.

Através dos resultados obtidos nesta seção podemos observar que nem sempre a capacidade C_{wo} calculada pode fornecer a QoS desejada. Para valores de $\mathcal{N} = 10, 50$, a probabilidade da capacidade relativa ser maior do que C_{wo}

varia entre 0.45 a 0.74, começando a decrescer para valores de $\mathcal{N} > 50$ e chegando a 0.22 para $\mathcal{N} = 500$. Estes resultados indicam que caso o servidor use o valor de C_{wo} para dimensionamento, existe uma alta probabilidade da demanda não ser atendida.

6. CONCLUSÃO

Obtivemos expressões analíticas para o cálculo da distribuição de requisitos de capacidade para a técnica de *patching*. O algoritmo elaborado é simples de implementar e eficiente. Os resultados mostraram que o uso da capacidade média calculada em trabalhos publicados na literatura pode não ser uma medida adequada ao dimensionamento do canal de distribuição de um servidor de vídeo, de forma a prover uma qualidade adequada. Em outras palavras, de forma a que a capacidade necessária à transmissão tenha uma baixa probabilidade de ultrapassar a capacidade alocada. Mostramos

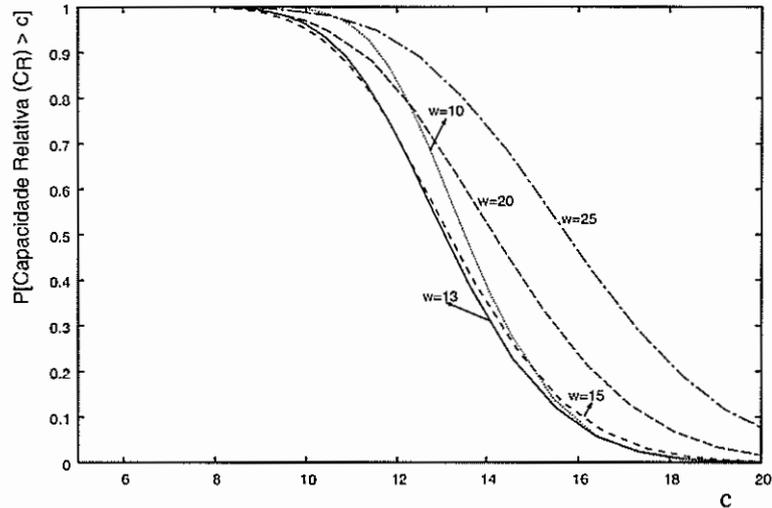


Figure 7. Distribuição da capacidade relativa para $N=100$.

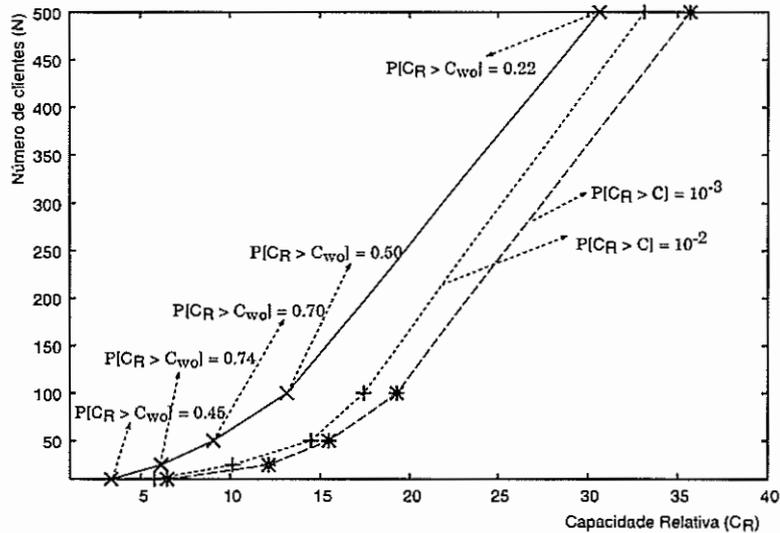


Figure 8. Variação da capacidade relativa com o número de clientes.

ainda que a frequência de ocorrência de intervalos onde a capacidade ultrapassa a média "ótima" não é desprezível.

Embora não tratado neste trabalho a metodologia de análise pode ser generalizada para estudar o dimensionamento de um sistema onde as chegadas não obedecem a um processo de Poisson. Pretende-se ainda generalizar o método de forma a estudar outras técnicas de compartilhamento de canal tais como os agrupamentos hierárquicos.

7. AGRADECIMENTOS

Este trabalho é parcialmente financiado pelo CNPq e FAPERJ.

REFERENCES

[1] E. de Souza e Silva, R. Leão, B. Ribeiro-Neto, and S. Campos, "Performance Issues of Multimedia Applications," in *Performance Evaluation of Complex Systems: Techniques and Tools* (S. T. Maria Carla Calzarossa, ed.), vol. LNCS 2459, pp. 374–404, Springer, September 2002.

[2] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic Batching Policies for an On-Demand Video Server," *Multimedia Systems*, no. 4, pp. 112–121, 1996.

[3] S. W. Carter and D. D. E. Long, "Improving Video-on-demand Server Efficiency through Stream Tapping," in *Sixth International Conference on Computer Communications and Networks*, pp. 200–207, 1997.

[4] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A Multicast Technique for True Video-on-Demand Services," in *Proceedings of ACM Multimedia*, pp. 191–200, 1998.

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

[6] L. Gao and D. Towsley, "Supplying Instantaneous Video-on-demand Services using Controlled Multicast," in *IEEE International Conference on Multimedia Computing and Systems*, pp. 117–121, 1999.

[7] D. Eager, M. Vernon, and J. Zahorjan, "Optimal and Efficient Merging Schedules for Video-on-Demand Servers," in *Proc. ACM Multimedia '99*, November 1999.

[8] A. Bar-Noy, G. Goshi, R. E. Ladner, and K. Tam, "Comparison of Stream Merging Algorithms for Media-on-Demand," in

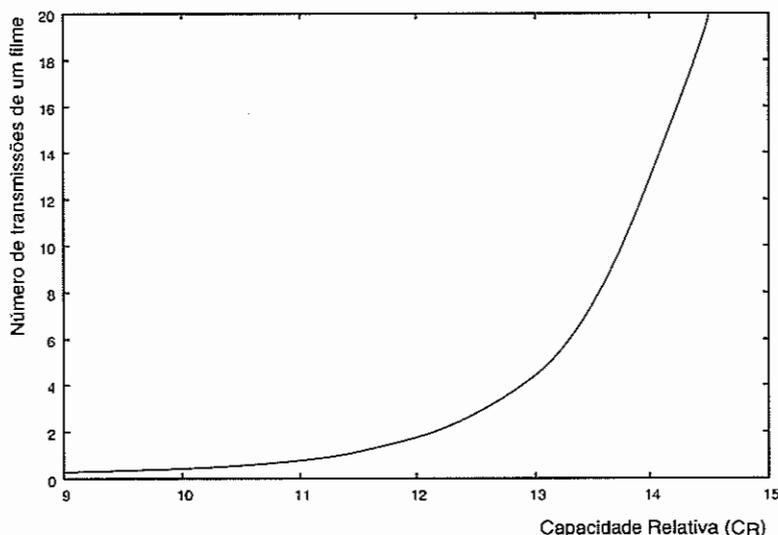


Figure 9. Capacidade relativa \times Número de transmissões de um filme.

Proc. MMCN'02, January 2002.

[9] C. C. Aggarwal, J. L. Wolf, and P. S. Wu, "A Permutation-based Pyramid Broadcasting Scheme for Video-on-Demand Systems," in *Proc. of the IEEE Conf. on Multimedia Systems*, 1996.

[10] K. Hua and S. Sheu, "Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-demand Systems," in *Proc. of ACM Sigcomm'97*, pp. 89–100, ACM Press, 1997.

[11] A. Hu, "Video-on-Demand Broadcasting Protocols: A Comprehensive Study," in *Proc. IEEE Infocom*, pp. 508–517, 2001.

[12] R. Zafalão, N. Fonseca, and C. de Souza, "O Protocolo Polyharmonic Broadcasting sujeito a Restrições de Banda Passante," in *Anais do Simpósio Brasileiro de Redes de Computadores*, pp. 397–410, 2003.

[13] D. Eager, M. Vernon, and J. Zahorjan, "Minimizing Bandwidth Requirements for On-Demand Data Delivery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, pp. 742–757, September 2001.

[14] E. de Souza e Silva, B. R. Neto, R. Leão, S. Campos, D. Towsley, L. Golubchik, and R. Muntz, "The COMIT Project: Application Level Adaptation and Control for Retrieval and Delivery of Continuous Media over the Internet Networks," in *Proceedings of the Projects Evaluation Workshop: ProTeM-CC-2001*, pp. 445–468, ProTeM-CC'2001/CNPq/Brazil, 2001.

[15] LAND, *Laboratory for modeling, analysis and development of networks and computing systems*. URL <http://www.land.ufrj.br>.

[16] S. Karlin and H. M. Taylor, *A Second Course in Stochastic Processes*. Academic Press, 1981.

[17] S. Ross, *Stochastic Processes*. John Wiley & Sons, 1983.

[18] J. Santos, R. Muntz, and B. Ribeiro-Neto, "Comparing Random Data Allocation and Data Striping in Multimedia Storage Servers," in *Proceedings of ACM SIGMETRICS*, pp. 44–55, 2000.

[19] A. de Q. Cardozo, E. de Souza e Silva, and R. Leão, "Um Protótipo de Servidor Multimídia com Mecanismos de QoS," in *Workshop da RNP*, 2003.

[20] M. Diniz, E. de Souza e Silva, and H. Gail, "Calculating the Distribution of a Linear Combination of Uniform Order Statistics," *INFORMS Journal on Computing*, vol. 14, no. 2, pp. 124–131, 2002.

Edmundo de Souza e Silva é Engenheiro e Mestre em Engenharia Elétrica pela PUC/RJ, e recebeu o grau de Ph.D. em Ciência da Computação da University of California, Los Angeles (UCLA) em 1984. Foi Cientista Visitante e Professor Visitante do centro de pesquisas T.J. Watson da IBM, Cientista Visitante do Laboratório de pesquisa da IBM em Tokyo, Professor Visitante dos Departamentos de Ciência da Computação da UCLA e da University of Southern California (USC). Foi também Pesquisador Visitante da Politécnica de Torino, da Chinese University of Hong Kong e IRISA/INRIA-Rennes. Tem participado de vários comitês de programa de conferências nacionais e internacionais. Foi PC co-Chair da IFIP Third Int. Conf. on Data Commun. Syst. and their Performance (1987), PC co-Chair da IEEE/GLOBECOM'99, PC vice-co-Chair do ITC'2001, e PC co-Chair da ACM/Sigmetrics'2002. Atualmente é Professor Titular da Universidade Federal do Rio de Janeiro, Departamento de Ciência da Computação do IM e do Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, e membro do Conselho Universitário. É também vice-chair do IFIP WG 7.3, e um membro do Board of Directors da ACM/SIGMETRICS. Suas áreas de interesse incluem a modelagem e análise de sistemas de computação e redes multimídia de comunicação.

Rosa M. M. Leão graduou-se em Informática pela Universidade Federal do Rio de Janeiro em 1983. Obteve seu título de mestre em Ciência da Computação em 1990 pela PUC-RIO e seu título de doutor em 1994 pela Université Paul Sabatier no LAAS/CNRS na área de modelagem de redes de alta velocidade. Foi pesquisadora visitante do IRISA/INRIA de Rennes e tem participado do comitê de programa de várias conferências nacionais e internacionais. Atualmente é professora do Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ. Suas áreas de interesse incluem redes e aplicações multimídia, engenharia de tráfego, e modelagem e análise de sistemas de comunicação.

Morganna C. Diniz graduou-se em Informática pela Universidade Federal do Ceará em 1986. Obteve os títulos de mestre e doutor em Engenharia de Sistemas pela COPPE-Sistemas/UFRJ em 1990 e 2000, respectivamente. Atualmente é professora dos Programas de Graduação e Mestrado das Faculdades de Administração e Economia IBMEC/RJ. Suas áreas de interesse incluem modelagem, métodos de solução e análise de modelos relacionados a redes multimídia.