Aplicação de Modelos Ocultos de Markov na Distribuição de Vídeo sob Demanda

Felipe A. Pinto e Carlos M. Pedroso

***Resumo*— As aplicações de transferência de vídeo já representam a maior parte do tráfego da Internet. Os serviços de Vídeo sob Demanda (VoD, *video on demand*) são os principais responsáveis por este tráfego. Espera-se nos próximos anos um incremento na qualidade dos vídeos transmitidos, o que irá pressionar ainda mais as infraestruturas de rede disponíveis. Neste cenário, o uso de métodos para melhorar a eficiência na transmissão de sistemas VoD tem reflexos profundos sobre o desempenho da rede e sobre o dimensionamento de capacidade dos sistemas. Os métodos mais eficientes disponíveis para distribuição de VoD utilizam estratégias que combinam a transmissão *multicast* com a capacidade de armazenamento disponível no equipamento do cliente. Neste artigo apresentamos um novo método para transmissão de conteúdo de vídeo que utiliza capacidades *multicast* e de armazenamento do equipamento do cliente, mas com a popularidade de acesso à vídeos caracterizado com um Modelo Oculto de Markov (HMM, *Hidden Markov Model*). A eficiência do método proposto é demonstrada a partir de simulações computacionais utilizando dados reais. Os resultados indicam ganhos significativos em comparação aos melhores métodos disponíveis atualmente.**

***Palavras-Chave*— Video streaming, Multimedia, Video on Demand, Hidden Markov Model.**

***Abstract*— Internet traffic is already dominated by video streaming applications. The Video on Demand (VoD) services are responsable for the majority of this applications. An increase in the quality of transmitted videos is expected in the coming years, which will further pressure the available network infrastructures. In this scenario, the use of methods to improve the efficiency of the transmission of VoD systems has repercussions on the performance of the network and on the capacity planning of the systems. The most efficient methods available for VoD distribution use strategies that combine multicast transmission with the storage capacity available on the customer's equipment. In this article we present a new method for transmitting video content that uses multicast and storage capabilities of client equipment, but with the popularity of videos characterized by a Hidden Markov Model (HMM). The efficiency of the proposed method is demonstrated from computational simulations using real data. The results indicate that proposed method outperforms the state-of-art method.**

***Keywords*— video streaming, video on demand, hidden Markov model.**

I. INTRODUÇÃO

Com o passar dos anos as tecnologias de acesso das redes banda larga evoluíram e hoje a disponibilidade de taxas mais altas é comum. Dentre as aplicações disponíveis para o usuário a transmissão de VoD se torna cada vez mais popular entre os usuários, fazendo com que o tráfego de vídeo na rede tenha um grande impacto no consumo de recursos das redes. Estima-se que em 2021 o tráfego de vídeo corresponda a 82% do tráfego total de Internet [1]. O serviço VoD possibilita que os usuários tenham maior interatividade com o sistema com o acesso a conteúdos de vídeo a qualquer momento, ao invés de ter acesso somente a conteúdos *broadcast* com horário definido. Alguns provedores aplicam algoritmos para recomendar conteúdos, o que gera uma alteração no interesse do usuário [2]. Os problemas relativos ao transporte de tráfego VoD em uma rede IP a nível de distribuição pode ser segmentado na análise de consumo de banda no núcleo da rede e escalabilidade do sistema. Além disso, atender mais usuários com uma única transmissão do servidor implica em menor quantidade de recursos de hardware exigida no servidor, com consequente redução de custos. Como os vídeos mais populares são acessados com uma frequência maior, existem diversas abordagens que utilizam deste fato para melhorar a eficiência durante a distribuição de vídeos [3].

A popularidade dos vídeos é comumente caracterizada com a distribuição de Zipf [4], onde a popularidade de um vídeo i é dada pela expressão $p_i = (1/i^\alpha)/(\sum_{j=1}^{N} 1/j^\alpha)$, com N indicando a quantidade de vídeos e α um fator de inclinação. As requisições de acesso que chegam em um servidor de vídeo podem ser caracterizadas através de um processo de Poisson com taxa λ [5].

Neste artigo, apresenta-se uma nova abordagem para previsão do tipo de conteúdo que será requisitado pelo usuário, de forma a explorar mais eficientemente a alocação de segmentos dos vídeos na memória cache do dispositivo do usuário de tal forma que em conjunto com uma estratégia de transmissão *multicast*, que permite transmitir um fluxo de dados para múltiplos usuários, possa reduzir o consumo de banda no núcleo da rede durante a distribuição de tráfego VoD. Também estamos interessados em estudar o efeito desta abordagem frente a diferentes cenários de taxa de requisição de usuário, popularidade da biblioteca de vídeos e ainda analisar o consumo de banda frente a diferentes tipos de usuários. A estratégia proposta modela o comportamento do usuário utilizando uma HMM. O treinamento da HMM pode ser realizado no servidor utilizando dados disponíveis sobre as classes de conteúdo acessadas pelos usuários (ex. drama, ficção, terror, etc. ou mesmo séries específicas). Estas preferências do usuário ainda não foram exploradas por outros métodos de distribuição de conteúdo VoD.

Além desta seção introdutória, este artigo está organizado da seguinte forma: a Seção II faz breve apresentação dos métodos de distribuição de vídeos clássicos e o estado da arte; a descrição do modelo proposto é feita na Seção III; a Seção

Felipe A. Pinto, Carlos M. Pedroso, Departamento de Engenharia Elétrica, Universidade Federal do Paraná (UFPR), Curitiba-PR, Brasil, E-mail: felipe.fap.ufpr@gmail.com, pedroso@eletrica.ufpr.br

IV apresenta a metodologia adotada, os cenários simulados e os principais resultados obtidos. Finalmente na Seção V são apresentadas as conclusões e trabalhos futuros.

II. Trabalhos relacionados

O uso de transmissão *unicast* (ponto-a-ponto) para distribuição de tráfego VoD, onde um fluxo de vídeo dedicado é enviado do servidor para cada cliente, não é eficiente. Existem diversos métodos disponíveis que visam aumentar a eficiência. Alguns métodos são baseados em *broadcast*, ou seja, o vídeo é enviado para todos os usuários a cada período de tempo T. Outros métodos utilizam-se do conhecimento prévio da popularidade dos vídeos. É possível encontrar métodos que exploram a capacidade de armazenamento na memória do dispositivo do cliente e capacidades de transmissão em *multicast*. Nesta seção serão discutidos os métodos clássicos e estado da arte dos algoritmos para distribuição de VoD, evidenciando as principais características de cada método.

A. Métodos Clássicos

1) Batching: Algoritmos baseados em *batching* são algoritmos que esperam um tempo T (janela *batching*) para iniciar a transmissão do fluxo de vídeo após a primeira requisição de um dado vídeo v_i ocorrida no tempo t_r. Durante o tempo $[t_r, t_r + T]$ o algoritmo aguarda que o servidor de vídeo receba novas requisições para v_i antes de iniciar a transmissão no tempo $t_r + T$, desta forma utilizando-se de transmissão *multicast* o servidor poderá entregar v_i para diversos usuários com apenas um *stream* [6]. Uma requisição para o vídeo v_i que ocorra após o termino da janela *batching* será atendida por um segundo *stream* que será agendado pelo servidor respeitando o tamanho da janela *batching*. Este algoritmo pode ser viável para vídeos populares onde muitas requisições serão atendidas com apenas um *stream*, porém para vídeos não populares este método não é eficiente, pois tende a torna-se uma transmissão *unicast*. Nesta abordagem o usuário poderá ter que aguardar até um tempo T antes de começar assistir ao vídeo, o que pode motivar o usuário a deixar a plataforma do serviço antecipadamente.

2) Patching: Neste algoritmo é explorado o *buffer* do equipamento do usuário, para que o mesmo possa receber o fluxo *multicast* principal mais um fluxo chamado *patching*. Nesta abordagem o usuário requisita um vídeo v_i no tempo t_r, o servidor de vídeo iniciará imediatamente a transmissão de v_i através de um fluxo *multicast* principal. Se no instante t_r houver um fluxo *multicast* principal ativo no servidor o mesmo iniciará a transmissão da porção do vídeo faltante através de um fluxo *patching* [7]. Esta abordagem tem um problema quando as requisições chegam no final do fluxo *multicast* principal, pois o *patching* será muito longo. Para sanar este problema o servidor deverá aceitar criar *patchings* enquanto as requisições para o vídeo v_i chegarem antes de um *limite de tempo ótimo* a partir do início do *stream multicast* principal [8]. Caso contrário um novo fluxo *multicast* principal deverá servir a esta requisição. Com isso, este método apresenta zero *delay* para o usuário, ou seja, em qualquer momento que houver uma requisição o usuário poderá inciar o vídeo.

B. Estado da Arte

1) Prepopulation Assisted Batching with Multicast Patching (PAB-MP): Este é um algoritmo pertencente à classe dos métodos que utilizam o dispositivo do usuário para armazenar segmentos iniciais dos vídeos (IVS, *initial video segment*) [9]. Ao fazer uma requisição para o vídeo v_i, o usuário começa a assistir o vídeo utilizando informações armazenadas no IVS. Durante este período o servidor pode aguardar antes de iniciar a transmissão da porção faltante do vídeo e realizar um *batching*, fazendo com que mais usuários possam usufruir do mesmo fluxo *multicast*. O tamanho do IVS irá definir a janela *batching*. Para requisições que ocorram após esse limite o servidor enviará um fluxo *patching multicast*, de forma a atender mais requisições com o mesmo fluxo *patching*. Requisições que cheguem após a janela *patching* serão atendidas com um novo fluxo *multicast*. Os IVS são transmitidos para os dispositvos do usuário nos momentos de baixo uso da rede e servidor. Jayasundara et al. [9] sugerem o algoritmo D-PLO (*Dynamic programming-based Prepopulation Lengths Optimization*) para determinar o tamanho de cada IVS que será armazenado no dispositivo do usuário. Adicionalmente os autores sugerem que os dispositivos de usuários compartilhem entre si os IVSs disponíveis localmente, em uma configuração *peer-to-peer*, para aumentar a probabilidade de encontrar o vídeo solicitado. Os autores mostram que o PAB-MB é mais eficiente que as estratégias *patching* e *batching* até então existentes.

2) Client Cache Enabled Multicast Patching (CCE-MP): CCE-MP aborda o problema de transmissão de vídeo através de segmentos de vídeo utilizando *multicast*, ou seja, para cada requisição de um dado vídeo v_i no tempo t_r, o dispositivo do usuário imediatamente começará a armazenar todos os segmentos ativos de *multicast* de v_i que estejam ativos e o servidor transmitirá somente os segmentos faltantes. O início da transmissão de um segmento de v_i será realizada o mais tarde possível em relação a t_r, para que o maior número de usuários possam utilizar este segmento de v_i [10]. Esta abordagem também pertence a classe de métodos que exploram a capacidade de armazenamento no cliente. Como os dispositivos de usuários possuem capacidade limitada, este método sugere o uso do algoritmo *Water-Filling* para alocação ótima dos IVSs, alcançando assim uma melhora na eficiência do sistema. Os autores mostram que o CCE-MP possui um desempenho superior ao PAB-MP.

III. Modelo Proposto

O conteúdo acessado pelo usuário de VoD apresenta um comportamento previsível [11] [12], fato este já explorado pelos sistemas de recomendação em uso pelos provedores [2]. Iremos explorar esta previsibilidade para permitir que o servidor possa fazer a alocação dos IVSs, melhorando o desempenho do CCE-MP. Neste artigo propomos um modelo baseado em HMM para prever o gênero dos próximos conteúdos que serão assistidos pelos usuários, de modo que possibilite ao servidor fazer a alocação do IVSs com maior eficiência, reduzindo o uso de banda no núcleo da rede e também de recursos nos servidores de VoD.

A. Modelos Ocultos de Markov

Em um modelo de *Markov* cada estado corresponde a um evento observável [13]. Nos modelos ocultos de *Markov* o estado atual do sistema não é diretamente observável. Estes modelos tem ampla aplicação em problemas de reconhecimento de padrões em diversas áreas do conhecimento como reconhecimento de voz, sequência de DNA, entre muitas outras. A Fig.1 exemplifica um HMM, onde E_k são os estados com $k = \{0, 1, 2\}$, p_{ij} são as probabilidades de transição do estado E_i para o E_j e W_k são os vetores de probabilidade de ocorrência, também chamado de probabilidade de emissão de um evento observável dado um estado E_k.

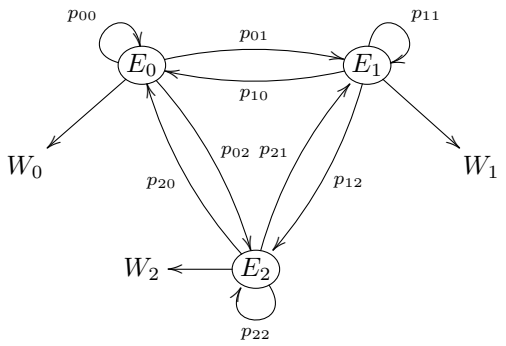

Fig. 1. Exemplo de Modelo Oculto de Markov

B. Previsão de Comportamento

Nesta subseção propomos uma maneira de utilizar HMM para prever o gênero do conteúdo que será acessado pelo usuário, onde cada estado do modelo representa os diferentes estados possíveis de um usuário. Assume-se que cada vídeo v_i presente na biblioteca do servidor de vídeo possui um único gênero associado de um conjunto finito $S = \{s_1, s_2, ..., s_M\}$ de M categorias e que o servidor de vídeo tem a capacidade de armazenar o histórico de vídeos acessados pelo usuário. Desta forma para cada estado E_k de um modelo com k estados existe um vetor de probabilidades de emissão $W_k = \{w_{k1}, w_{k2}, ..., w_{kM}\}$, onde $\sum_{t=1}^{M} w_{k,t} = 1$, que representa a probabilidade de ocorrência de cada uma das M categorias durante o estado E_k. A matriz quadrada P de dimensão k representa a probabilidade de transição entre os estados. Os parâmetros P e W_k do modelo podem ser treinados a partir dos dados do histórico de acessos contidos no servidor através do algoritmo de Baum-Welch e o estado provável do sistema pode ser determinado a partir da observação da sequência de conteúdos acessados pelo usuário utilizando o algoritmo *forward* [14]. Para uma melhor eficiência o servidor de vídeo pode fazer esse treinamento periodicamente conforme houver evolução no histórico de acessos do usuário.

C. Alocação de IVS

Com o HMM devidamente treinado, o algoritmo *forward* pode indicar o estado atual mais provável do usuário dada uma sequência de eventos observados. O vetor W_k correspondente pode ser utilizado para melhor definir a utilização da capacidade de armazenamento do equipamento do usuário. Seja C a capacidade útil que pode ser utilizada pelo servidor de vídeo para armazenar os IVSs, então para um dado W_k a capacidade C será dividida entre as categorias seguindo a probabilidade de emissão da mesma, ou seja, para uma categoria s_n o espaço reservado para os IVSs desta categoria pode ser dado por $c_{k,n} = C.w_{k,n}$. Para determinar a duração do IVS, denotada por li, dos vídeos de uma dada categoria s_n propõe-se utilizar o algoritmo *Water-Filling* [10], com as restrições:

$$\begin{cases} l_i = 0, & \text{se categoria de } v_i \neq s_n \\ l_i \leq L, & \text{se categoria de } v_i = s_n \\ \sum_{t=1}^{N} l_t \leq c_{k,n} & \end{cases} \quad (1)$$

onde L denota a duração total do vídeo e N é o número total de vídeos.

D. Modelo de Acesso de Usuário

Nesta subseção são apresentados os padrões de acesso de usuário em relação à categoria dos vídeos. Foram analisados dois padrões extremos e um caso com dados amostrais:

- **Polarizado:** Este tipo de usuário possui alta probabilidade de continuar assistindo a mesma categoria de conteúdo, como no fenômeno *binge watching* [15]. Este comportamento é descrito em um modelo HMM com uma matriz de transição de probabilidades P próxima a identidade, ou seja, uma vez que o usuário encontra-se no estado E_k este tende a permanecer no mesmo estado e o vetor de probabilidade de emissão W_k é altamente polarizado, isto é, o usuário tem preferência por apenas uma categoria.
- **Randômico:** Este usuário não possui preferência em relação ao conteúdo assistido, as categorias de acesso são randômicas. Este fato é representado por uma matriz P uniforme, onde a probabilidade de transição para qualquer outro estado ou ficar no mesmo estado é a mesma, por outro lado o vetor W_k possui probabilidade de emissão iguais entre as diferentes categorias.
- **Dados Amostrais:** Foram coletados históricos de acessos de 43 usuários[1]do Netflix [16] totalizando 36105 vídeos assistidos em um intervalo médio de 2.4 anos. Cada entrada foi classificada com uma das seguintes 16 categorias: Filme Drama, Filme Comédia, Filme Ação, Filme Aventura, Filme Animação, Filme Romance, Filme Ficção Científica, Filme Comédia Romântica, Filme Comédia Dramática, Filme Comédia de Ação, Filme Suspense, Filme Terror/Horror, Filme Fantasia, Seriado, Documentário e Outros. A partir destes dados foi utilizado o algoritmo Baum-Welch do software R [17] para treinar o HMM e obter a matriz P e os vetores W_k. Neste artigo foi aplicada uma HMM com 3 estados. A matrix P treinada é apresentada em (2), os valores treinados dos vetores W_k não são apresentados neste trabalho por terem dimensões muito extensas. Observa-se

[1]Dados disponíveis no endereço: http://www.eletrica.ufpr.br/pedroso/sbrt2018-VoD

TABELA I
PARÂMETROS DAS SIMULAÇÕES

Parâmetro	Descrição	Valor
N	Número de Vídeos	100
M	Número de Categorias	5
K	Número de Estados do HMM	3
α	Parâmetro da Lei de Zipf	0.8
r	Taxa do Vídeo	2Mbps
L	Duração do Vídeo	60 min
C	Capacidade de Armazenamento	120 min
λ	Taxa de requisições	10 requisições/min

que o algoritmo Baum-Welch detectou uma polarização razoável no comportamento do usuário, o que favorece a eficiência do método proposto.

$$\mathbf{P} = 10^{-3}{}_{\mathrm{x}} \begin{bmatrix} 979 & 1 & 20 \\ 253 & 273 & 474 \\ 2 & 911 & 87 \end{bmatrix} \quad (2)$$

IV. AVALIAÇÃO DE DESEMPENHO

Através de simulações computacionais foi feito uma análise comparativa do consumo de banda do método *Patching* [8] e CCE-MP [10] com o método proposto. Estudamos o desempenho do método proposto frente à diferentes taxas de requisições no servidor de vídeo e também nos interessamos em verificar o impacto do fator α da Lei de Zipf. A simulação considera que o vídeo requisitado será assistido pelo usuário em sua totalidade e ainda todos os vídeos possuem a mesma duração e taxa de transmissão. Esta suposição também é feita pelos autores dos métodos *Patching*, PAB-MP e CCE-MP, o que simplifica a análise sem perda de generalidade. A Tabela I mostra os demais parâmetros das simulações. Adicionalmente, foi considerado que a popularidade dos vídeos de uma mesma categoria segue a distribuição de Zipf com o parâmetro α e a seleção do estado inicial na HMM, que foi escolhida aleatoriamente.

Em um primeiro experimento foi considerado o consumo de banda em função da taxa de requisições λ recebidas pelo servidor. Neste teste o parâmetro α da distribuição de Zipf foi mantida constante em 0.8, que é um valor típico reportado na literatura [4] [9]. O resultado é apresentado na Fig.2. O gráfico apresenta o desempenho do *Paching*, CCE-MP e do método proposto. Para o método proposto foram considerados os casos de comportamento de Randômico e Polarizado de usuário. Observa-se que para baixas taxas de requisições não há uma diferença expressiva entre os métodos, contudo ao aumentar a taxa de requisições o modelo proposto se sobressai aos demais quando considerado Usuários Polarizados, pois com o método proposto existe um melhor aproveitamento da capacidade de armazenamento do dispositivo do usuário alocando-o para os IVS dos vídeos preferenciais do usuário. Quando analisado o comportamento de Usuários Randômicos nota-se que o CCE-MP é mais eficiente. No entanto, o comportamento de usuários reais aproxima-se mais do perfil Polarizado, como ilustrado na Equação (2), o que tende a favorecer o método proposto.

Em um segundo experimento foi avaliado o consumo de banda em relação à variação da popularidade dos vídeos da

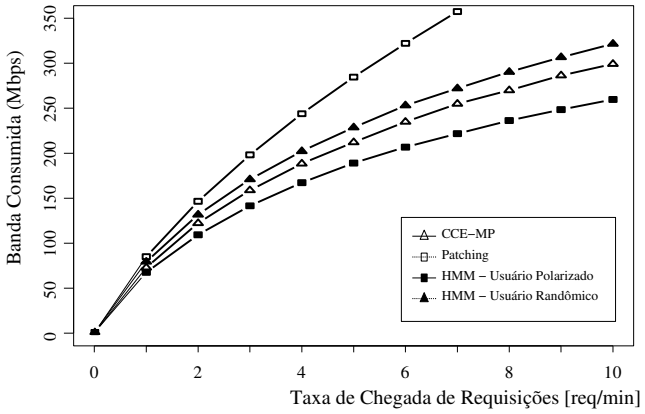

Fig. 2. Impacto da taxa de requisições λ para $\alpha = 0.8$

biblioteca do servidor, representada matematicamente pelo parâmetro α da Lei de Zipf. Os resultados são apresentados na Fig.3. É possível notar que para baixos valores de α (popularidade uniforme) o modelo proposto com Usuários Randômicos possui uma resposta similar ao CCE-MP tradicional, enquanto que com Usuários Polarizados existe uma economia de 21% no consumo de banda. Alguns estudos mostram que o parâmetro α da Lei de Zipf varia entre 0.5-1.0 [18] [9] e o modelo proposto com Usuários Polarizados se sobressai em toda faixa de interesse. Para $\alpha > 1.2$ o método CCE-MP torna-se mais eficiente que o modelo proposto com Usuários Polarizados. Isso ocorre, pois a partir deste valor de α a Lei de Zipf faz com que poucos vídeos sejam mais acessados dos que os demais da biblioteca, ou seja, a partir deste ponto é mais eficiente utilizar a popularidade dos vídeos e não a preferência do usuário. No entanto, os estudos citados indicam que o parâmetro α normalmente é menor que 1.0 para sistemas VoD.

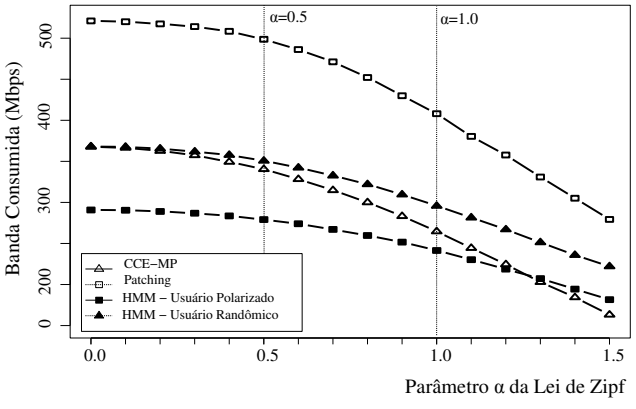

Fig. 3. Impacto do parâmetro α da Lei de Zipf para $\lambda = 10$ req/min

Um terceiro experimento foi realizado utilizando dados reais de usuários. Uma HMM foi treinada com dados de 43 usuários do serviço VoD da Netflix. Com isso foi possível fazer um estudo do desempenho em um cenário com $N = 320$ vídeos e $M = 16$ categorias. Foi considerado que cada categoria possui o mesmo número de vídeos. A Fig. 4 mostra o consumo

de banda em relação a taxa de requisições. Nota-se que o modelo proposto economizou 59.53% da banda da rede para $\lambda = 8$ req/min se comparado ao CCE-MP. Esta margem tende aumentar quando a taxa de requisições λ aumentar. A Fig. 5 mostra o impacto do parâmetro α para $\lambda = 10$ req/min considerando os dados reais de usuários. Observa-se que o desempenho do método proposto é muito superior ao CCE-MP. Também é possível observar que a variação do parâmetro α leva a uma grande variação de desempenho do CCE-MP.

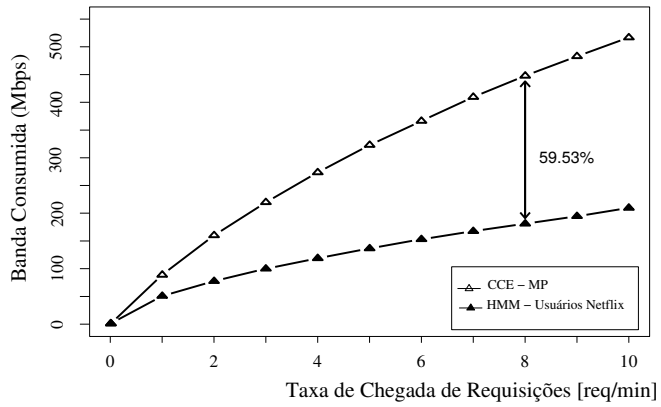

Fig. 4. Impacto da taxa de requisições λ para $\alpha = 0.8$, $N = 320$, $M = 16$ e HMM treinada com o algoritmo Baum-Welch a partir de registros de 43 assinantes da Netflix

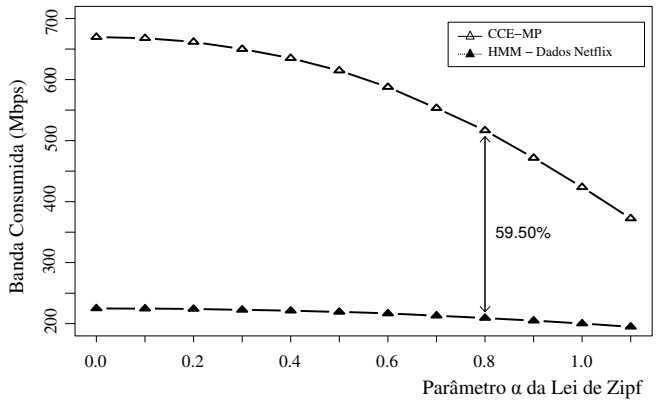

Fig. 5. Impacto do parâmetro α da Lei de Zipf para $\lambda = 10$ req/min, $N = 320$, $M = 16$ e HMM treinada com o algoritmo Baum-Welch a partir de registros de 43 assinantes da Netflix

V. Conclusões e Trabalhos Futuros

Neste trabalho sugerimos realizar a alocação do IVS utilizando as preferências do usuário, utilizando para isso o HMM. Através de simulações computacionais pudemos demonstrar que este modelo possui desempenho superior ao modelo de referência (CCE-MP) e pode ser utilizado conjuntamente com a transmissão *multicast* e armazenamento no dispositivo do cliente de forma a gerar uma economia da banda no núcleo da rede. É importante notar que a margem do ganho se altera conforme o comportamento do usuário se altera e os modelos de Usuários Polarizados e Randômicos foram utilizados como referência, pois em um sistema real é esperado que o comportamento do usuário esteja em algum lugar entre estes modelos. Os algoritmos atuais não exploram as preferências do usuário e portanto utilizam-se do perfil randômico do usuário.

Trabalhos futuros envolvem a avaliação da capacidade de armazenamento do dispositivo do usuário, quantidade de estados do HMM, quantidade de gêneros utilizados na classificação do conteúdo da biblioteca do servidor VoD e ainda a análise da frequência de troca de IVS dos usuários.

Referências

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2016-2021," White Paper - Cisco public, Tech. Rep., 2017.

[2] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, pp. 13:1–13:19, Dec. 2015. [Online]. Available: http://doi.acm.org/10.1145/2843948

[3] H. Feng, Z. Chen, and H. Liu, "Design and optimization for vod services with adaptive multicast and client caching," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2017.

[4] B. Xia, C. Yang, and T. Cao, "Modeling and analysis for cache-enabled networks with dynamic traffic," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2506–2509, Dec 2016.

[5] A. Azgin, G. AlRegib, and Y. Altunbasak, "Cooperative delivery techniques to support video-on-demand service in iptv networks," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2149–2161, Dec 2013.

[6] A. Dan, D. Sitaram, and P. Shahabuddin, "Scheduling policies for an on-demand video server with batching," in *Proceedings of the Second ACM International Conference on Multimedia*, ser. MULTIMEDIA '94. New York, NY, USA: ACM, 1994, pp. 15–23. [Online]. Available: http://doi.acm.org/10.1145/192593.192614

[7] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services," in *Proceedings of the Sixth ACM International Conference on Multimedia*, ser. MULTIMEDIA '98. New York, NY, USA: ACM, 1998, pp. 191–200. [Online]. Available: http://doi.acm.org/10.1145/290747.290771

[8] L. Gao and D. Towsley, "Threshold-based multicast for continuous media delivery," *IEEE Transactions on Multimedia*, vol. 3, no. 4, pp. 405–414, Dec 2001.

[9] C. Jayasundara, M. Zukerman, T. A. Nirmalathas, E. Wong, and C. Ranaweera, "Improving scalability of vod systems by optimal exploitation of storage and multicast," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 489–503, March 2014.

[10] H. Feng, Z. Chen, and H. Liu, "Minimizing bandwidth requirements for vod services with client caching," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–7.

[11] P. Ren and X. Wang, "User preference and behavior pattern in push vod systems," in *2014 IEEE 5th International Conference on Software Engineering and Service Science*, June 2014, pp. 425–429.

[12] Y. Xiao, J. Lai, and Y. Liu, "A user participation behavior prediction model of social hotspots based on influence and markov random field," *China Communications*, vol. 14, no. 5, pp. 145–159, May 2017.

[13] C. M. Grinstead and J. L. Snell, *Introduction to Probability*. American Mathematical Society, 2nd edition, 1998.

[14] P. Larue, P. Jallon, and B. Rivet, "Modified k-mean clustering method of hmm states for initialization of baum-welch training algorithm," in *2011 19th European Signal Processing Conference*, Aug 2011, pp. 951–955.

[15] M. Claeys, N. Bouten, D. D. Vleeschauwer, W. V. Leekwijck, S. Latre, and F. D. Turck, "Cooperative announcement-based caching for video-on-demand streaming," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 308–321, June 2016.

[16] "Netflix Brasil," https://www.netflix.com/br/, acessado: 01/04/2018.

[17] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2018. [Online]. Available: https://www.R-project.org/

[18] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 4, pp. 333–344, Apr. 2006. [Online]. Available: http://doi.acm.org/10.1145/1218063.1217968