UNIVERSIDADE FEDERAL DO PARANÁ

FELIPE ALEX PINTO

APLICAÇÃO DE MODELOS OCULTOS DE MARKOV NA DISTRIBUIÇÃO DE VÍDEO SOB DEMANDA

CURITIBA

2018

FELIPE ALEX PINTO

APLICAÇÃO DE MODELOS OCULTOS DE MARKOV NA DISTRIBUIÇÃO DE VÍDEO SOB DEMANDA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração Telecomunicações, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como parte das exigências para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Carlos Marcelo Pedroso

CURITIBA

2018

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

P659a Pinto, Felipe Alex
Aplicação de modelos ocultos de Markov na ̃distribuição de vídeo sob demanda / Felipe Alex Pinto. – Curitiba, 2018.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, 2018.

Orientador: Carlos Marcelo Pedroso .

1. Vídeos para Internet. 2. Sistemas de comunicação em banda larga. 3. Hidden Markov Models. I. Universidade Federal do Paraná. II. Pedroso, Carlos Marcelo. III. Título.

CDD: 004.61

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

MINISTÉRIO DA EDUCAÇÃO
SETOR TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA ELÉTRICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **FELIPE ALEX PINTO** intitulada: **APLICAÇÃO DE MODELOS OCULTOS DE MARKOV NA DISTRIBUIÇÃO DE VÍDEO SOB DEMANDA**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 05 de Setembro de 2018.

CARLOS MARCELO PEDROSO
Presidente da Banca Examinadora (UFPR)

KEIKO VERONICA ONO FONSECA
Avaliador Externo (UFPR)

EVELIO MARTÍN GARCÍA FERNÁNDEZ
Avaliador Interno (UFPR)

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Carlos Marcelo Pedroso pela atenção e ensinamentos, que ao longo desta jornada sempre me incentivaram a seguir em frente. Agradeço também aos demais professores do PPGEE/UFPR que também contribuíram para minha formação.

Agradeço a minha família pelo incentivo para buscar novos desafios e pela compreensão da minha ausência. Em especial agradeço ao meu irmão, Murilo Álvaro Pinto, que me ajudou no tratamento dos históricos de acesso de vídeos.

Agradeço aos amigos, colegas, conhecidos e demais pessoas que contribuíram de forma voluntária com este trabalho compartilhando seus históricos de vídeos acessados.

Por fim, agradeço as pessoas que estiveram ao meu lado nas horas difíceis e que iluminaram meu caminho tornando meus dias mais felizes.

RESUMO

O uso de aplicações de vídeo sob demanda já domina o tráfego atual da Internet e continua com uma forte perspectiva de crescimento nos próximos anos. Existem algoritmos desenvolvidos com o objetivo de reduzir o consumo de banda e reduzir a carga de processamento do servidor durante a distribuição de vídeo. Estes algoritmos baseiam-se majoritariamente na popularidade dos vídeos existentes na biblioteca do servidor. Algoritmos mais recentes exploram a capacidade de armazenamento do dispositivo do usuário para alocar segmentos iniciais dos vídeos, o que permite ao servidor a distribuição mais eficiente explorando melhor a capacidade de transmissão *multicast* da rede além de proporcionar ao usuário início imediato do vídeo requisitado. Nesta dissertação propõe-se um método de distribuição de vídeo capaz de prever a categoria do conteúdo que será acessado pelo usuário através de um Modelo Oculto de *Markov* (HMM, *Hidden Markov Model*), de modo a aumentar a eficiência na distribuição do vídeo. O desempenho do método foi avaliado através de um simulador baseado em eventos discretos desenvolvido em linguagem C. As análises comparam o impacto de diferentes taxas de requisições recebidas pelo servidor de vídeo, diferentes padrões de popularidade dos vídeos e para diferentes capacidades de armazenamento do dispositivo do usuário. Os resultados indicam que o uso do método pode diminuir significativamente o consumo de banda na rede IP durante a transmissão de vídeo quando comparado com métodos existentes. Com a aplicação do método proposto é possível atender um maior número de requisições com o mesmo *hardware*, o que pode ser visto também como uma redução de custo de implementação para servidores VoD.

Palavras-chave: Vídeo sob demanda. Distribuição de vídeo. Hidden Markov Models.

ABSTRACT

The use of video-on-demand applications have already overcome current Internet traffic and continues with a strong growth prospect in the coming years. There are algorithms designed to reduce bandwidth consumption and reduce the server processing load during video distribution. These algorithms are mostly based on the popularity of existing videos in the server library. More recent algorithms explore the storage capacity of the user's device to allocate initial segments of the videos, which allows the server to more efficiently distribute by exploiting the network's multicast transmission capability, in addition to providing the user with immediate video start-up requested. In this study we propose a video distribution method capable of predicting the category of content that will be accessed by the user through a Hidden Markov Model (HMM) in order to increase efficiency in video distribution. The performance of the method was evaluated through a discrete event-based simulator developed in C language. The analyzes compare the impact of different rates of requests received by the video server, different patterns of video popularity and different storage capacities of the user's device. The results indicate that the use of the method can significantly reduce bandwidth consumption in the IP network during video transmission when compared to existing methods. With the application of the proposed method it is possible to meet a larger number of requests with the same hardware, which can also be seen as a reduction of implementation cost for VoD servers.

Key words: Video-on-demand. Video distribution. Hidden Markov Models.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

CCE-MP	Client Caching Enabled Multicast Patching
D-PLO	Dynamic Programming-based Prepopulation Length Optimization
GIP	Greedy IVS Placement
GRASAR	Greedy Replica Allocation based on Square rooted Arrival Rate
HMM	Hidden Markov Model
IP	Internet Protocol
IPTV	Television over IP
ISP	Internet Service Provider
IVS	Initial Video Segment
PAB-MP	Prepopulation Assisted Batching with Multicast Patching
STB	set-top-box
UFPR	Universidade Federal do Paraná
VoD	video-on-demand
VoIP	voice over IP

LISTA DE SÍMBOLOS

μ	Multiplicador de Lagrange
λ	Taxa de uma distribuição de Poisson
α	Parâmetro de inclinação da distribuição de Zipf
b_i	Banda consumida pela transmissão do vídeo i
B	Banda total consumida
N	Número total de vídeos
r	Taxa de transmissão dos vídeos
L	Duração dos vídeos
l_i	Duração do IVS
C	Capacidade de armazenamento no dispositivo do usuário
π	Vetor de probabilidades iniciais do HMM
P	Matriz de probabilidade de transição do HMM
W	Matriz de probabilidade de emissão do HMM
K	Número de estados
M	Número de categorias
O	Sequência de observações
Q	Sequência de estados
τ	modelo HMM

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTO

A popularização do acesso à Internet se dá ao mesmo passo que os *Internet Service Providers* (ISPs) expandem suas redes e com o passar dos anos as tecnologias de acesso das redes banda larga evoluíram de tal forma que é possível transmitir dados com taxas mais altas dando subsídio para que novos serviços fossem oferecidos aos usuários. Os serviços providos via internet banda larga são comumente associados ao serviço *triple play*, onde o usuário tem acesso a transmissão de dados (Internet), voz (VoIP, *voice over IP*) e TV sobre *Internet Protocol* (IP). Dentre estas aplicações a transmissão de vídeo sob demanda (VoD, *video on demand*) se torna cada vez mais popular entre os usuários, fazendo que o tráfego de vídeo 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 (CISCO, 2017).

Este contínuo aumento de usuários associado com o aumento da variedade de mídias disponíveis nas plataformas VoD fazem com que os servidores VoD tenham que enfrentar problemas de escalabilidade, tanto do ponto de vista da distribuição deste tráfego, bem como do ponto de vista do *hardware*, uma vez que existe um limite físico de acesso aos discos rígidos que armazenam os conteúdos. Isso faz com que os servidores fiquem mais caros e com redundância de conteúdo. É importante notar que as ISPs nem sempre são a provedora do serviço VoD, as ISPs em sua premissa inicial são responsáveis por todas as camadas da rede, desde a rede de acesso até o núcleo da rede, onde há conectividade com servidores externos e outras ISPs. Além do aspecto físico, as ISPs precisam garantir a qualidade do tráfego dos usuários e isso passará a ser um desafio com o aumento do tráfego adicional VoD proveniente de provedores externos.

O serviço de VoD se popularizou, pois diferente do modelo de TV convencional, que permite ao usuário acesso a conteúdo *broadcast* com programação definida pela emissora, o modelo VoD possibilita que os usuários tenham maior interatividade com o sistema, como acesso a conteúdos de vídeo a qualquer momento e a possibilidade de parar, avançar e assistir novamente qualquer conteúdo. Alguns provedores VoD aplicam algoritmos para recomendar conteúdos o que gera uma alteração no comportamento do usuário (GOMEZ-URIBE; HUNT, 2015). Os problemas inerentes ao transporte de tráfego VoD em uma rede IP a nível de distribuição podem ser segmentado na análise de consumo de banda no núcleo da rede e escalabilidade do sistema. Como este tipo de tráfego é altamente previsível, por exemplo, conteúdos mais populares são acessados com uma frequência maior, existem diversas abordagens que utilizam deste fato para reduzir a ocupação de banda e aumentar a possibilidade de atender mais usuários durante a distribuição de vídeos (FENG; CHEN; LIU, 2017).

De forma geral, com técnicas mais robustas as ISP/provedores VoD serão capazes de atender mais usuários sem comprometer a banda disponível no núcleo da rede e a escalabilidade do sistema, otimizando os recursos durante a distribuição de tráfego VoD. Existem várias técnicas para otimizar a entrega de tráfego VoD que são discutidas na literatura. Técnicas de distribuição de vídeo baseadas em *unicast*, onde cada vídeo requisitado é entregue de forma pontual, não supre a necessidade atual de otimização de recursos da rede. Porém, técnicas mais atuais, que levam em consideração a frequência de acesso dos vídeos, como *patching*, *batching*, *user caching* se mostram mais promissoras, apresentado uma melhor eficiência.

O aumento da capacidade de armazenamento dos dispositivos dos usuários, permitiu que novas técnicas de distribuição de vídeo explorassem essa capacidade. Alguns algoritmos como *patching* utilizam essa capacidade para armazenar temporariamente mais de um fluxo de vídeo relativo ao vídeo requisitado. Contudo, alguns algoritmos como *Prepopulation Assisted Batching with Multicast Patching* (PAB-MP) apresentado por (JAYASUNDARA et al., 2014) e *Client Caching Enabled Multicast*

Patching (CCE-MP) apresentado por (FENG; CHEN; LIU, 2016), além de utilizarem o armazenamento temporário do vídeo requisitado, exploram a capacidade do dispositivo de armazenar trechos iniciais de alguns vídeos (IVS, *initial video segment*) na memória não-volátil do dispositivo, permitindo assim que o usuário possa assistir ao IVS local oportunizando ganhos com transmissões *multicast* já em andamento.

1.2 OBJETIVOS

1.2.1 Objetivo geral

O objetivo deste trabalho é propor e avaliar um método de distribuição de vídeo sob demanda que utiliza as preferências do usuário para prever o conteúdo acessado e melhorar o índice de acerto na memória *cache* local. Para prever o conteúdo que será assistido pelo usuário do serviço VoD, utilizar essa informação para diminuir o consumo de banda no núcleo da rede e aumentar a eficiência na distribuição de vídeos, através da alocação dos IVSs baseado na preferência do usuário e comparar com outros algoritmos presentes na literatura.

1.2.2 Objetivos específicos

Os objetivos específicos desta dissertação são:

- Criar algoritmo baseado em comportamento de usuário, considerando as categorias dos vídeos acessados pelo mesmo;
- Avaliar o desempenho do método proposto comparado com outros algoritmos encontrados na literatura;
- Desenvolver um simulador de eventos discretos em linguagem C para avaliação do algoritmo.

1.3 ESTRUTURA DA DISSERTAÇÃO

Além deste capítulo introdutório, esta dissertação está estruturada da seguinte forma. O Capítulo 2 descreve os conceitos fundamentais que englobam a distribuição de vídeo que são relevantes para o entendimento do método proposto, mostrando a topologia considerada e evidenciando os algoritmos de distribuição de vídeo existentes na literatura que são mais relevantes. O Capítulo 3 descreve o método proposto e as técnicas utilizadas. No Capítulo 4 é discutida a metodologia aplicada e os resultados obtidos. Por fim, no Capítulo 5 são discutidas as conclusões e trabalhos futuros.

2 VÍDEO SOB DEMANDA

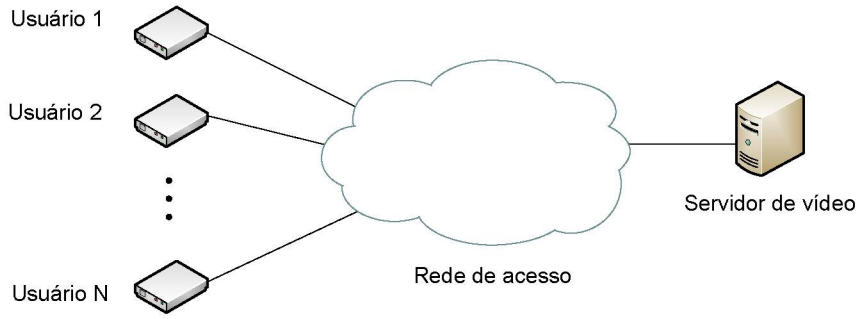
2.1 DISTRIBUIÇÃO E CARACTERIZAÇÃO DE TRÁFEGO VOD

O serviço VoD ou *streaming* de vídeo pode ser ofertado através de diversas tecnologias. Existem provedores do serviço que utilizam o acesso à *Internet* do usuário através de uma conexão banda larga fixa ou móvel, outros utilizam acesso a cabo, porém independente da tecnologia utilizada o sistema de distribuição de vídeo pode ser caracterizado de forma simplificada através dos elementos: servidor de vídeo, rede de distribuição e usuários do serviço. Outros elementos que são comumente encontrados nas rede de ISPs, como roteadores, *switches* e outros servidores não foram considerados nesta topologia, pois os mesmos não são elementos impactantes na análise de algoritmos de distribuição de vídeo. Outros aspectos na transmissão de vídeos como atraso, perda de pacotes, priorização do tráfego e codificação do conteúdo não são objetos de estudo neste trabalho. A Figura 2.1 mostra como estão dispostos estes elementos, onde:

- Servidor de Vídeo: É o elemento que armazena todos os vídeos disponíveis em um sistema de VoD, recebe as requisições dos usuários e determina quando iniciar ou parar a transmissão para a rede de acesso de um determinado vídeo requisitado, de acordo com o algoritmo de distribuição aplicado.
- Rede de Distribuição: É o elemento que interconecta o servidor de vídeo ao usuário, este geralmente é o núcleo da rede de uma ISP, que engloba roteadores, *switches* e pode ter políticas de controle de tráfego nestes elementos de acordo com o estado de ocupação do núcleo da rede, ou seja, pode haver descartes de pacotes de vídeos caso a rede esteja com alta ocupação e com isso diminuir a percepção da qualidade do serviço VoD para o usuário.
- Usuário: É o elemento que sinaliza para o servidor de vídeo todas as operações

do usuário, como: requisição, saída ou parada de um vídeo. Este pode ser um *set-top-box* (STB), computador ou outro dispositivo terminal do cliente que gere estas sinalizações dentro do sistema VoD.

Figura 2.1: Topologia de um sistema de distribuição de vídeo



A troca de informação entre elementos dentro de uma rede IP pode se dar a partir de maneiras diferentes: *unicast*, *broadcast* e *multicast*. A transmissão *unicast* é a transmissão entre dois elementos da rede, ou seja, existem apenas um remetente e um destinatário da informação. A transmissão *broadcast* é definida por ter apenas um remetente e todos os elementos do domínio da rede serão os destinatários da informação, enquanto que a transmissão *multicast* apenas os remetentes interessados na informação irão receber a mesma. Em sistemas VoD a transmissão de vídeos utilizando *unicast* onera o sistema devido à redundância do tráfego enviado pelo servidor de vídeo, que gera um consumo maior da banda disponível para transmissão. Sistemas VoD que utilizam algoritmos de distribuição baseados em *multicast* mostram-se mais eficientes.

Existem métodos que exploram outros aspectos da transmissão de vídeos em uma rede IP para reduzir o consumo de banda no núcleo da rede e manter a qualidade do vídeo percebida pelo usuário. Alguns métodos fazem a compressão dos dados transmitidos utilizando diferentes *codecs* para transmissão de cada vídeo, outros algoritmos sugerem métodos de descarte seletivo de pacotes em caso de congestionamento na rede para que a percepção da qualidade do vídeo pelo usuário não se altere. Outros métodos ao identificarem congestionamento podem fazer a transmissão com taxas

mais baixas forçando uma redução na resolução do vídeo, desta forma a transmissão ocupa menos banda na rede até que o sistema se normalize e haja banda disponível para transmissão com a taxa máxima.

As requisições que chegam a um servidor VoD a partir de todos os usuários do sistema são normalmente caracterizadas por um processo de Poisson (BAO et al., 2012) com taxa λ. Estas requisições geram uma diferente popularidade para cada vídeo, que a literatura sugere que segue a distribuição de Zipf (WANG et al., 2002), onde a probabilidade de acesso de um vídeo i presente na biblioteca do servidor VoD é dada por

$$p_i = \frac{\gamma}{i^{\alpha}}; \quad \gamma = \frac{1}{\sum_{j=1}^{N} \frac{1}{j^{\alpha}}}; \quad \sum_{j=1}^{N} p_j = 1, \tag{2.1}$$

onde: p_i é a probabilidade de requisição do vídeo i; α é o parâmetro de inclinação, que define o grau de polarização dos acessos; N número total de vídeos na biblioteca do servidor VoD, e; γ é uma constante de normalização para que a soma de todas as probabilidades p_i tenha valor unitário.

A equação 2.1 considera que os vídeos estão organizados pela popularidade, do vídeo mais popular para o menos popular, desta forma tem-se que $p_1 > p_2 > \cdots > p_N$ para os casos onde $\alpha > 0$ ou $p_1 = p_2 = \cdots = p_N$ para $\alpha = 0$. A Figura 2.2 mostra um exemplo do impacto do parâmetro α da Lei de Zipf para um sistema onde $N = 100$. Observa-se que quanto maior o valor de α mais acentuada é a inclinação da reta, ou seja, uma menor quantidade de vídeos terão a maior probabilidade de acesso e serão responsáveis pela maior quantidade de requisições. Na literatura existem referências à utilização da distribuição de Zipf para descrever a popularidade dos vídeos com o parâmetro α entre 0.5-1.0 (YU et al., 2006)(CLAEYS et al., 2016).

2.2 ALGORITMOS DE DISTRIBUIÇÃO DE TRÁFEGO VOD

A transmissão de vídeos em uma rede IP demanda uma estratégia de distribuição aqui apresentada como algoritmos de distribuição de tráfego VoD. Os algoritmos clássicos

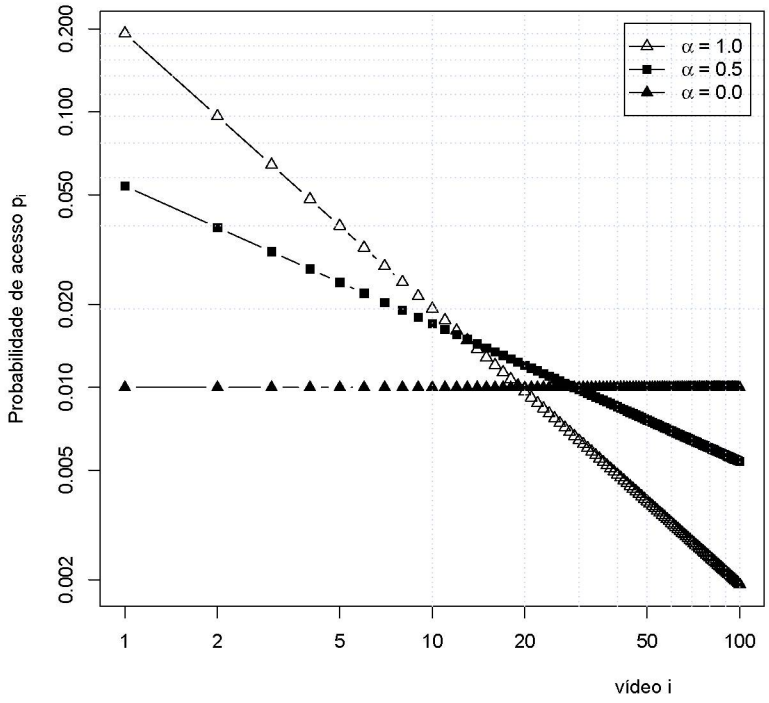


Figura 2.2: Distribuição de Zipf com diferentes valores de α

para este fim baseiam-se em técnicas de *unicast* e *broadcast*, algoritmos mais recentes encontrados na literatura exploram o conhecimento da popularidade dos vídeos e capacidade de armazenamento no dispositivo do usuário. Nesta sessão serão discutidos os algoritmos mais relevantes.

2.2.1 Batching

Considere que várias requisições para um mesmo vídeo v_i cheguem ao servidor VoD, representadas por setas verticais na Figura 2.3. Em uma janela de tempo definida por $[t_{r,1}, T + t_{r,1}]$, onde $t_{r,1}$ é o instante de tempo que o servidor VoD recebeu a primeira requisição da primeira rajada de requisições e T é chamado de janela *batching*, o servidor VoD iniciará a transmissão do vídeo v_i através de um fluxo *multicast* no instante $T + t_{r,1}$ atendendo todas as requisições que chegaram dentro da janela *batching* (DAN; SITARAM; SHAHABUDDIN, 1994). As requisições que cheguem após a janela *batching* fará com que o servidor VoD agende o início de uma segunda transmissão

do vídeo v_i no instante $T + t_{r,2}$, onde $t_{r,2}$ denota o instante de tempo que houve a primeira requisição do vídeo v_i da segunda rajada de requisições para este vídeo. Este processo se repete para todos os vídeos da biblioteca do servidor VoD e durante todo o tempo de operação do sistema VoD.

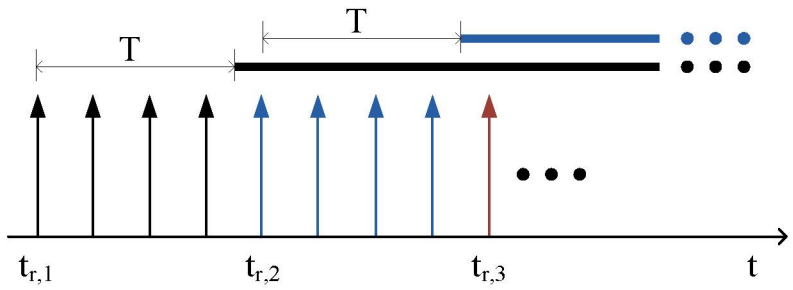


Figura 2.3: Algoritmo *batching* para distribuição de vídeo

Neste cenário, quanto maior o número de requisições dentro do intervalo $[t_{r,n}, T + t_{r,n}]$ melhor será o desempenho do algoritmo. Contudo, os vídeos menos populares fazem com que o algoritmo *batching* seja reduzido a uma transmissão *unicast*, transmitindo de forma redundante parte dos vídeos e onerando o sistema. Este algoritmo adiciona um atraso para o início do vídeo, que pode ser até T para as requisições que chegam no instante $t_{r,n}$, o que pode motivar os usuários a abandonarem a plataforma antecipadamente.

2.2.2 Patching

O algoritmo *patching* explora a capacidade de armazenamento temporário do dispositivo do usuário, de modo que o mesmo possa receber mais de um fluxo de dados (EAGER; VERNON; ZAHORJAN, 2001). Desta forma, para uma dada requisição do vídeo v_i que ocorra no instante t_r o servidor VoD iniciará imediatamente a transmissão do vídeo com um fluxo *multicast* principal. Requisições de v_i que cheguem no servidor após t_r, o servidor verificará se há algum fluxo *multicast* principal sendo transmitido, caso não haja o servidor iniciará um novo fluxo *multicast* de v_i imediatamente, porém caso haja um fluxo principal de v_i sendo transmitido, o usuário irá armazenar o

fluxo principal, a partir do momento da requisição, e o servidor iniciará imediatamente um fluxo *unicast* para transmissão da porção faltante de v_i chamado de *patching*. A duração do fluxo *patching* pode ser definida como $t_{r,g} - t_{r,n}$, onde $t_{r,g}$ é o instante de tempo onde houve requisição que gerou o fluxo *multicast* principal e $t_{r,n}$ é o instante de tempo da requisição atual.

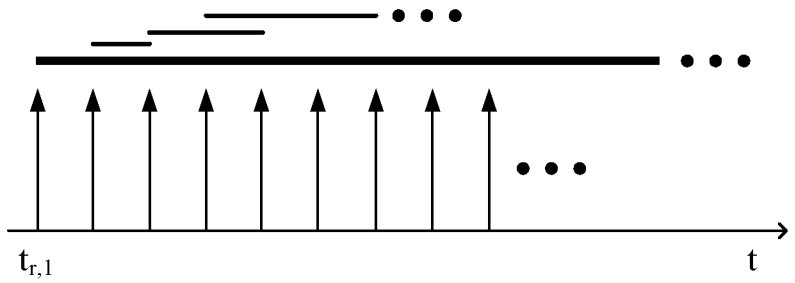


Figura 2.4: Algoritmo *patching* para distribuição de vídeo

A Figura 2.4 mostra o comportamento do servidor VoD frente a requisições de um mesmo vídeo v_i, onde as linhas horizontais representam o fluxo de dados transmitidos pelo servidor, incluindo o fluxo principal e *patching*. Este algoritmo elimina o atraso para início do vídeo, porém as requisições que chegam no final de um fluxo principal de v_i geram fluxos $patchings$ com longa duração, o que diminui a eficiência do algoritmo. Para reduzir esse efeito e utilizar os fluxos principais de maneira mais eficiente, foi proposto a utilização de um limiar ótimo, que garante uma economia de consumo de banda no núcleo da rede (GAO; TOWSLEY, 2001), dado por

$$T_i = \frac{\sqrt{2L_i\lambda_i + 1} - 1}{\lambda_i}, \tag{2.2}$$

onde L_i é a duração total do vídeo v_i e λ_i é a taxa de requisições de v_i que chegam ao servidor VoD. O limiar ótimo T_i, chamado também de janela *patching*, delimita o intervalo de tempo onde o servidor poderá gerar fluxos *patching*. As requisições que chegam fora da janela *patching* o servidor iniciará um novo fluxo principal de v_i para a primeira requisição e as demais requisições que ocorram dentro da nova janela *patching* serão atendidas com o mesmo fluxo principal. Esta variação do algoritmo

patching é apresentada na Figura 2.5.

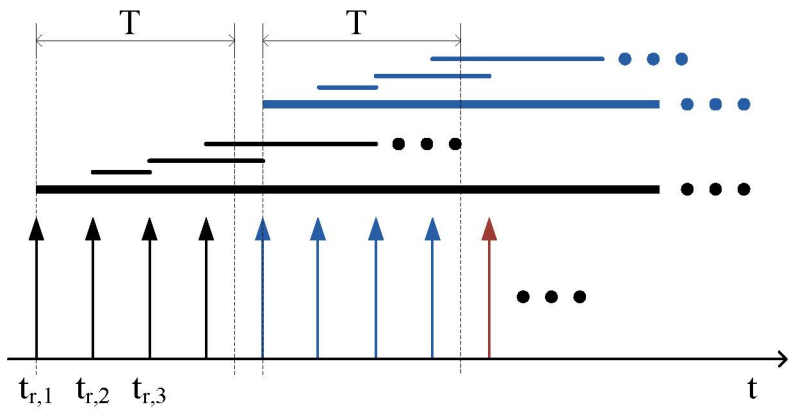


Figura 2.5: Algoritmo *patching* com limiar ótimo para distribuição de vídeo

Os vídeos menos populares ($\lambda_i \rightarrow 0$) a janela *patching* tende a ser a duração do vídeo, ou seja, o algoritmo aceitará gerar fluxos *patching* a qualquer instante, fazendo com que o sistema não use de forma eficiente os fluxos principais.

2.2.3 PAB-MP

O algoritmo *Prepopulation Assisted Batching with Multicast Patching* (PAB-MP), combina os benefícios dos algoritmos *batching* (subsessão 2.2.1) e *patching* (subsessão 2.2.2) com a capacidade de armazenamento dos dispositivos dos usuários para reduzir o consumo de banda na rede. Além da capacidade de receber diferentes fluxos de um mesmo vídeo v_i requisitado pelo usuário, o PAB-MP explora a capacidade do dispositivo do usuário de armazenar segmentos iniciais dos vídeos da biblioteca do servidor VoD (IVS, *initial video segment*) (JAYASUNDARA et al., 2014).

Quando uma requisição para o vídeo v_i chega ao servidor VoD, o usuário começa a assistir ao respectivo IVS que pode estar armazenado localmente no dispositivo do usuário ou compartilhado de outro usuário através de uma conexão *peer-to-peer*. Durante este período o servidor VoD 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*, ou seja, o tempo máximo que o servidor VoD poderá aguardar antes de iniciar a transmissão de um fluxo *multicast*. Para requisições que ocorram após esse limite o servidor enviará um fluxo *patching multicast*, que será atrasado em um tempo definido pelo duração do IVS, de forma a atender mais requisições com o mesmo fluxo *patching*, requisições que cheguem após a janela *patching*, conforme definido pela Equação 2.2, serão atendidas com um novo fluxo *multicast*.

Os IVS são transmitidos para os dispositivos do usuário nos momentos de baixo uso da rede e do servidor VoD. (JAYASUNDARA et al., 2014) sugerem o uso dos algoritmos D-PLO (*Dynamic programming-based Prepopulation Lengths Optimization*), GRASAR (*Greedy Replica Allocation based on Square rooted Arrival Rate*) e GIP (*Greedy IVS Placement*), para determinar a duração, a alocação e definir as réplicas de cada IVS que será armazenado no dispositivo do usuário, respectivamente. Desta forma o método introduz complexidade ao servidor e gera tráfego entre os dispositivos dos usuários.

2.2.4 CCE-MP

A abordagem *Client Caching Enabled with Multicast Patching*(CCE-MP) é um algoritmo que explora a capacidade do dispositivo do usuário de armazenamento de IVS e de recebimento de múltiplos fluxos *multicast* de um mesmo vídeo (FENG; CHEN; LIU, 2016). Ao se fazer uma requisição de um dado vídeo v_i no tempo t_r, o dispositivo do usuário imediatamente iniciará a tocar o IVS e começará a armazenar todos os fluxos ativos de *multicast* de segmentos de v_i. Estes segmentos de v_i são armazenados no dispositivos do usuário e não estão necessariamente na ordem cronológica do vídeo, com isso o dispositivo do usuário deverá reproduzi-los no momento correto para o usuário, 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. Como os dispositivos de usuários possuem capacidade limitada, os autores sugerem o algoritmo *Water-Filling* para alocação dos IVSs, alcançando assim uma melhora na eficiência do sistema, cujo consumo de banda pode ser calculado com

$$b_i = r \ln \left(\frac{L - l_i}{l_i + \frac{r}{\lambda_i}} + 1 \right); \quad B = \sum_{i=1}^{N} b_i; \tag{2.3}$$

onde: b_i é a banda consumida pelo vídeo v_i; B é a banda total consumida; r é a taxa de transmissão do vídeo; L é a duração total do vídeo; l_i é a duração do IVS do vídeo v_i; λ_i é a taxa de requisição para o vídeo v_i.

Os autores consideraram que r e L são os mesmos para todos os vídeos, sem perda de generalidade. Os autores também mostram que o CCE-MP possui um desempenho superior ao PAB-MP e *Batching*. Por este motivo, o CCE-MP foi escolhido para comparação com o método proposto nesta dissertação.

ALGORITMO WATER-FILLING

O algoritmo *Water-Filling* é utilizado pelo CCE-MP para resolver o problema de alocação dos IVSs no dispositivo do usuário. A solução ótima deste problema é determinar a duração do IVS de cada vídeo (l_i) de tal forma que (2.3) seja minimizada, sujeito às restrições

$$\textstyle\sum_{i=1}^{N} l_i \leq C \tag{2.4}$$

$$0 \leq l_i \leq L, \forall i \in \{1, 2, ..., N\}, \tag{2.5}$$

onde C é a capacidade de armazenamento do dispositivo de cada usuário. Considerando a função Lagrangeana

$$\mathcal{L} = \sum_{i=1}^{N} b_i + \mu \left(\sum_{i=1}^{N} l_i - C \right),$$

com μ sendo o multiplicador de Lagrange, e as condições de otimalidade *Karush-Kuhn-Tucker* (KKT), tem-se

$$\frac{\partial \mathcal{L}}{\partial l_i} = -\frac{r}{l_i + \frac{r}{\lambda_i}} + \mu = 0 \tag{2.6}$$

$$\frac{\partial \mathcal{L}}{\partial \mu} = \sum_{i=1}^{N} l_i - C = 0. \tag{2.7}$$

Assumindo $\beta = r/\mu$ (2.6) pode ser reescrita como

$$l_i = \beta - \frac{r}{\lambda_i}. \tag{2.8}$$

Considerando a restrição (2.5), tem-se

$$l_i = \min\left(\left(\beta - \frac{r}{\lambda_i}\right)^+, L\right), \tag{2.9}$$

sabendo que $x^+ = \max(x, 0)$. O valor de β pode ser determinado a partir da soma dos IVSs para todos os N vídeos. Considerando (2.7) e que $\lambda_i = p_i\lambda$, tem-se

$$\begin{aligned} \sum_{i=1}^{N} l_i = \sum_{i=1}^{N}\left(\beta - \frac{r}{\lambda_i}\right) &= C \\ N\beta &= C + \sum_{i=1}^{N}\frac{r}{\lambda_i} \\ \beta &= \frac{1}{N}\left[C + \frac{r}{\lambda}\sum_{i=1}^{N}\frac{1}{p_i}\right]. \end{aligned} \tag{2.10}$$

O algoritmo *Water-Filling* dará preferência para os vídeos mais populares, isto é, os vídeos mais populares terão o IVS maior em relação aos menos populares, sendo que a duração do IVS de cada vídeo é dada por (2.9) onde β é definida de acordo com (2.10).

Com isso, observa-se que o método PAB-MP e CCE-MP utilizam o IVS de forma

estática, ou seja, uma vez que sejam definidos os IVSs que estarão presentes no dispositivo do usuário, o sistema não tem a capacidade de atualizar os mesmos de acordo com a preferência do usuário. O próximo Capítulo apresenta um método baseado na preferência do usuário para tornar a alocação dos IVSs dinâmica, de modo a capacidade de armazenamento do usuário seja utilizada de forma mais eficiente.

3 MÉTODO PROPOSTO

Os algoritmos mais recentes de distribuição de vídeo exploram a capacidade de armazenamento dos dispositivos do usuário. Neste capítulo é apresentado um método para prever a categoria do conteúdo que será acessado pelo usuário de modo que a capacidade de armazenamento do usuário seja utilizada de forma mais eficiente. Para tanto, este aspecto será modelado através de um Modelo Oculto de *Markov*. Uma vez identificado o estado mais provável de um determinado usuário utilizando um HMM, o servidor poderá utilizar a capacidade do dispositivo do usuário para armazenar os IVSs dos vídeos de acordo com a probabilidade de requisição da categoria do vídeo.

3.1 MODELO OCULTO DE MARKOV

O Modelo Oculto de *Markov* (HMM, *Hidden Markov Models*) é uma ferramenta matemática que permite modelar processos estocásticos *Markovianos*, ou seja, processos onde o próximo estado depende apenas do estado atual. Este modelo é amplamente utilizado em reconhecimento de voz, análise de sequência de DNA e entre outras áreas do conhecimento. O HMM, assim como as cadeias de *Markov*, possuem estados que estão individualmente associados a uma distribuição de probabilidade de ocorrência de um evento, sendo que somente os eventos são observáveis e não os estados. Desta forma, um HMM é caracterizado por três conjuntos de probabilidades: probabilidade de estado inicial (π), probabilidade de transição entre os estados (P) e probabilidade de emissão de um evento (W). Em um HMM com K estados e M eventos observáveis, cada uma destas probabilidades podem ser escritas na forma matricial da seguinte forma

$$\pi = [\pi_1 \quad \pi_2 \quad \dots \quad \pi_K], \quad \sum_{i=1}^{K} \pi_i = 1 \tag{3.1}$$

$$\mathsf{P} = \begin{array}{c} \\ E_1 \\ E_2 \\ \vdots \\ E_K \end{array} \begin{array}{c} \begin{array}{ccccc} E_1 & E_2 & E_3 & \dots & E_K \end{array} \\ \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1K} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{K1} & p_{K2} & p_{K3} & \cdots & p_{KK} \end{bmatrix} \end{array}, \quad \sum_{i=1}^{K} p_{ki} = 1, \forall k \tag{3.2}$$

$$\mathsf{W} = \begin{array}{c} \\ E_1 \\ E_2 \\ \vdots \\ E_K \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & \dots & e_M \end{array} \\ \begin{bmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{1M} \\ w_{21} & w_{22} & w_{23} & \cdots & w_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{K1} & w_{K2} & w_{K3} & \cdots & w_{KM} \end{bmatrix} \end{array}, \quad \sum_{i=1}^{M} w_{ki} = 1, \forall k. \tag{3.3}$$

Os elementos π_i denotam a probabilidade do sistema ser iniciado no estado i, p_{ij} denotam a probabilidade de transição do estado i para o estado j, enquanto os elementos w_{ij} denotam a probabilidade de observação do evento j dado um estado i. Ainda é comum encontrar representações na forma de grafos ou pela notação $\tau = (P, W, \pi)$. Na Figura 3.1 é mostrado um grafo com $K = 3$ estados, E_1, E_2 e E_3, e W_k representa a probabilidade de emissão dos eventos no estado E_k ou simplesmente a linha k da matriz W.

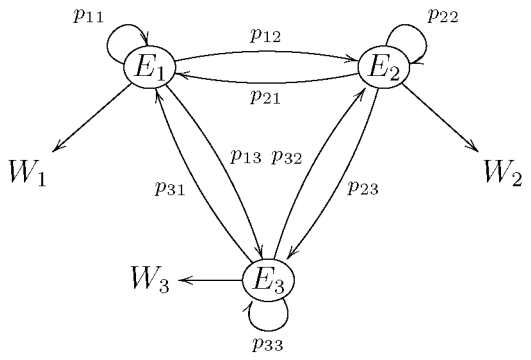


Figura 3.1: Exemplo de Modelo Oculto de Markov ($K = 3$)

Dentro de aplicações de um HMM existem três problemas principais inerentes que foram estudados por (RABINER; JUANG, 1986). Estes problemas são:

1. *Avaliação*: Dado um HMM τ e uma sequência de eventos observados $O = o_1, o_2, ..., o_T$, este problema quer definir qual a probabilidade da sequência O ter sido gerada pelo HMM τ, em resumo, é buscado $P(O|\tau)$;

2. *Decodificação*: Dado um HMM τ e uma sequência de eventos observados $O = o_1, o_2, ..., o_T$, este problema quer definir qual a probabilidade da sequência de estados Q mais provável que gerou a sequência O, em notação matemática deseja-se saber $P(Q|O, \tau)$;

3. *Aprendizado*: Neste problema deseja-se estabelecer os parâmetros de um HMM τ a partir de sequências de eventos observados ditos observações de treinamento.

Os problemas acima citados são conhecidos na literatura e comumente resolvidos com os algoritmos *forward-backward*, *Viterbi* e *Baum-Welch*(DUGAD; DESAI, 1996).

3.2 APLICAÇÃO DO HMM NA DISTRIBUIÇÃO DE TRÁFEGO VOD

Nesta subseção é apresentado o método para se utilizar um HMM em um ambiente de distribuição de conteúdo VoD, como apresentado na Figura 2.1, onde é considerado uma topologia simplificada evidenciando o servidor VoD, usuários do sistema e a rede de transporte. Nesta topologia o servidor VoD possui as seguintes funcionalidades:

1. *Biblioteca de Vídeos*: armazena todos os vídeos e suas características, como duração, taxa de transmissão, popularidade e classificação de gênero;

2. *Gerenciador de Transmissão de Vídeo*: este núcleo do servidor definirá o método utilizado para distribuição de vídeo;

3. *Gerenciador de IVS por usuário*: este núcleo é responsável por definir qual o estado do HMM presente do usuário para que o servidor possa fazer a alocação dos IVS de acordo com as probabilidades de emissão do estado correspondente.

Suponha um usuário que possui a seguinte sequência de acessos conhecida: Drama, Drama, Drama, Comédia, Suspense, Suspense, Drama, Ação, Comédia, Drama. Observando essa sequência pode-se dizer que dentro dos dez acessos observados deste usuário foram acessados 50% Drama, 20% Comédia, 20% Suspense e 10% Ação. Essa informação pode ser utilizada para prever a próxima categoria que será acessada pelo usuário.

Contudo, realizar a previsão desta forma leva a dois problemas. Primeiro, será necessário determinar quantas observações anteriores seriam levadas em consideração para prever a próxima categoria acessada pelo usuário, segundo, esta forma não contempla facilmente mudanças de comportamento do usuário. Essa mudança de comportamento pode ser gerada a partir de uma mudança de interesse do usuário ou mesmo induzida por sistemas de recomendação (GUPTA; MOHARIR, 2016).

A proposta desta dissertação é modelar a preferência de conteúdo acessado pelos usuários através de um Modelo Oculto de *Markov*. Desta forma os problemas citados no parágrafo anterior serão automaticamente resolvidos, pois através de um HMM toda a sequência de categorias já acessadas pelo usuário pode ser utilizada para identificar o estado mais provável que o usuário se encontra, e as devidas probabilidades de ocorrência das categorias definidas por este estado. Neste modelo cada estado do HMM representa um perfil de usuário, com as respectivas probabilidades de ocorrência de cada uma das categorias de vídeo.

Para ilustrar a aplicação do HMM neste caso, suponha um servidor com as características apresentadas na Tabela 3.1 e um HMM ($K = 3$), cuja matriz de probabilidade de emissão é dada por

$$\mathsf{W} = \begin{array}{c} \\ E_1 \\ E_2 \\ E_3 \end{array} \begin{array}{c} \begin{array}{ccccc} \textit{Drama} & \textit{Romance} & \textit{Suspense} & \textit{Seriado} & \textit{Terror} \end{array} \\ \begin{bmatrix} 0.4 & 0.1 & 0.1 & 0.2 & 0.2 \\ 0.1 & 0.3 & 0.4 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 & 0.3 & 0.3 \end{bmatrix} \end{array}. \tag{3.4}$$

Note que o vetor W_k (linha k da matriz W) corresponde a probabilidade de acesso das M categorias $\{Drama, Romance, Suspense, Seriado, Terror\}$ e que a capacidade de armazenamento C dos usuários será dividida de acordo com W_k para a alocação dos IVSs. Seja c_{km} o armazenamento que estará disponível para o estado k e categoria $m \in \{1, \quad 2, \quad \cdots, \quad M\}$, então pode-se dizer que $c_{km} = C \cdot w_{km}$; e ainda considerando que existe o mesmo número de vídeos para cada categoria e supondo que para os vídeos pertencentes a mesma categoria a popularidade possui uma distribuição de Zipf, propomos determinar todos os IVSs associados à w_{km} utilizando o algoritmo *Water-Filling* com as restrições $0 \leq l_i \leq L$ e $\sum_{i=(m-1)\cdot N/M+1}^{m \cdot N/M} l_i \leq c_{km}$.

Tabela 3.1: Parâmetros

Parâmetro	Descrição	Valor
N	Número de Vídeos	50
M	Número de Categorias	5
K	Número de Estados do HMM	3
α	Parâmetro da Lei de Zipf	1.0
r	Taxa do Vídeo	2 Mbps
L	Duração do Vídeo	900 MB (1h)
C	Capacidade de Armazenamento	1800 MB (2h)
λ	Taxa de requisições	10 requisições/min

Neste trabalho é considerado que cada conteúdo possui apenas uma classificação em relação ao gênero, isto é, o vídeo pode ser classificado como Drama, Comédia, Terror/Horror, Ficção, Séries, etc. Suponha um usuário, cujo estado mais provável atual seja E_1, onde a probabilidade de emissão é dada por $W_1 = [0.4\ 0.1\ 0.1\ 0.2\ 0.2]$, isso implica que o dispositivo do usuário está usando seu espaço de armazenando de IVS proporcionalmente à coluna W_1 da Tabela 3.2. Ao receber uma requisição deste usuário, o servidor VoD iniciará a transmissão de acordo com as diretrizes do gerenciador de transmissão do vídeo e irá definir qual é o estado mais provável do usuário. Esta definição é feita resolvendo o problema apresentado na sessão 3.1. Uma vez definido o estado mais provável do usuário, o servidor VoD fará a atualização dos IVSs do referido usuário de acordo com a probabilidade de transição de estado. Ainda é possível que conforme os usuários façam mais requisições aumentando os

dados no histórico, o servidor VoD poderá fazer um novo treinamento dos parâmetros P e W, de forma que o HMM possa fazer uma melhor previsão.

O servidor VoD pode fazer a atualização dos IVSs dos usuários em um momento oportuno de baixa utilização da rede para que não cause impactos no consumo de banda e ainda o HMM pode ter seus parâmetros atualizados periodicamente através dos histórico de acesso dos usuários. Desta forma o método será capaz de se adaptar para proporcionar a maior eficiência frente ao novo padrão de acesso.

Cada estado do HMM representa um perfil de usuário, ou seja, um padrão de acesso das categorias dos vídeos. O número total de estados K do modelo HMM depende de quantos perfis de usuário são possíveis descrever dentro da população de usuários. Este número pode ser determinado a partir de técnicas de clusterização.

Para avaliar o modelo proposto foram criados cenários para simulação que são explicados no próximo Capítulo.

Tabela 3.2: Duração dos IVSs

v_i	m	$l_i[MB](W_1)$	$l_i[MB](W_2)$	$l_i[MB](W_3)$
1	1	91.77053571	37.77053571	37.77053571
2	1	87.37708333	33.37708333	33.37708333
3	1	82.98363095	28.98363095	28.98363095
4	1	78.59017857	24.59017857	24.59017857
5	1	74.19672619	20.19672619	20.19672619
6	1	69.80327381	15.80327381	15.80327381
7	1	65.40982143	11.40982143	11.40982143
8	1	61.01636905	7.016369048	7.016369048
9	1	56.62291667	0.852380952	0.852380952
10	1	52.22946429	0	0
11	2	37.77053571	73.77053571	55.77053571
12	2	33.37708333	69.37708333	51.37708333
13	2	28.98363095	64.98363095	46.98363095
14	2	24.59017857	60.59017857	42.59017857
15	2	20.19672619	56.19672619	38.19672619
16	2	15.80327381	51.80327381	33.80327381
17	2	11.40982143	47.40982143	29.40982143
18	2	7.016369048	43.01636905	25.01636905
19	2	0.852380952	38.62291667	20.62291667
20	2	0	34.22946429	16.22946429
21	3	37.77053571	91.77053571	37.77053571
22	3	33.37708333	87.37708333	33.37708333
23	3	28.98363095	82.98363095	28.98363095
24	3	24.59017857	78.59017857	24.59017857
25	3	20.19672619	74.19672619	20.19672619
26	3	15.80327381	69.80327381	15.80327381
27	3	11.40982143	65.40982143	11.40982143
28	3	7.016369048	61.01636905	7.016369048
29	3	0.852380952	56.62291667	0.852380952
30	3	0	52.22946429	0
31	4	55.77053571	37.77053571	73.77053571
32	4	51.37708333	33.37708333	69.37708333
33	4	46.98363095	28.98363095	64.98363095
34	4	42.59017857	24.59017857	60.59017857
35	4	38.19672619	20.19672619	56.19672619
36	4	33.80327381	15.80327381	51.80327381
37	4	29.40982143	11.40982143	47.40982143
38	4	25.01636905	7.016369048	43.01636905
39	4	20.62291667	0.852380952	38.62291667
40	4	16.22946429	0	34.22946429
41	5	55.77053571	37.77053571	73.77053571
42	5	51.37708333	33.37708333	69.37708333
43	5	46.98363095	28.98363095	64.98363095
44	5	42.59017857	24.59017857	60.59017857
45	5	38.19672619	20.19672619	56.19672619
46	5	33.80327381	15.80327381	51.80327381
47	5	29.40982143	11.40982143	47.40982143
48	5	25.01636905	7.016369048	43.01636905
49	5	20.62291667	0.852380952	38.62291667
50	5	16.22946429	0	34.22946429

4 ANÁLISE DE DESEMPENHO

Neste capítulo são apresentados os materiais utilizados e a metodologia aplicada para avaliação de desempenho do modelo proposto.

4.1 MATERIAIS

Inicialmente foi desenvolvido um simulador de eventos discretos em linguagem C para reproduzir os resultados dos algoritmos *patching* (GAO; TOWSLEY, 2001) e CCE-MP (FENG; CHEN; LIU, 2016), que são utilizados para comparação com o método proposto. Posteriormente foi desenvolvido um simulador para o método proposto, também em linguagem C. O desempenho foi avaliado primeiramente através de simulações computacionais que utilizando casos extremos, ou seja, usuários que só requisitavam conteúdo de uma mesma categoria (comportamento polarizados) e usuários que não demonstravam preferência por nenhuma categoria de conteúdo (comportamento randômico). Na sequência foram coletados 50 históricos de usuários do Netflix fornecidos por colaboradores desta dissertação. Os vídeos destes históricos foram classificados com uma das 16 categorias mais populares para avaliação do método proposto. Todas as simulações computacionais foram realizadas em um servidor com o sistema operacional Linux disponível no Laboratório de Sistemas de Comunicação do Departamento de Engenharia Elétrica da UFPR. Além dos simuladores de evento discreto foram criadas rotinas em *bash* para automatização das simulações.

4.2 METODOLOGIA

A metodologia utilizada para o uso do simulador consiste em três etapas. A primeira delas é a definição do modelo HMM, isto é, a definição dos parâmetros P, W e π; a segunda parte é a geração de padrões de acessos de usuários. Nesta etapa será

gerada a sequência de categorias que cada usuário irá requisitar durante a simulação obedecendo o modelo definido anteriormente. A terceira e última etapa é a simulação, onde cada usuário fará uma requisição de acordo com a sequência de categorias pré-estabelecidas na etapa anterior. A Figura 4.1 mostra o diagrama de dependências destas etapas.

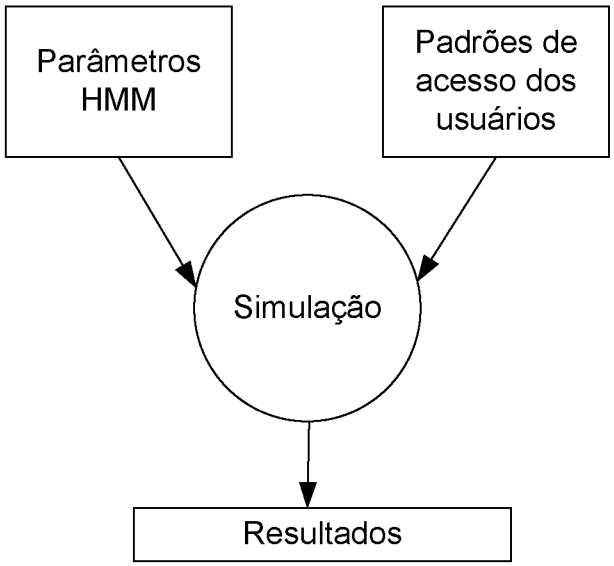


Figura 4.1: Esquemático das fases de simulação

A avaliação do método foi realizada a partir da análise de três variáveis: taxa de requisições, popularidade dos vídeos e capacidade de armazenamento. Cada conjunto de parâmetros foram simulados cinco vezes para obtenção do intervalo de confiança.

4.2.1 Definição dos Parâmetros do HMM

Um HMM é definido por suas matrizes de probabilidade de transição (P), probabilidade de emissão (W) e probabilidade de estado inicial (π). Neste trabalho foram realizadas simulações com três características de usuários, isto significa que foram definidos três HMM diferentes. Foram definidos modelos teóricos para representar casos extremos de usuários, ou seja, usuários que não tinham preferência por nenhum tipo de conteúdo chamado, Comportamento Randômico e aqueles usuários que tem

preferência acentuada por um tipo de conteúdo, chamado Comportamento Polarizado. O terceiro e último HMM foi definido através de dados reais, para estimar o comportamento do método proposto frente a situações encontradas em um sistema real.

4.2.1.1 Comportamento Randômico

Nestes casos os usuários não possuem preferência definida por tipos específicos de conteúdo. Este tipo de comportamento pode ser descrito através de um HMM, onde a matriz P tem uma probabilidade equivalente de transição para todos os estados, enquanto que a matriz W possui probabilidades iguais para todos os possíveis eventos. Isso faz com que os eventos observáveis sejam equiprováveis. Considerando um HMM ($K = 3$) e um conjunto de cinco possíveis categorias, as matrizes P, W e π podem ser escritas como

$$P = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}, \quad W = \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix}, \quad \pi = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}.$$

4.2.1.2 Comportamento Polarizado

Os usuários com comportamento polarizado são caracterizados por terem preferência para um tipo de categoria e terem alta probabilidade de permanecer assistindo o mesmo conteúdo, como no fenômeno *binge watching* (CLAEYS et al., 2016), onde o usuário assiste diversos episódios de um mesmo seriado sequencialmente. Este fato, faz com que os vídeos requisitados pelo usuário pertençam à mesma categoria enquanto o usuário estiver no mesmo estado. Este comportamento é descrito por um HMM, cujos parâmetros são

$$P = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}, \quad W = \begin{bmatrix} 0.8 & 0.1 & 0.1 & 0 & 0 \\ 0 & 0.8 & 0.1 & 0.1 & 0 \\ 0 & 0 & 0.8 & 0.1 & 0.1 \end{bmatrix}, \quad \pi = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}.$$

4.2.1.3 Dados Reais

O HMM que descreve os Usuários Netflix foi treinado a partir de históricos de vídeos acessados por cinquenta usuários reais da plataforma Netflix que colaboraram com este trabalho. Os históricos obtidos contemplavam acessos em um período de 2.4 anos e compreendem 3597 títulos diferentes com mais de 40000 acessos. Os conteúdos assistidos pelos usuários foram classificados com uma das 16 categorias mais comuns: 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.

O treinamento do HMM foi realizado através do algoritmo *Baum-Welch* implementado no software R (R Core Team, 2018). Os dados dos 50 históricos foram concatenados, de modo que representassem as requisições recebidas por um único servidor VoD. Esta suposição não interfere na análise em termos da sequência de categorias recebidas pelo servidor VoD, uma vez que em um histórico de requisições obtido não é possível afirmar que as requisições dos vídeos foram provenientes de um único indivíduo. Após o treinamento, os parâmetros do HMM obtido foram

$$P = \begin{bmatrix} 0.98 & 0.00 & 0.02 \\ 0.26 & 0.27 & 0.47 \\ 0.00 & 0.91 & 0.09 \end{bmatrix}, \quad \pi = \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix},$$

$$W = \begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & & \\ 0.11 & 0.08 & 0.11 & 0.03 & 0.06 & 0.04 & 0.04 & 0.06 & \ldots & \\ 0.14 & 0.08 & 0.10 & 0.04 & 0.07 & 0.05 & 0.04 & 0.04 & & \end{bmatrix}$$

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.97 & 0.00 & 0.00 \\ \ldots & 0.04 & 0.02 & 0.05 & 0.04 & 0.03 & 0.15 & 0.07 & 0.06 \\ 0.04 & 0.02 & 0.06 & 0.05 & 0.02 & 0.14 & 0.06 & 0.05 \end{bmatrix}.$$

As 16 colunas de W representam na mesma ordem as 16 categorias mencionadas no parágrafo anterior. Observa-se que o elemento $w_{1,14} = 0.97$ representa a probabilidade do usuário requisitar um vídeo da categoria $Seriado$ quando o mesmo encontra-se no estado E_1. Este fato demonstra que nos dados obtidos os usuários possuem um comportamento mais próximo do comportamento polarizado descrito na subseção 4.2.1.2. Os valores obtidos para W_2 e W_3, possuem valores semelhantes e que estão distribuídos entre todas as categorias, contudo através da análise da matriz P é possível observar que uma vez no estado E_3 o usuário tem 91% de probabilidade de transitar para o estado E_2. Enquanto que um usuário no estado E_2 tem probabilidade de 26% e 47% de chance de transitar para E_1 e E_3, respectivamente. Um usuário no estado E_1 tem 98% de chance de permanecer no mesmo estado.

4.2.2 Criação do padrão de acesso dos usuários

Para que seja possível fazer uma simulação do método proposto é preciso estabelecer a sequência de categorias que será acessada por cada usuário. Um gerador de acessos foi desenvolvido em linguagem C para este propósito; e este não faz parte do núcleo do simulador de eventos discretos do método proposto. Isto implica que o servidor VoD simulado não conhece os acessos definidos por este gerador de sequência de acessos. Para exemplificar a utilização do gerador de acessos assuma um conjunto de 5 categorias $\{A, B, C, D, E\}$, este gerador terá como entrada o modelo HMM(P,W,π) definido anteriormente, seja este para Comportamento Randômico, Polarizado ou Da-

dos Reais. Através de um gerador de números aleatórios o gerador de acessos simula a transição de estados no HMM e gera as requisições do mesmo, exemplo: Usuário 1: A-B-B-B-C-D-D-A-..., Usuário 2: E-E-E-E-E-A-A-B-.... O Algoritmo 1 mostra a lógica programada para o gerador de acessos.

Algoritmo 1: GERADOR DE SEQUÊNCIA DE CATEGORIA POR USUÁRIO

Entrada: $P, W, \pi, maxRequisicoes$

Saída: Sequência de Categorias (σ)

i: contador do número máximo de requisições que serão geradas
piAleatorio: número aleatório restrito ao intervalo [0,1]
estadoAtual: estado atual do usuário
PAleatorio: número aleatório restrito ao intervalo [0,1] que determinará a transição de estado
WAleatorio: número aleatório restrito ao intervalo [0,1] que determinará a categoria requisitada

início

$i = 0$
$piAleatorio = rand()$
$estadoAtual = EstadoInicial(\pi, piAleatorio)$
while $i < maxRequisicoes$ **do**
$PAleatorio = rand()$
$WAleatorio = rand()$
$\sigma\,[i] \leftarrow Categoria(W, estadoAtual, WAleatorio)$
$estadoAtual = EstadoAtual(P, PAleatorio, estadoAtual)$
$i++$
end

fim

retorna σ

4.2.3 Simulação

A simulação computacional do método proposto consiste em utilizar as informações do HMM em conjunto com a sequência de categorias que serão requisitadas por todos os usuários, para que assim se possa obter o consumo de banda no núcleo da rede do sistema em estado estacionário, desprezando assim as informações relativas ao *warm-up* da simulação.

O simulador é baseado em eventos discretos e realiza a geração de números aleatórios usando o método da inversa (JAIN, 1991). O intervalo entre chegadas de requisições foi gerado através de um processo de Poisson. A cada iteração da simulação o servidor VoD irá servir a requisição atual, atualizar os IVSs do usuário, cuja requisição foi servida, considerando a probabilidade do usuário transitar para outro estado do HMM.

Durante este processo o total de dados transmitidos do servidor VoD para os usuários, sem considerar a atualização dos IVSs, será salva para posteriormente calcular o consumo de banda total considerando o tempo simulado. Durante a operação do sistema o servidor VoD realiza atualizações dos IVSs no dispositivo do usuário, no cálculo de consumo de banda não foram considerados estes dados, pois estes dados podem ser transmitidos para o usuário em um tempo oportuno, onde a ocupação momentânea da rede seja baixa.

4.3 DESEMPENHO

Nesta sessão são apresentados os resultados das simulações computacionais do método proposto.

4.3.1 Análise da variação da taxa de requisições

Como mencionado nas sessões anteriores, foi utilizado HMM para modelar o comportamento do usuário em relação à categoria do conteúdo de vídeo requisitado. As simulações foram realizadas com três algoritmos de distribuição de vídeo: *Patching*, CCE-MP e o método proposto em uma topologia com $N = 100$ e $\alpha = 0.8$. Os demais parâmetros das simulações são apresentados na Tabela 3.1. Neste cenário foram considerados comportamento randômico e polarizado como descrito nas sessões 4.2.1.1 e 4.2.1.2, respectivamente. Foi considerado que todos os vídeos possuem a mesma taxa de transmissão e a mesma duração para que o resultado obtido não tivesse influência destas variáveis.

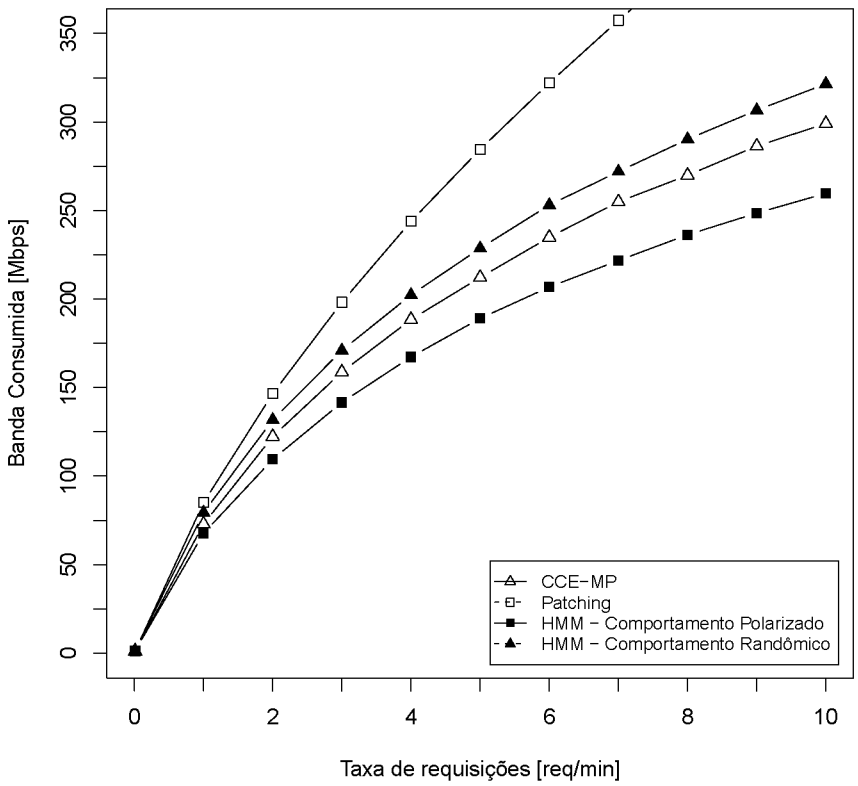


Figura 4.2: Impacto da Taxa de Requisições (λ)

A Figura 4.2 mostra os resultados obtidos em termos de largura de banda con-

sumida em função da taxas de requisições. Para um intervalo de confiança de 95% os resultados obtidos mostram uma variação máxima de ± 2Mbps. Nota-se que para taxas mais baixas ($\lambda < 2$) não é evidente o ganho em relação ao consumo de banda entre os métodos *Patching*, CCE-MP e o método proposto. Contudo, quando a taxa é mais alta ($\lambda > 8$) é evidente o ganho do método proposto, para o caso de comportamento polarizado. Isso ocorre, pois quando o comportamento do usuário é polarizado a capacidade de armazenamento do dispositivo do usuário é utilizada com IVSs de vídeos que tem a maior probabilidade de ser requisitado pelo usuário, pois o servidor VoD sabe qual é o estado mais provável do usuário. Entretanto, com comportamento randômico, o método proposto mostra um pior desempenho em relação ao CCE-MP. Isto ocorre devido ao fato do servidor VoD utilizar a capacidade de armazenamento do usuário de forma uniforme, ou seja, os vídeos pertencentes a mesma categoria terão um espaço disponível menor no dispositivo do usuário do que o CCE-MP. Porém foi observado que o comportamento de usuários reais tende a ser mais próximo do perfil polarizado. Isto foi observado nos históricos de acesso e esta previsibilidade também é reportada na literatura (GUPTA; MOHARIR, 2016)(HUANG et al., 2018).

4.3.2 Análise da popularidade dos vídeos

A popularidade dos vídeos da biblioteca do servidor VoD é caracterizada pelo parâmetro de inclinação α da Lei de Zipf apresentado na Equação 2.1. As simulações para esta análise foram feitas a partir dos dados apresentados na Tabela 3.1 com $N = 100$ e α variando de 0 até 1.5 com passo de 0.1. Foram utilizados os comportamentos randômicos e polarizados para comparação do método proposto com o *Patching* e CCE-MP.

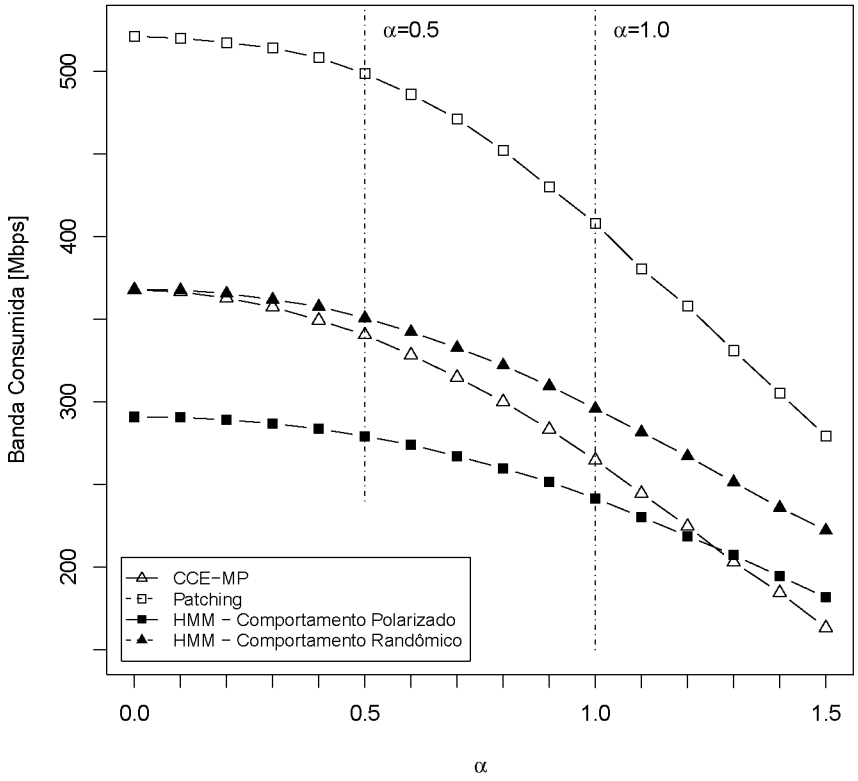


Figura 4.3: Impacto da Popularidade dos Vídeos (α)

A Figura 4.3 mostra que o método proposto com usuários randômicos e polarizados se sobressaem em relação ao *Patching*. Contudo, quando analisado o método CCE-MP, observa-se que o método proposto com usuários com comportamento randômico para valores de α próximos de zero se equivalem, isso se dá, pois com $\alpha = 0$ os vídeos são equiprováveis de serem acessados, com isso $\sum_{i=1}^{N} 1/p_i = 1$ em (2.10), isso implica que todos os vídeos terão a mesma duração do IVS no dispositivo do usuário. Porém quando o valor de α aumenta o método CCE-MP passa a ter um maior índice de acertos na distribuição de IVS, enquanto que o método proposto com usuários randômicos divide essa capacidade entre os vídeos de todas as categorias. Este fato faz com que o método CCE-MP se sobressaia em relação ao método proposto com usuários randômicos. O mesmo não ocorre quando analisa o método proposto operando com usuários com comportamento polarizado. Para baixos valores

de α, o método proposto aloca uma porção maior da capacidade de armazenamento para os vídeos, cujas categorias possuem a maior probabilidade de serem acessadas pelo usuário, aumentando a probabilidade de que o IVS esteja presente no STB do usuário. Contudo, quando o valor de α atinge um limiar em torno de 1.2 e 1.3 existe uma inversão, o método CCE-MP passa a ser mais eficiente que o método proposto quando utilizado com usuários com comportamento polarizado. Isto ocorre, pois neste cenário (valor de α alto) apenas alguns vídeos da biblioteca detém o maior número de requisições o que faz com que o CCE-MP utilize a capacidade de armazenamento do usuário com mais eficência. Porêm, os valores mais comuns reportados na literatura para α estão entre 0.5 e 1.0 (JAYASUNDARA et al., 2014), e nesta faixa de interesse o método proposto com usuários polarizados se sobressai em relação ao CCE-MP. Os resultados obtidos mostram uma variação máxima de $\pm$2Mbps com intervalo de confiança de 95%.

4.3.3 Análise da capacidade de armazenamento

A capacidade de armazenamento no dispositivo do usuário disponível para alocação dos IVSs tem impacto direto sobre a eficiência do método proposto. Este impacto foi analisado através de simulações com $N = 100$ e demais parâmetros da Tabela 3.1, através da métrica $C/(NL)$, que representa a capacidade de armazenamento de um único usuário pelo tamanho total utilizado para armazenamento de todos os vídeos da biblioteca do servidor VoD. Nesta análise foram verificados o comportamento dos métodos *Patching*, CCE-MP e o método proposto.

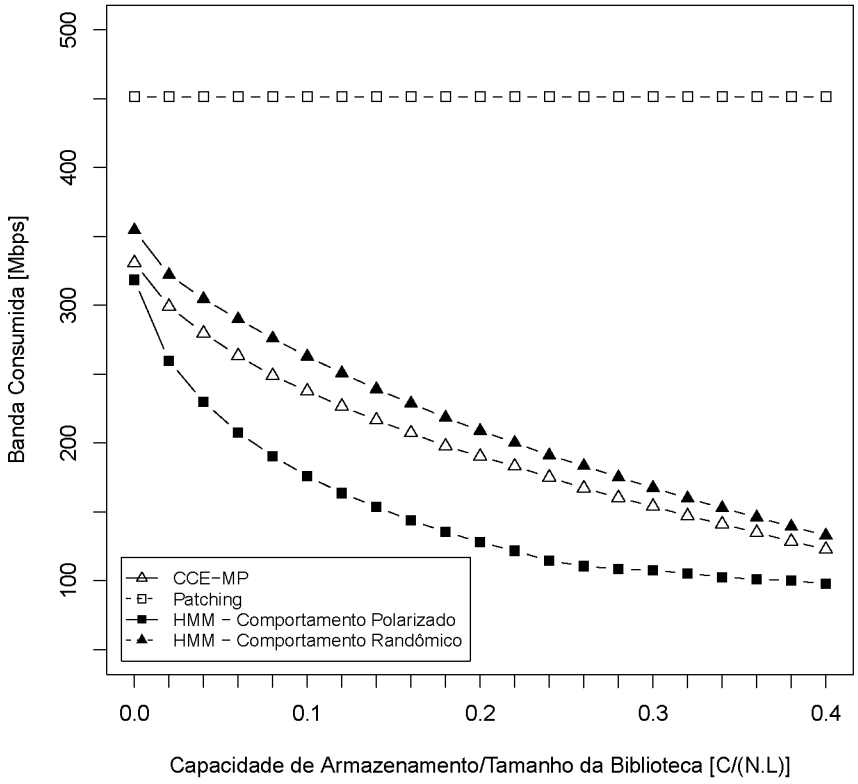


Figura 4.4: Impacto da Capacidade de Armazenamento ($C/(NL)$)

A Figura 4.4 mostra que o *Patching* não sofre alteração na resposta frente à diferentes capacidade de armazenamento, isso ocorre, pois o método apenas utiliza a popularidade dos vídeos e a capacidade de armazenamento não é explorada. O método proposto se sobressai em relação ao CCE-MP em toda a faixa analisada para usuários com comportamento polarizado, isso ocorre pois o método proposto explora com maior eficiência a capacidade de armazenamento do dispositivo do usuário. Nota-se que quanto maior for a capacidade de armazenamento a eficiência entre o método proposto e o CCE-MP diminui, isso ocorre, pois quanto maior for o espaço disponível no dispositivo do usuário maior será o número de IVS disponível no STB, consequentemente aumento da probabilidade de acerto, o que implica em uma melhor performance do método CCE-MP.

O método proposto com usuários com comportamento randômico mostrou uma

performance inferior ao CCE-MP em toda faixa analisada. Isso ocorre, pois usuários randômicos demandam uma distribuição da capacidade de armazenamento do dispositivo uniforme entre todas as categorias, enquanto que na mesma circunstância o método CCE-MP utiliza a capacidade de armazenamento para os vídeos mais populares, consequentemente melhora o desempenho. Para um intervalo de confiança de 95% os resultados obtidos mostram uma variação de $\pm$1Mbps.

4.3.4 Análise dos Dados Reais

Os comportamentos randômicos e polarizados, idealizados anteriormente, não descrevem o comportamento de um sistema real, entretanto, é esperado que o comportamento real esteja entre estes dois comportamentos. Alguns fenômenos como *binge watching* sugerem que o comportamento do usuário é mais próximo do polarizado. Para verificar esta tendência foram coletados dados do histórico do Netflix de 50 usuários como descrito na sessão 4.2.1.3. A análise do impacto da taxa de requisições e popularidade dos vídeos frente ao método proposto com usuários Netflix são comparados com o CCE-MP em uma topologia com os parâmetros $N = 320$, $M = 16$ e $\alpha = 0.8$. Os demais parâmetros são os apresentados na Tabela 3.1. Na Figura 4.5 é possível observar que foi possível reduzir o consumo de banda em 59.53% para $\lambda = 8$ requisições/minuto e este valor tende a aumentar conforme λ aumentar. Enquanto que na Figura 4.6 é possível observar um ganho de 59.50% em relação ao CCE-MP para um $\alpha = 0.8$, esta margem de ganho tende a diminuir quando o valor de α aumentar conforme discutido anteriormente. Ambos os resultados apresentaram uma variação máxima de $\pm$2Mbps com intervalo de confiança de 95%.

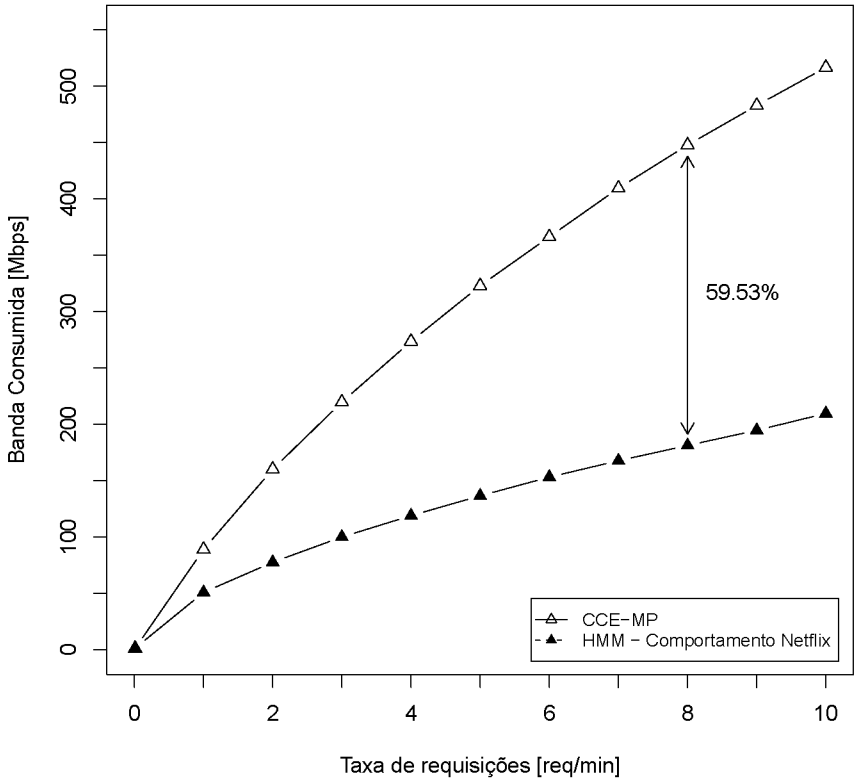


Figura 4.5: Impacto da Taxa de Requisições (λ) com HMM treinada com históricos de usuários do Netflix, com $\alpha = 0.8$

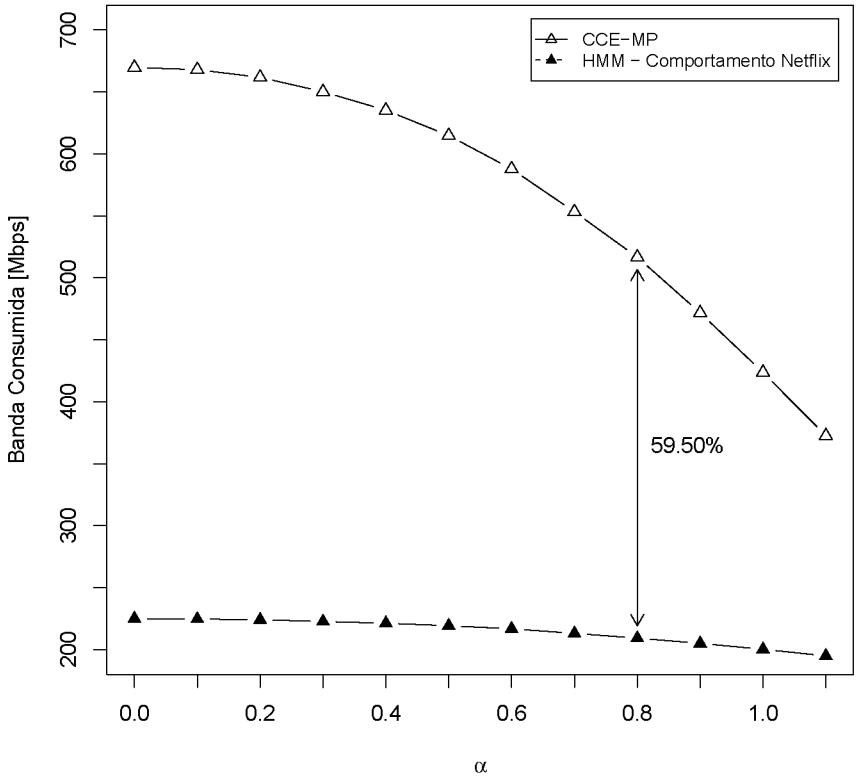


Figura 4.6: Impacto da Popularidade dos Vídeos (α) com HMM treinada com históricos de usuários do Netflix, com $\lambda = 10$ req/min

Os históricos de requisições obtidos mostram que existe uma forte polarização do comportamento do usuário, isso faz com que o método proposto tenha uma maior probabilidade de alocar os IVSs que serão requisitados. Com isso, existe um maior índice de acertos o que gera em uma redução no consumo de banda no núcleo da rede e diminui a eficiência do CCE-MP.

4.3.5 Análise do Impacto da Quantidade de Estados no Modelo Oculto de Markov (K)

As simulações realizadas anteriormente do método proposto foram baseadas em um modelo oculto de *Markov* com três estados, nesta subseção é analisado o impacto da

variação da quantidade de estados no modelo para diferentes taxas de chegadas e popularidade dos vídeos. Nestas simulações foram utilizados os mesmos dados reais apresentados na sessão 4.2.1.3 para treinar quatro modelos ocultos de *Markov* modelos com três, quatro, cinco e dez estados. A topologia utilizada para as simulações consiste nos parâmetros $N = 320$, $M = 16$ e $\alpha = 0.8$. Os demais parâmetros são os apresentados na Tabela 3.1.

A Figura 4.7 mostra o consumo de banda em função da taxa de requisições para diferentes modelos ocultos de *Markov*. É possível observar que não há uma diferença expressiva entre os resultados, quando considerado uma variação máxima de $\pm$2Mbps com intervalo de confiança de 95%.

Contudo, a Figura 4.8 mostra que o modelo HMM com $K = 4$ apresenta um resultado melhor em relação ao modelo $K = 3$, enquanto que $K = 5$ e $K = 10$ apresentam desempenho inferior. Com isso, podemos induzir que os históricos coletados podem ser representados por quatro diferentes perfis de usuário.

Os históricos de acesso real coletados são amostras de comportamento dentro de um sistema real, entretanto estas amostras não representam todo o sistema real, que possui muito mais usuários ativos. Isso implica que em um sistema real o modelo HMM teria um número de estados maior do que os utilizados para esta simulação. Este número de estados pode ser determinado a partir de métodos de clusterização como dendograma (EVERITT; SKRONDAL, 2002) e k-means (JAIN, 2010).

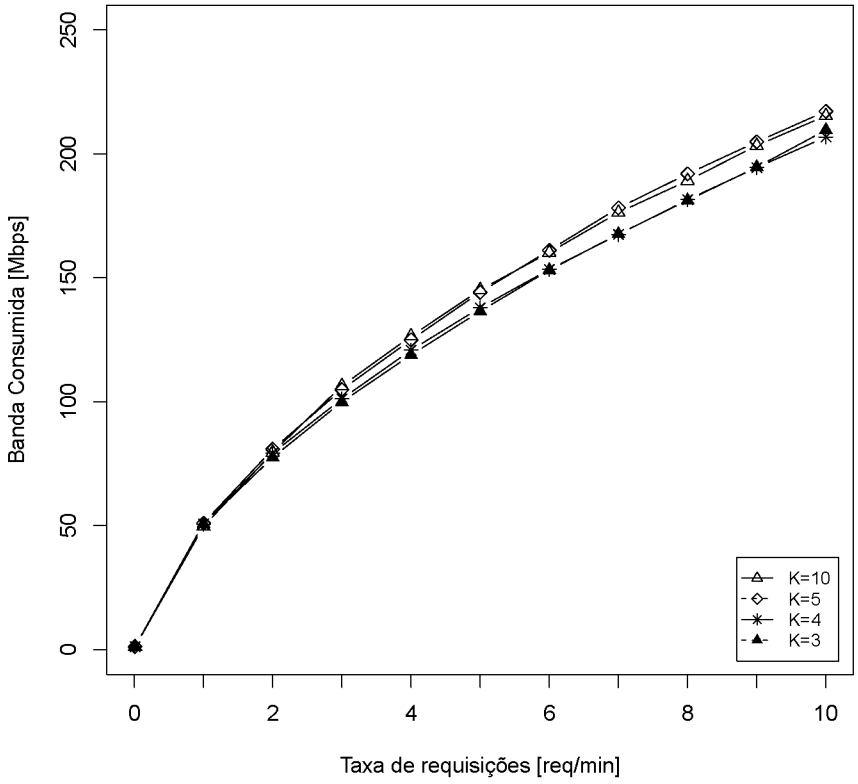


Figura 4.7: Impacto da Taxa de Requisições (λ) com HMM treinada com históricos de usuários do Netflix, com $\alpha = 0.8$

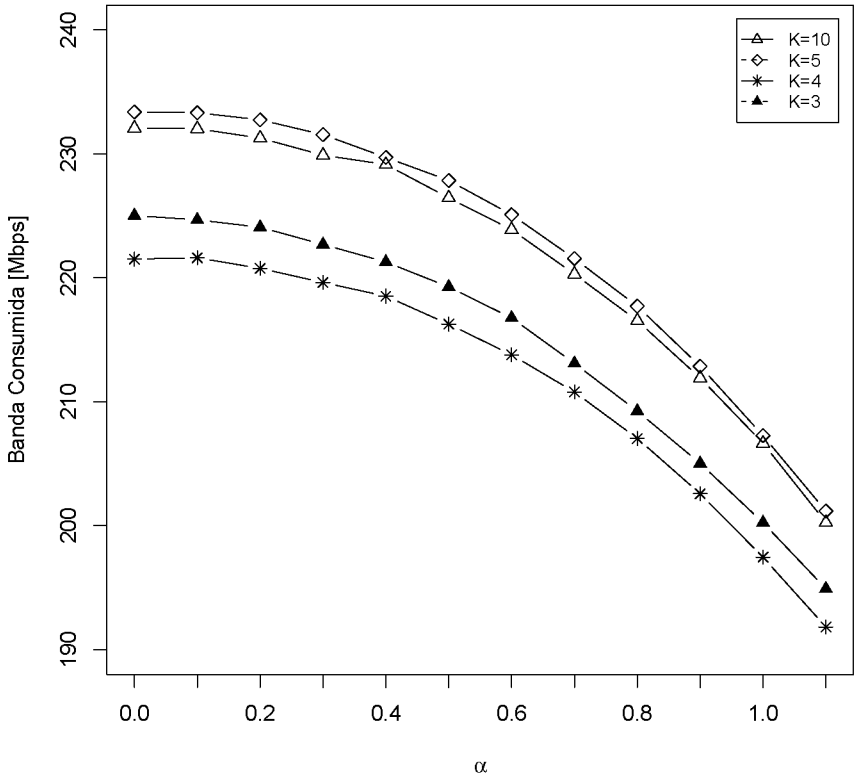


Figura 4.8: Impacto da Popularidade dos Vídeos (α) com HMM treinada com históricos de usuários do Netflix, com $\lambda = 10$ req/min

5 CONCLUSÕES E TRABALHOS FUTUROS

A contribuição deste trabalho foi a apresentação de um modelo que explora a previsibilidade do comportamento do usuário de plataformas de vídeo sob demanda para distribuição de vídeo através de um HMM. O modelo proposto mostra-se eficiente para distribuição de vídeo em diversos ambientes com diferentes taxas de requisição e popularidade dos vídeos. Quando comparado ao método atual encontrado na literatura, que não explora a previsibilidade do comportamento do usuário, os resultados foram promissores mostrando uma grande melhora quando usuários polarizados são modelados com o método proposto. Os resultados obtidos mostram que o uso da preferência do usuário é muito superior à popularidade dos vídeos para distribuição de VoD.

Durante as simulações foram treinados modelos ocultos de *Markov* com diferentes números de estados. Observou-se que existe um número ótimo que modela os dados de uma forma mais eficiente, entretanto é desejável um método para determinar qual o número ótimo de estados do HMM para caracterizar o comportamento dos usuários de diferentes plataformas VoD.

A abordagem deste trabalho foi utilizar o HMM como um classificador, onde o servidor VoD é responsável por determinar o estado atual do usuário. Porém ainda existe a possibilidade do dispositivo do usuário treinar um HMM local, isto é, cada usuário teria seu próprio classificador; e o sistema deveria ser avaliado através de duas formas: (a) os parâmetros do HMM (P e W) seriam treinados a partir dos dados de um único usuário, (b) utilização de um esquema ponto-a-ponto para que os usuários possam compartilhar IVS, de forma que o servidor VoD não precise transmitir o IVS, fazendo com que a carga no servidor diminua.

REFERÊNCIAS

BAO, Y. et al. An energy-efficient client pre-caching scheme with wireless multicast for video-on-demand services. In: **2012 18th Asia-Pacific Conference on Communications (APCC)**. 2012. p. 566–571. ISSN 2163-0771.

CISCO. **Cisco visual networking index: Forecast and methodology, 2016-2021**. 2017.

CLAEYS, M. et al. Cooperative announcement-based caching for video-on-demand streaming. **IEEE Transactions on Network and Service Management**, v. 13, n. 2, p. 308–321, June 2016. ISSN 1932-4537.

DAN, A.; SITARAM, D.; SHAHABUDDIN, P. Scheduling policies for an on-demand video server with batching. In: **Proceedings of the Second ACM International Conference on Multimedia**. New York, NY, USA: ACM, 1994. (MULTIMEDIA '94), p. 15–23. ISBN 0-89791-686-7.

DUGAD, R.; DESAI, U. B. **A Tutorial On Hidden Markov Models**. Bombay Powai, Mumbai 400 076, India, May 1996. Signal Processing and Artificial Neural Networks Laboratory Department of Electrical Engineering Indian Institute of Technology.

EAGER, D.; VERNON, M.; ZAHORJAN, J. Minimizing bandwidth requirements for on-demand data delivery. **IEEE Transactions on Knowledge and Data Engineering**, v. 13, n. 5, p. 742–757, Sep 2001. ISSN 1041-4347.

EVERITT, B.; SKRONDAL, A. **The Cambridge dictionary of statistics**. 2002. ISBN 9780521860390.

FENG, H.; CHEN, Z.; LIU, H. Minimizing bandwidth requirements for vod services with client caching. In: **2016 IEEE Global Communications Conference (GLOBECOM)**. 2016. p. 1–7.

_____. Design and optimization for vod services with adaptive multicast and client caching. **IEEE Communications Letters**, PP, n. 99, p. 1–1, 2017. ISSN 1089-7798.

GAO, L.; TOWSLEY, D. Threshold-based multicast for continuous media delivery. **IEEE Transactions on Multimedia**, v. 3, n. 4, p. 405–414, Dec 2001. ISSN 1520-9210.

GOMEZ-URIBE, C. A.; HUNT, N. The netflix recommender system: Algorithms, business value, and innovation. **ACM Trans. Manage. Inf. Syst.**, ACM, New York, NY, USA, v. 6, n. 4, p. 13:1–13:19, December 2015. ISSN 2158-656X.

GUPTA, S.; MOHARIR, S. Request patterns and caching for vod services with recommendation systems. **CoRR**, abs/1609.02391, 2016. Disponível em: <http://arxiv.org/abs/1609.02391>.

HUANG, L. et al. User behavior analysis and video popularity prediction on a large-scale vod system. **ACM Trans. Multimedia Comput. Commun. Appl.**, ACM, New York, NY, USA, v. 14, n. 3s, p. 67:1–67:24, jun. 2018. ISSN 1551-6857. Disponível em: <http://doi.acm.org/10.1145/3226035>.

JAIN, A. K. Data clustering: 50 years beyond k-means. **Pattern Recogn. Lett.**, Elsevier Science Inc., New York, NY, USA, v. 31, n. 8, p. 651–666, jun. 2010. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2009.09.011>.

JAIN, R. **The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling**. Wiley, 1991. ISBN 9780471503361. Disponível em: <https://books.google.com.br/books?id=HetQAAAAMAAJ>.

JAYASUNDARA, C. et al. Improving scalability of vod systems by optimal exploitation of storage and multicast. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 24, n. 3, p. 489–503, March 2014. ISSN 1051-8215.

R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2018. Disponível em: <https://www.R-project.org/>.

RABINER, L.; JUANG, B. An introduction to hidden markov models. **ASSP Magazine, IEEE**, v. 3, n. 1, p. 4 –16, jan. 1986. ISSN 0740-7467.

WANG, B. et al. Optimal proxy cache allocation for efficient streaming media distribution. In: **Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies**. 2002. v. 3, p. 1726–1735 vol.3. ISSN 0743-166X.

YU, H. et al. Understanding user behavior in large-scale video-on-demand systems. **SIGOPS Oper. Syst. Rev.**, ACM, New York, NY, USA, v. 40, n. 4, p. 333–344, abr. 2006. ISSN 0163-5980.