Gerenciamento dinâmico de buffers em players de vídeo IPTV para melhoria do QoE

Marcos Paulo Mendes
Universidade Federal de Juiz de Fora
Rua José Lourenço Kelmer, s/n - Juiz de Fora
Minas Gerais, Brasil
marcospaulo@ice.ufjf.br

Marcelo Ferreira Moreno
Universidade Federal de Juiz de Fora
Rua José Lourenço Kelmer, s/n - Juiz de Fora
Minas Gerais, Brasil
moreno@ice.ufjf.br

ABSTRACT

Due to recent technological evolution, the provision of IPTV services has grown considerably. One of the services normally included in IPTV is Linear TV, where audiovisual contents are made available in the form of program schedules. In this context, challenges emerge regarding the distribution of multimedia content over this serevice. One of these challenges is specifically the user's perception about the quality of the IPTV service, measured in terms of quality of experience (QoE). Thus, this article aims to analyze the problems that may occur when receiving content and to propose solutions to these problems. Specifically, due to the congestion of transmission media or overload in the endpoints, a key issue is the statistical variation of time delay for multimedia content delivery (packet jitter). This paper's proposal comprises a dynamic management of buffers in IPTV terminal devices, taking into account the individual characteristics of Linear TV service, thereby improving the quality of user experience.

Keywords

IPTV; Linear TV; Dynamic Buffer; Jitter; QoE

1. INTRODUÇÃO

É fato que a evolução tecnológica traz diversos benefícios ao cotidiano. Também é fato que esses benefícios tendem a ser cada vez mais exigentes em termos de uso de recursos computacionais. Sendo assim, com a evolução dos meios físicos de transmissão permitindo um tráfego de informações cada vez maior, a possibilidade de serviços convergentes em uma única rede se torna cada vez mais real. Ou seja, um mesmo ponto de conexão poderá ser o responsável pelo acesso à internet, recepção do sinal de TV Digital, voz, video-conferência, dentre outros. Esse conceito é conhecido como *Next Generation Networking* (NGN), e é sobre ele que está o desenvolvimento da tecnologia de transmissão IPTV (*Internet Protocol Television*) e sua arquitetura [8].

A arquitetura IPTV possibilita a transmissão de conteúdo em alta definição, com múltiplos canais de áudio e, junto com o conteúdo, é possível fazer a distribuição de dados, que podem ser aplicações multimídia. É possível também se ter um sistema de correção de erros que podem surgir da existência de ruídos na transmissão [1, 2]. A distribuição de conteúdo IPTV é feita via protocolo IP (*Internet Protocol*) [18]. Uma vez que o áudio e vídeo são comprimidos e codificados, são entregues ao cliente utilizando o protocolo RTP (*Real Time Transport Protocol*) [15] sobre UDP (*User Datagram Protocol*) [13, 8]. Como a distribuição multimídia possui certa tolerância a perda de pacotes, se houver perda de alguns pacotes do conteúdo de áudio e vídeo não será crítico para a reprodução. Além disso, a distribuição multimídia possui um baixo nível de tolerância ao atraso na entrega. Sendo assim, a escolha do UDP é feita pelo fato da redução de sobrecargas para controle (*overhead*) que são geradas na transmissão por TCP (*Transmission Control Protocol*) [14]. Esse *overhead* inclui retransmissão de pacotes, que insere um atraso e o controle de congestionamento que, por sua vez, reduz a taxa de transmissão em caso de congestionamento.

Apesar de haver a garantia de Qualidade de Serviço na entrega de conteúdo IPTV [10], o recebimento deste conteúdo não está livre de atrasos [18], podendo ser desde uma variação na entrega por parte do servidor a um possível atraso no momento de recebimento e exibição do conteúdo por parte do cliente. Estes atrasos se tornam perceptíveis quando a sua existência não é considerada nas implementações de players multimídia [16].

O objetivo deste artigo é encontrar uma solução para o gerenciamento avançado do buffer de conteúdo audiovisual especificamente para serviço IPTV de TV Linear. Neste tipo de serviço, nem todos os parâmetros normalmente utilizados para cálculo de tamanho do *buffer* [4] podem ser obtidos, uma vez que em TV Linear não se pode determinar o comprimento total da reprodução. Além disso, não se pode levar em consideração apenas as características de rede [12] na construção de algoritmos adaptativos para o gerenciamento de *buffer*. Este artigo visa propor uma solução para gerenciamento que seja dinâmico, evitando assim ineficiência de se ter um tamanho de buffer constante para o início da reprodução. A proposta leva em consideração não apenas as características do meio de comunicação, mas também as características dos *endpoints* participantes.

2. FUNDAMENTOS E TRABALHOS RELACIONADOS

Quando um conteúdo chega ao cliente, ele deve ser reproduzido no momento certo. Em geral, esse não será o momento em que o pacote RTP chegou ao cliente, pois os pacotes levarão tempos ligeiramente diferentes para serem

In: Workshop de Trabalhos em Iniciação Científica (WTIC), 13., 2016, Teresina. Anais do XXII Simpósio Brasileiro de Sistemas Multimídia e Web. Porto Alegre: Sociedade Brasileira de Computação, 2016. v. 2.

ISBN: 978-85-7669-332-1

©SBC – Sociedade Brasileira de Computação

transportados na rede. Mesmo que os pacotes sejam despachados exatamente com os intervalos certos entre eles no servidor, eles chegarão ao cliente com tempos relativamente diferentes [16]. Segundo [3], diversos fatores são responsáveis pela degradação da rede como a perda de pacotes, atraso fixo, variação de atrasos, corrupção de pacotes, duplicação de pacotes e reordenação de pacotes.

Segundo Begazo [4], a solução para o problema da variação do atraso é através da utilização de *buffers* na recepção. Essa ideia consiste em armazenar um determinado número de pacotes antes destes serem reproduzidos. Com isto, a variação estatística no atraso, conhecido como *jitter*, pode ser controlado de tal forma a se obter uma reprodução suave, evitando assim interrupções contínuas. Na figura 1 pode se ver como seria o funcionamento desta solução. Os quadrados cinzas numerados são pacotes enviados pelo Servidor ao Cliente. Nota-se que a reprodução só será iniciada após a recepção de 4 pacotes, amenizando ao usuário a percepção do atraso que haveria entre os pacotes 2 e 3 e entre os pacotes 5 e 6. Além disso, nesta solução se tem uma incógnita,

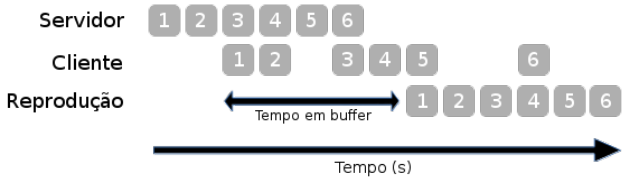

Figura 1: Representação da atuação do buffer em uma reprodução

que é justamente a determinação do tempo em que o cliente irá preencher o *buffer*. Se este valor for constante, o buffer poderá ser efetivo em alguns casos e totalmente ineficiente para outros casos.

Uma vez que o *buffer* consegue tornar transparente ao usuário a ocorrência de atrasos, tem-se um aumento significativo na qualidade de experiência (*Quality of Experience* - QoE), que é um aspecto que tem se tornado cada vez mais importante para o sucesso de serviços e produtos voltados para IPTV [5] e multmídia em geral.

A análise para se determinar o tempo de vídeo em *buffer* pode ser realizada levando em consideração diferentes parâmetros. Como pode ser visto em Bhadada e Sharma [4], parâmetros como a velocidade do meio de transmissão, a taxa de *frames* por segundo e o tamanho total do conteúdo são utilizados para calcular o tempo de *buffer* inicial. Para serviços de TV Linear, fazer uso dessa solução se torna inviável, uma vez que é impossível se determinar o tamanho total do conteúdo, haja visto que a entrega do conteúdo é feita mediante a geração do mesmo, não tendo assim um fim de conteúdo definido. Em Narbutt e Murphy [12], são apresentados algoritmos para o cálculo do atraso decorrente de fatores no meio de transmissão. Assim como em Bhadada e Sharma [4], características decorrentes de ações no cliente não são consideradas, como por exemplo o tempo gasto para demultiplexar o conteúdo. Sendo assim, fatores como o tempo de demultiplexação em um instante no cliente ou, no caso de se realizar análise apenas por características do meio de transmissão, uma pequena sobrecarga no servidor não são consideradas.

Uma vez que o fator de análise é o tempo de duração do conteúdo disponível para reprodução, outros parâmetros como a velocidade do meio, taxa de *frames* por segundo e outros que são utilizados para a realização do cálculo do *buffer* inicial em Bhadada e Sharma [4], são descartados dessa análise, não por serem menos importantes, mas pelo fato de estarem ligados diretamente à duração do conteúdo disponível. Se houver uma diminuição na velocidade de entrega do meio de transmissão (*throughput*), por exemplo, a duração de conteúdo recebida irá reduzir, o que implica diretamente a existência de um possível atraso que pode ser detectado por esta duração de conteúdo recebida. Além disso, o tempo de duração recebido também é afetado diretamente por um atraso por parte do servidor na entrega de um conteúdo em um dado momento, fator este que não é levado em consideração analisando-se apenas os fatores citados anteriormente, como a velocidade do meio. Como o tempo de duração recebido será analisado mediante a demultiplexação de um conteúdo, outro fator que será analisado de forma direta é justamente o atraso por parte do cliente em demultiplexar um conteúdo. Esse fator também não é considerado levando em conta apenas os parâmetros citados anteriormente.

3. GERENCIAMENTO DINÂMICO DO BUFFER

Como se pode observar, a análise pelo tempo de duração do conteúdo pode determinar atrasos até mesmo não previstos que podem afetar diretamente o recebimento de conteúdo do servidor pelo cliente. Como visto, um fator de atraso que pode ocorrer no servidor ou no cliente não pode ser determinado apenas levando em consideração o meio de transmissão, mas também deve levar em consideração o estado comportamental dos dois pontos da comunicação.

Uma vez que é possível obter o PTS e a base de tempo de um *frame* ou amostra de áudio, é possível convertê-lo para uma unidade de tempo como segundos, para fins de análise e controle. Sendo assim, é possível o cálculo de duração de um *frame* ou amostra de áudio, bastando fazer a diferença entre o PTS do sucessor com seu respectivo PTS. Esse resultado é obtido na unidade do fluxo. Sendo assim, multiplicando essa diferença por uma base obtida em como o vídeo foi codificado, obtêm-se este valor em segundos. O cálculo é feito da seguinte forma:

$$d(i) = (P_{i+1} - P_i)B \quad (1)$$

onde $d(i)$ é a duração do i-ésimo *frame* ou amostra de áudio, em segundos, P é o valor do PTS e B é a base de tempo.

Com este cálculo em mãos, a análise passa a ser pela duração em segundos que se tem na recepção dos dados. Sendo assim, para um melhor nível de QoE, é esperado que dentro de um intervalo de tempo t, o valor da soma das durações dos dados recebidos não seja menor que o valor de t, a menos que a diferença entre esses valores possa ser suprida pelo trecho de vídeo presente em *buffer*, representado por T, dado em segundos. Na figura 2 tem-se uma representação gráfica de T.

3.1 Gerenciamento do buffer pelo tempo

Para ser feita a análise, um intervalo de tempo I será definido. Para uma melhor precisão na obtenção do cálculo, este intervalo deve ser mensurado em tempo real (*wall clock*) e deve-se evitar mensurá-lo com contadores de clock do processador, pois o Sistema Operacional hospedeiro da

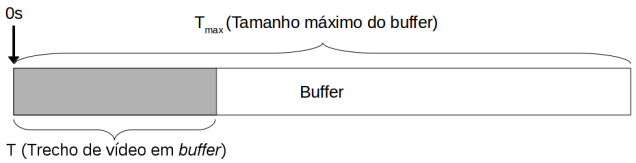

Figura 2: Variáveis e o que representam na gerência do buffer

aplicação pode ser multitarefa, o que implica que o atraso por outro processo estar utilizando o processador também é um atraso a ser considerado.

Considerando que com a função $d(i)$ se obtém a duração de um *frame* ou amostra de áudio, a duração no intervalo de tempo I pode ser calculada somando a duração de cada *frame* recebido no intervalo I. O cálculo é feito da seguinte forma:

$$s_I(j,q) = B\sum_{i=j}^{j+q}(P_{i+1} - P_i) \quad (2)$$

onde j é a quantidade de *frames* recebidos de um único fluxo (áudio ou vídeo) até o início do intervalo I e q é a quantidade recebida durante o intervalo I.

A duração total recebida até o k-ésimo intervalo I é calculada da seguinte forma:

$$s(j) = B\sum_{i=0}^{j}(P_{i+1} - P_i) \quad (3)$$

onde j representa a quantidade de *frames* recebidos.

Como dois pacotes não podem ser entregues ao mesmo tempo e como uma amostra de vídeo pode ter duração distinta da amostra de áudio, então a duração total já recebida s_t de áudio pode ser diferente da duração já recebida de vídeo. Como a análise leva em conta a duração disponível em *buffer*, então o que se tem disponível é a menor duração entre eles, uma vez que a maior duração poderá apenas um dos fluxos possuir. Sendo assim, o cálculo da duração efetiva é feito da seguinte forma:

$$s_e = min(s_{audio}, s_{video}) \quad (4)$$

De posse do valor de s_e, a análise poderá ser feita levando em consideração o total de intervalos I, que é kI, onde k é a quantidade de análises já realizadas. Sendo assim, kI é o tempo total passado desde o início das análises.

A cada intervalo I, a duração de dados recebida neste intervalo deve ser maior ou igual que kI, ou seja, $s_e >= kI$. Se isto for sempre verdade, então não haverá atraso, uma vez que se tem mais duração recebida do que o tempo total em análise. Quando $s_e < kI$, houve atraso na entrega em algum intervalo de kI. Seja $\delta = s_e - kI$, ou seja, a diferença entre a duração total efetiva s_e recebida e o total de intervalos kI. Se $\delta < 0$, então houve atraso em algum momento da análise total. Sendo assim, se $T \neq 0$, então este atraso já pode ter sido analisado e ele já está sendo amenizado pelo *buffer*. Finalmente, define-se Δ, o parâmetro de análise para obtenção de T, como:

$$\Delta = \delta + T \quad (5)$$

que pode também ser escrito como:

$$\Delta = s_e - kI + T \quad (6)$$

Com o valor de Δ, pode-se inferir diretamente um novo valor para T. Se $\Delta < 0$, então tem-se um novo atraso que não foi levado em conta no dimensionamento de T. Neste caso, deve-se ainda comparar o valor de Δ com T. Se $-\Delta <= T$, então o trecho de vídeo em *buffer* tem duração suficiente para cobrir este atraso. Porém, se $-\Delta > T$, então T não consegue cobrir o atraso ocorrido e deve ser atualizado, fazendo com que $T = -\Delta$. De uma forma algébrica, para cada análise realizada no intervalo I, T pode ser obtido da seguinte forma:

$$T = max(T, -\Delta), \forall \Delta < 0 \quad (7)$$

4. RESULTADOS PRELIMINARES

Para os testes realizados, foram utilizados vídeos na resolução 1920x1080, com a codificação do vídeo sendo H.264 e áudio AAC. Além disso, foi utilizado o MPEG-2 TS para a multiplexação dos conteúdos. Todos os vídeos testados possuíam uma incidência de *frame* I (*frame* com imagem completa [6]) a cada 10 *frames*, uma taxa de 30 FPS e 2Mbps de *bitrate* médio. A recepção do conteúdo foi feita através de uma implementação [11] do padrão ITU-T para terminal IPTV [9] e o modelo utilizado para validar a performance da transmissão foi o G.1050 [7]. Além disso, foi utilizado o VLC media player [17] como servidor de *streaming*, simulando a transmissão de um serviço de TV Linear.

Nos gráficos aqui representados, as linhas pontilhadas representam o valor de T no instante de tempo de análise em segundos. As linhas contínuas representam o valor de duração contido no buffer neste mesmo instante de análise. Para a geração destes gráficos, as informações foram colhidas a cada segundo num intervalo de 20 minutos de reprodução (1200 segundos).

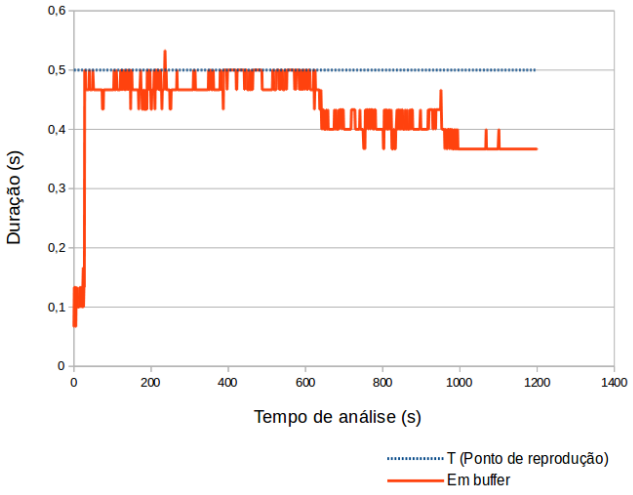

Figura 3: Comparação entre a duração contida no buffer e o T para a transmissão 1

A figura 3 ilustra o comportamento do buffer na reprodução da transmissão 1. A transmissão 1 é caracterizada pelo *streaming* dos vídeos sem a introdução de atrasos simulados. Conforme o gráfico apresentado, houve pequenos atrasos durante toda a transmissão decorrente da decodificação e distribuição do conteúdo por parte do servidor, porém em todos eles o valor de T foi suficiente para amenizá-los.

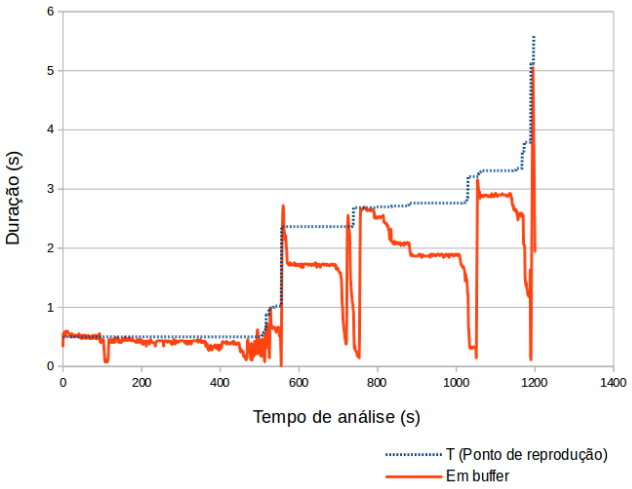

Figura 4: Comparação entre a duração contida no buffer e o T para a transmissão 2

Já a figura 4 ilustra o comportamento do buffer para a transmissão 2, em que atrasos foram simulados. Conforme [7], o *jitter* a ser utilizado deve estar entre 0 e 50 milisegundos. Sendo assim, foi escolhido que a cada 200 pacotes, ocorra um incremento na latência de 50 milisegundos. Para as taxas de perda de pacotes aleatória e de reordenação de pacotes, foi considerado para ambas o valor de 0%, haja visto que a compensação da variação do atraso é o foco deste trabalho, sem considerar perdas. Conforme o gráfico apresentado, pode-se notar os momentos de ocorrência de variações no atraso. Como a latência é incrementada a cada 200 pacotes, em diversos desses momentos o tamanho corrente de T não é suficiente para amenizar o *jitter* e, para se adequar à variação, o tamanho de T é modificado dinamicamente.

5. CONCLUSÃO

A percepção do telespectador com relação ao uso do serviço IPTV é de grande valia. Sendo assim, para serviços IPTV, a qualidade de experiência (QoE) é um desafio na provisão destes serviços.

Neste contexto, o presente artigo buscou propor uma solução a gerência de buffer para players de vídeo IPTV de TV Linear, melhorando a qualidade de experiência. Como esta gerência é dinâmica, ou seja, ocorre de forma automatizada e periódica, o comportamento esperado é a adequação constate do tamanho de trechos de vídeo em *buffer* capaz de amenizar as variações momentâneas de retardo.

Como trabalho futuro, pretende-se realizar a análise e adaptação da solução para serviços IPTV de Vídeo sob Demanda (VoD). Além disso, pretende-se definir limiares, tanto inicial quanto máximo, para o tamanho de trecho de vídeo em *buffer* para ambos serviços IPTV.

6. REFERÊNCIAS

[1] S. D. J. Barbosa and L. F. G. Soares. *Fundamentos, Padrões, Autoria Declarativa e Usabilidade*, chapter 3, pages 105–174. PUC–Rio, Rio de Janeiro, 2008.

[2] C. E. C. F. Batista, M. F. Moreno, and L. F. G. Soares. Recuperação e tolerância a falhas em apresentações de aplicações hipermídia distribuídas. In *XVII Simpósio Brasileiro de Sistemas Multimídia e Web*, Florianópolis, Oct 2011.

[3] D. C. Begazo. Avaliação objetiva e subjetiva de qualidade de vídeo via rede ip com variação de atraso. Dissertação de mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo, oct 2012.

[4] R. Bhadada and K. L. Sharma. Evaluation and analysis of buffer requirements for streamed video data in video on demand applications. *IETE Journal of Research*, 56:242–248, sep 2010.

[5] L. H. Duque. Avaliação da qualidade de vídeo em redes iptv com acesso baseado em adsl. Dissertação de mestrado, Universidade de Brasília, Brasília, jun 2008.

[6] ISO/IEC. Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s - part 2: Video. Standard 11172-2, International Organization for Standardization/International Electrotechnical Commission, 1993.

[7] ITU-T. Network model for evaluating multimedia transmission performance over internet protocol. Recommendation G.1050, International Telecommunication Union, Geneva, nov 2005.

[8] ITU-T. Iptv functional architecture. Recommendation Y.1910, International Telecommunication Union, Geneva, sep 2008.

[9] ITU-T. Iptv terminal devices: Basic model. Recommendation H.721, International Telecommunication Union, Geneva, mar 2009.

[10] ITU-T. Requirements for the support of iptv services. Recommendation Y.1901, International Telecommunication Union, Geneva, jan 2009.

[11] M. F. Moreno, E. Barrere, and C. P. Jr. Serviços iptv voltados à disseminação do conhecimento. In *XXI Simpósio Brasileiro de Sistemas Multimídia e Web*, Manaus, Oct 2015.

[12] M. Narbutt and L. Murphy. Adaptive playout buffering for audio/video transmission over the internet. In *17th IEE UK Teletraffic Symposium*, 2001.

[13] J. Postel. User datagram protocol. STD 6, RFC Editor, August 1980. `http://www.rfc-editor.org/rfc/rfc768.txt`.

[14] J. Postel. Transmission control protocol. STD 7, RFC Editor, September 1981. `http://www.rfc-editor.org/rfc/rfc793.txt`.

[15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications. STD 64, RFC Editor, July 2003. `http://www.rfc-editor.org/rfc/rfc3550.txt`.

[16] A. S. Tanenbaum and D. Wetherall. *Redes de computadores*. Pearson Education do Brasil, 5 edition, 2011.

[17] VideoLAN. Vlc media player. `http://www.videolan.org/vlc/`, 2014.

[18] J. Welch and J. Clark. A proposed media delivery index (mdi). RFC 4445, RFC Editor, April 2006.