

UNIVERSIDADE FEDERAL DO PARANÁ

DANIEL MAURICIO KLASSEN

IMPLEMENTAÇÃO DE UM SUAVIZADOR DE TRÁFEGO PARA MELHORIA DA
QUALIDADE DE EXPERIÊNCIA EM SISTEMAS IPTV

CURITIBA
2013

DANIEL MAURICIO KLASSEN

IMPLEMENTAÇÃO DE UM SUAVIZADOR DE TRÁFEGO PARA MELHORIA DA
QUALIDADE DE EXPERIÊNCIA EM SISTEMAS IPTV

Dissertação apresentada como requisito parcial à
obtenção do grau de Bacharel em Engenharia
Elétrica, no Curso de Engenharia Elétrica, Setor
de Tecnologia, Universidade Federal do Paraná.

Orientador: Prof. Dr. Carlos Marcelo Pedroso

CURITIBA
2013

TERMO DE APROVAÇÃO

DANIEL MAURICIO KLASSEN

IMPLEMENTAÇÃO DE UM SUAVIZADOR DE TRÁFEGO PARA MELHORIA DA QUALIDADE DE EXPERIÊNCIA EM SISTEMAS IPTV

Dissertação aprovada como requisito parcial para a obtenção do grau de Bacharel em Engenharia Elétrica, o Curso de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, pela seguinte banca examinadora:

Prof. Dr. Carlos Marcelo Pedroso
Orientador – Departamento de Engenharia Elétrica, UFPR

Prof. Dr. Evelio Martín García Fernández
Departamento de Engenharia Elétrica, UFPR

Prof. Dr. Luis Henrique Assumpção Lolis
Departamento de Engenharia Elétrica, UFPR

Curitiba, 22 de Março de 2013

AGRADECIMENTOS

A Deus:

Não a nós, Senhor, nenhuma glória para nós, mas sim ao Teu nome, por Teu amor e por Tua fidelidade.

Salmos 115:1

A minha esposa Gabrielle:

Pelo amor e dedicação a mim investidos, um aprendizado diário com direita a muita sabedoria. Amor da minha vida!

A minha Família:

Valores, princípios que perduram, o investimento na educação trazem retorno com esse trabalho.

Ao Professor Pedroso:

Por acreditar na minha competência de concluir com êxito o trabalho, pelo conhecimento adquirido nessa jornada, trazendo esse trabalho como resultado.

"Sou um só, mas sou um. Não posso realizar tudo, mas posso fazer algo. E por ser incapaz de tudo, não vou recusar o pouco que posso fazer."

Edward Hale

RESUMO

O avanço do domínio das técnicas de tráfego de informações possibilita o uso de aplicações que contenham transmissão de vídeo instantâneo. Uma dessas aplicações é a televisão pela rede IP (*Internet Protocol*), o IPTV (*Internet Protocol Television*). A sua ampla utilização pode causar congestionamento na rede, havendo a necessidade de recorrer a técnicas de otimização dos recursos da rede. O sistema IPTV normalmente utiliza o padrão MPEG para codificação do vídeo, que utiliza a redundância espacial e temporal para reduzir a quantidade de informação para representar a sequência de imagens. A compressão resultante reduz um pouco a qualidade da imagem, e altera a necessidade de recursos de rede, que de um fluxo de taxa constante passa a ter uma característica de taxa variável. Com o fluxo agregado dos sinais de IPTV, picos de tráfego podem sobrecarregar os *buffers* dos roteadores. A sobrecarga gera descarte de pacotes e atraso variável, ocasionando perda do conteúdo da transmissão. No entanto o fluxo de vídeo codificado pode ser suavizado. São sugeridos três métodos suavizadores: o Balde de Fichas e duas variantes, o SRTCM (*Single Rate Three Color Marker*) e o TRTCM (*Two Rate Three Color Marker*). A percepção de qualidade por parte do usuário é chamada Qualidade de Experiência (*Quality of Experience*). Uma das medidas mais utilizadas para avaliar a QoE é o MOS (Mean Opinion Score), que avalia e qualifica o grau de satisfação do usuário em relação a qualidade do sinal de vídeo recebido. O valor do MOS pode ser estimado de forma objetiva, através da ferramenta EvalVid, que o calcula através do PSNR (*Peak Signal-to-Noise Ratio*) do vídeo recebido em relação ao transmitido.

Palavras-chave: Sistemas Multimídia, IPTV, MPEG, Suavização de Tráfego, Qualidade de Experiência.

ABSTRACT

The advancement of the field of traffic information techniques enables the use of video streaming. One of these applications is the television over IP (IPTV - Internet Protocol Television). Its use can lead to network congestion, requiring the use of network optimization techniques. The IPTV system normally uses the MPEG standard for video coding, which uses the spatial and temporal redundancy to reduce the amount of information necessary to represent the images. The compression reduces the image quality, and changes the need for network resources, requiring the transmission of a variable bit rate flow. The bursts of video traffic can overload the buffers of the routers. The overload generates packet discard and variable end-to-end delay, causing packet loss and degrading the viewer perceived quality. However, the encoded video stream can be smoothed. Three methods are suggested: the Token Bucket and two of its variants, SRTCM (Single Rate Three Color Marker) and TRTCM (Two Rate Three Color Marker). The perception of quality from the user is called Quality of Experience (QoE). One of the most widely used measures to evaluate QoE is the MOS (Mean Opinion Score), which evaluates and describes the degree of user satisfaction regarding the quality of the incoming video signal. The MOS value can be estimated objectively by EvalVid tool which calculates by PSNR (Peak Signal-to-Noise Ratio) of the received video.

Key-words: Multimedia Systems, IPTV, MPEG, Traffic Smoothing, Quality of Experience

LISTA DE FIGURAS

FIGURA 1 - SISTEMA DE TRANSMISSÃO DE TELEVISÃO ABERTA.....	20
FIGURA 2 - TIPOS DE CONEXÃO DE TELEVISÃO POR ASSINATURA. SÃO DEMONSTRADOS TRÊS TIPOS DE CONEXÃO: VIA SATÉLITE, VIA ERB E VIA CABO.....	21
FIGURA 3 - ANÁLISE DE REGIÕES HOMOGÊNEAS EM UMA IMAGEM FONTE: IAIN E. G. RICHARDSON	23
FIGURA 4 - REDUNDÂNCIA ESPACIAL E TEMPORAL EM UM VÍDEO FONTE: IAIN E. G. RICHARDSON	24
FIGURA 5 - GOP SINALIZANDO O QUADRO I E OS QUADROS P E B DEPENDENTES FONTE: MICHAEL ISNARDI, 1999	25
FIGURA 6 - GOP COM SINALIZAÇÃO DAS IMAGENS P E AS SUAS PREDIÇÕES FUTURAS FONTE: MICHAEL ISNARDI, 1999	26
FIGURA 7 - GOP SINALIZANDO AS IMAGENS B COM PREDIÇÕES BIDIRECIONAIS FONTE: MICHAEL ISNARDI, 1999	26
FIGURA 8 - GOP COM OS CONJUNTOS DE IMAGENS COM PREDIÇÕES FUTURAS E BIDIRECIONAIS. FONTE: MICHAEL ISNARDI, 1999	27
FIGURA 9 - QOE DEVE SER GARANTIDA ENTRE O USUÁRIO E APLICAÇÃO E QOS DEVE SER GARANTIDA ENTRE A APLICAÇÃO E A REDE	27
FIGURA 10 - MÉTODO DE SUAVIZAÇÃO SRTCM, EXISTE UMA ENTRADA DE FICHAS, AS FICHAS DE TRASBORDO PASSAM PARA O SEGUNDO BALDE. EXISTEM TRÊS ESTADOS POSSÍVEIS DE SAÍDA.....	33
FIGURA 11 - MÉTODO DE SUAVIZAÇÃO TRTCM, EXISTE DUAS ENTRADAS DE FICHAS E TRÊS ESTADOS POSSÍVEIS DE SAÍDA.....	34
FIGURA 12 - FLUXOGRAMA DE FUNÇÕES PARA A EXECUÇÃO DO EVALVID FONTE: JIRKA KLAUE, 2003	37
FIGURA 13 - ARQUIVO DE TRACE GERADO, PELO VIDEO SENDER.	38
FIGURA 14 - ARQUIVO DE TRACE GERADO COM O TEMPO DE CADA PACOTE, A SUA IDENTIFICAÇÃO E O TAMANHO DO PACOTE UDP.....	38
FIGURA 15 - AVALIAÇÃO REALIZADA PELO ET. CÁLCULO DE PERDA DE PACOTES E DE QUADROS, SENDO DEMONSTRADO O TAMANHO TOTAL, O PERDIDO E A PERDA RESPECTIVA EM PORCENTAGEM. O	

QUADRO H É O QUADRO DE INÍCIO DE TRANSMISSÃO A SUA COMPACTAÇÃO É EQUIVALENTE AO QUADRO I.	39
FIGURA 16 - VALORES CALCULADOS EM SIMULAÇÃO COM RESULTADO DE BAIXA PERDA E ALTAS PERDAS E SUAS INFLUÊNCIAS NO RESULTADO DO PSNR.	40
FIGURA 17 – TOPOLOGIA SIMPLIFICADA DO SISTEMA DE REDE DO IPTV	42
FIGURA 18 - TOPOLOGIA CRIADA PARA A SIMULAÇÃO DE TRANSMISSÃO DE IPTV NO SOFTWARE NS2	42
FIGURA 19 - MOS CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A Φ_1	46
FIGURA 20 - MOS CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A $\Phi_1/2$...	47
FIGURA 21 - <i>DELAY</i> CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A Φ_1	48
FIGURA 22 - <i>DELAY</i> CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A $\Phi_1/2$.49	
FIGURA 23 - <i>JITTER</i> CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A Φ_1 ...	49
FIGURA 24 - <i>JITTER</i> CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A $\Phi_1/2$ 50	
FIGURA 25 - MOS CÁLCULADO COM <i>BUFFER</i> INFINITO E BALDE IGUAL A Φ_1 .50	
FIGURA 26 - MOS CÁLCULADO COM <i>BUFFER</i> INFINITO E BALDE IGUAL A $\Phi_1/2$	51
FIGURA 27 - <i>DELAY</i> CÁLCULADO COM <i>BUFFER</i> INFINITO E BALDE IGUAL A Φ_1	52
FIGURA 28 - <i>DELAY</i> CÁLCULADO COM <i>BUFFER</i> INFINITO E BALDE IGUAL A $\Phi_1/2$	53
FIGURA 29 - <i>JITTER</i> CÁLCULADO COM <i>BUFFER</i> INFINITO E BALDE IGUAL A Φ_1	54
FIGURA 30 - <i>JITTER</i> CÁLCULADO COM <i>BUFFER</i> INFINITO E BALDE IGUAL A $\Phi_1/2$	54

LISTA DE TABELAS

TABELA 1 - TABELA DE AVALIAÇÃO MOS	30
TABELA 2 - TABELA DE AVALIAÇÃO MOS ATRAVÉS DO PSNR	36
TABELA 3 - VALORES DE PARÂMETROS OBTIDOS A PARTIR DO TRACE DO VIDEO SIMULADO	41
TABELA 4 - VALORES DE OCUPAÇÃO DE CANAL UTILIZADOS NA SIMULAÇÃO.	43
TABELA 5 - CÁLCULO DO TAMANHO DO BALDE. MÉTODO BALDE E SRTCM E TRTCM. NO SRTCM E TRTCM SÃO UTILIZADOS DOIS BALDES IGUAIS, A SUA SOMA É EQUIVALENTE AO TAMANHO DO BALDE DE FICHAS. TAMANHO MÁXIMO LIMITADO PELA EXPRESSÃO 9	45
TABELA 6 - RESULTADO DO MOS A PARTIR DA SIMULAÇÃO COM 1% DE PERDA.....	46
TABELA 7 - RESULTADO DO <i>DELAY</i> A PARTIR DA SIMULAÇÃO COM 1% DE PERDA.....	48
TABELA 8 - RESULTADO DO <i>JITTER</i> A PARTIR DA SIMULAÇÃO COM 1% DE PERDA.....	49
TABELA 9 - RESULTADO DO MOS A PARTIR DA SIMULAÇÃO COM <i>BUFFER</i> INFINITO.....	51
TABELA 10 - RESULTADO DO <i>DELAY</i> A PARTIR DA SIMULAÇÃO COM <i>BUFFER</i> INFINITO.....	52
TABELA 11 - RESULTADO DO <i>JITTER</i> A PARTIR DA SIMULAÇÃO COM <i>BUFFER</i> INFINITO.....	53

LISTA DE SIGLAS

IP	-	<i>Internet Protocol</i>
IPTV	-	<i>Internet Protocol Television</i>
QoS	-	<i>Quality of Service</i>
QoE	-	<i>Quality of Experience</i>
FPS	-	<i>Frames per Second</i>
MPEG	-	<i>Moving Picture Experts Group</i>
I	-	<i>Intra Coded Picture</i>
P	-	<i>Predictive Coded Picture</i>
B	-	<i>Bidirectionally Predictive Coded Picture</i>
VBR	-	<i>Variable Bit Rate</i>
SLS	-	<i>Service Level Specification</i>
SRTCM	-	<i>Single Rate Three Color Marker</i>
TRTCM	-	<i>Two Rate Three Color Marker</i>
MOS	-	<i>Mean Opinion Score</i>
ITU	-	<i>International Telecommunications Union</i>
PSNR	-	<i>Peak Signal-to-Noise Ratio</i>
ERB	-	<i>Estação Rádio Base</i>
HG	-	<i>Home Gateway</i>
STB	-	<i>Set Top Box</i>
UDP	-	<i>User Datagram Protocol</i>
RFC	-	<i>Request for Comments</i>
IETF	-	<i>Internet Engineering Task Force</i>
RTP	-	<i>Real-time Transport Protocol</i>
DVD	-	<i>Digital Versatile Disk</i>
RGB	-	<i>Red, Green & Blue</i>
ISO	-	<i>International Organisation for Standardisation</i>
CD	-	<i>Compact Disc</i>
GOP	-	<i>Group of Pictures</i>
VOIP	-	<i>Voice Over Internet Protocol</i>
CIR	-	<i>Committed Information Rate</i>
CBS	-	<i>Committed Burst Size</i>

EBS	-	<i>Excess Burst Size</i>
PIR	-	<i>Peak Information Rate</i>
PBS	-	<i>Peak Burst Size</i>
NS	-	<i>Network Simulator</i>
OTCL	-	<i>Object-oriented Tool Command Language</i>
MIT	-	<i>Massachusetts Institute of Technology</i>
SNR	-	<i>Signal Ratio Noise</i>
MSE	-	<i>Mean Square Error</i>

SUMÁRIO

1 INTRODUÇÃO	17
1.1 OBJETIVO.....	19
1.2 METODOLOGIA.....	19
2 SISTEMA IPTV	19
2.1 MPEG.....	22
2.1.1 Compressão.....	22
2.1.2 FUNCIONAMENTO DO MPEG.....	24
2.1.1 QoS	28
2.1.2 QoE	28
2.3 MOS	29
3 SUAVIZAÇÃO DE TRÁFEGO	30
3.1 MÉTODOS CONTROLADORES DE TRÁFEGO	30
3.1.1 Balde Furado.....	30
3.1.2 Balde de Fichas.....	31
3.1.3 SRTCM (<i>Single Rate Three Color Marker</i>).....	32
3.1.4 TRTCM (<i>Two Rate Three Color Marker</i>)	33
4 TECNOLOGIAS UTILIZADAS	34
4.1 NS2 – NETWORK SIMULATOR 2	34
4.2 EVALVID	35
4.2.1 VS – <i>Video Sender</i>	37
4.2.3 FV – <i>Fix Video</i>	39
4.2.4 PSNR	39
4.2.5 MOS	39
4.3 TÉCNICAS DE SUAVIZAÇÃO	40
4.4 ORIGEM DOS DADOS	40

5 SIMULAÇÃO	41
5.1 PARÂMENTROS DE SIMULAÇÃO.....	41
6 RESULTADOS OBTIDOS	45
6.1 PERDA IGUAL A 1%.....	46
6.2 <i>BUFFER</i> INFINITO.....	51
7 DISCUSSÃO	55
7 CONCLUSÕES E TRABALHOS FUTUROS	56
REFERÊNCIAS	57
APÊNDICE	58

1 INTRODUÇÃO

O avanço dos sistemas de telecomunicações permite o suporte de diferentes tipos de transmissões em tempo real. A transmissão do vídeo assume um papel muito importante no meio das aplicações da *Internet*. Uma dessas aplicações de vídeo é a transmissão de televisão utilizando o protocolo IP (*Internet Protocol*), o IPTV (*Internet Protocol Television*). Uma rede para transmissão de sistemas IPTV deve fornecer níveis apropriados de qualidade de serviço (QoS – *Quality of Service*) e qualidade de experiência de usuário (QoE – *Quality of Experience*) (ITU-T SG12). Por se tratar de uma transmissão em tempo real de vídeo com alta qualidade, ela exige da rede a transferência de um grande volume de tráfego. Um vídeo é composto por um conjunto de imagens, que são denominados de quadros, transmitidos de um servidor para o cliente a certa taxa, dada em quadros por segundo (fps – *frames per second*). Cada quadro é composto por uma imagem com tamanho igual da resolução do vídeo. Quando o quadro é trocado, realiza-se a troca de uma imagem inteira por outra. No entanto, existe uma grande correlação entre uma imagem com as imagens anteriores, de forma que, as mudanças dos conteúdos, entre cada quadro, podem ser pequenas – um vídeo apresenta dois níveis de redundância: espacial e temporal. Essa redundância permite compactar o vídeo, permitindo uma redução nas taxas de transmissão. O MPEG (*Moving Picture Experts Group*) é uma técnica de compressão que utiliza a redundância espacial dentro da mesma imagem e temporal entre imagens subsequentes. Esse método consiste em utilizar um mecanismo de correlação entre os blocos correspondentes nas imagens atuais e nas imagens futuras. Nesse processo são utilizados três tipos diferentes de quadros: I, P e B. O quadro I (*Intra coded picture*) é um quadro codificado sem referência de outros quadros. O quadro P (*Predictive coded picture*) utiliza codificação preditiva de compensação referente à de um quadro I ou P anterior. E por fim, o quadro B (*Bidirectionally predictive coded picture*) é codificado a partir da interpolação entre um quadro anterior (I ou P) e um quadro posterior I ou P. Com a compactação, o novo vídeo transmitido possui uma característica de taxa de transmissão variável (VBR - *Variable Bit Rate*), pois o tamanho dos quadros codificados depende fundamentalmente da informação que ele carrega. Do ponto de vista do planejamento de recursos, essa variação de taxa de transmissão na rede é

prejudicial, pois dificulta o planejamento de capacidade e os mecanismos de monitoramento das condições exigidas para obter nos níveis desejados de QoE/QoS, de acordo com a especificação do nível de serviço necessário (SLS – *Service Level Specification*). Mesmo com sistemas operando com fatores de utilização relativamente baixos, pode ocorrer perda de pacotes devido à presença de rajadas de tráfego. A perda de pacotes é provocada quando ocorre a coincidência de múltiplos picos no tráfego agregado e não há espaço disponível no *buffer* do roteador. Para reduzir essa perda de pacotes, é sugerida a implementação de uma suavização de tráfego do fluxo IPTV na origem, limitando a variabilidade do tráfego de vídeo. Os efeitos esperados são os seguintes: (i) redução da perda de pacotes (devido ao atraso máximo fim-a-fim e ao descarte nas filas) no tráfego agregado, com consequente melhoria na QoE e (ii) piora no tempo de troca de canal, em virtude do maior tamanho do *buffer anti-jitter* exigido no receptor para acomodar o atraso inserido pela suavização. No trabalho serão utilizados três tipos de suavização, Balde Furado e duas de suas derivações, a SRTCM (*Single Rate Three Color Marker*) e a TRTCM (*Two Rate Three Color Marker*).

Para analisar a QoE existe o teste do MOS (*Mean Opinion Score*). O MOS é uma avaliação qualitativa de QoE onde a nota máxima é o 5 (Excelente) e a mínima é o 1 (Ruim). As avaliações de vídeos através do método MOS exigem uma série de regras, como ambiente com configurações definidas pela ITU¹ (*International Telecommunications Union*) e um número de 4 a 40 de usuários, para cada teste, para que o levantamento estatístico seja aceitável. A título de simulações isso se torna inviável, pois são feitos muitos testes, geralmente em curto espaço de tempo, e com constante mudança em diversos parâmetros, o que eleva muito o número de avaliações necessárias. Para facilitar esse levantamento de dados e seus resultados a Universidade Técnica de Berlim desenvolveu um *software* chamado EvalVid (Klaue, 2003) que permite calcular os resultados do MOS a partir da comparação do PSNR (*Peak Signal-to-Noise Ratio*) entre o vídeo original e o vídeo efetivamente recebido pelo cliente.

¹ Norma para avaliação do MOS: ITU-R BT.500-13 **Methodology for the subjective assessment of the quality of television pictures**, 2012

1.1 OBJETIVO

Com a suavização do tráfego na transmissão de IPTV, espera-se obter uma melhoria da qualidade de experiência. Espera-se reduzir perdas de pacotes, além da perda menor será estudado o *delay* e o *jitter* resultantes na transmissão e as suas consequências. O *delay* é o atraso fim-a-fim da transmissão do vídeo e o *jitter* é a variação de atraso entre os pacotes transmitidos. Com os a estimativa do MOS para cada método de suavização em estudo, será possível avaliar as vantagens do uso desta abordagem e também analisar qual método testado é o mais eficiente.

1.2 METODOLOGIA

Será avaliado o MOS resultante da transmissão de vídeo suavizado em comparação com o MOS resultante da transmissão do mesmo vídeo sem a aplicação da suavização. Serão utilizados vídeos normalmente empregados em testes de sistemas multimídia. Para realização dos testes será empregado o simulador NS2, utilizado para simulação da transmissão de vídeos em uma configuração que imita as condições operacionais de um sistema real. O ambiente de simulação abrangerá ambientes com diversos níveis de utilização do canal de transmissão. O trabalho contempla, inicialmente, conceitos sobre o sistema IPTV, o codec MPEG e os métodos de suavização. Após, são apresentados os resultados das simulações e conclusões.

2 SISTEMA IPTV

A *Internet* é a rede mundial de computadores, segundo um levantamento realizado, existem aproximadamente 2,4 bilhões de usuários e 634 milhões de *sites* cadastrados (Pigdom, 2013). A *Internet* é a rede de que interliga outras redes de computadores, o que permite a conexão entre computadores distantes fisicamente. O avanço e o domínio das tecnologias da transmissão e de processamento de dados possibilitaram a utilização de aplicações que antes eram impraticáveis. Uma dessas aplicações é o IPTV. Um sistema IPTV pode ser simplificado em três

componentes: os servidores, a rede IP e os clientes. Para que a transferência de dados ocorra, o cliente solicita ao servidor o acesso ao conteúdo. O servidor valida e autentica este cliente e envia o fluxo de vídeo solicitado utilizando a rede IP.

Atualmente o serviço de televisão é fornecido de duas formas, o sinal aberto e o sinal pago. O sistema IPTV é utilizado para o fornecimento de televisão sobre uma rede já existente, podendo ser definida como um novo método de transmissão de televisão paga.

No sistema de televisão aberta a emissora de televisão transmite o seu sinal através de uma ERB (Estação Rádio Base) livremente. Todos os usuários que tiverem um meio adequando de recepção da transmissão terão acesso ao conteúdo. A Figura 1 ilustra a estrutura de rede de um sistema de transmissão de televisão aberta. No sistema de televisão paga, o sinal pode ser transmitido através de ERB, sistema de satélite ou através de conexão física entre o servidor e o cliente. Para que cliente tenha acesso ao conteúdo será necessário ele possuir um *set top box*, equipamento que possui a chave de acesso e demodula o sinal recebido para a

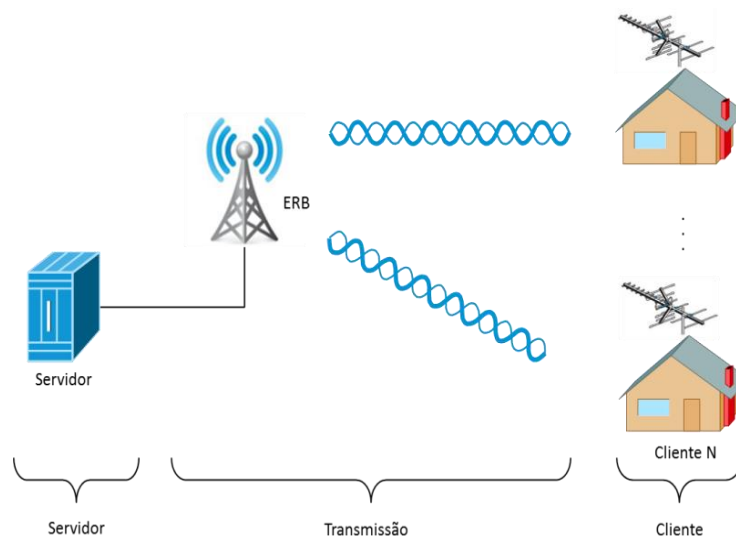


FIGURA 1 - SISTEMA DE TRANSMISSÃO DE TELEVISÃO ABERTA

exibição na televisão. Dentre os três sistemas, somente o da conexão física permite a interação do cliente com o conteúdo fornecido. Nos outros sistemas a comunicação é normalmente unilateral e o cliente somente decodifica o sinal. A Figura 2 ilustra os diversos elementos do sistema de televisão paga.

Uma operadora pode transmitir o sinal de IPTV utilizando a rede IP existente. O cliente recebe o sinal de televisão pela rede de transmissão da operadora, o que possibilita a operadora acrescentar diversas aplicações de interatividade nas

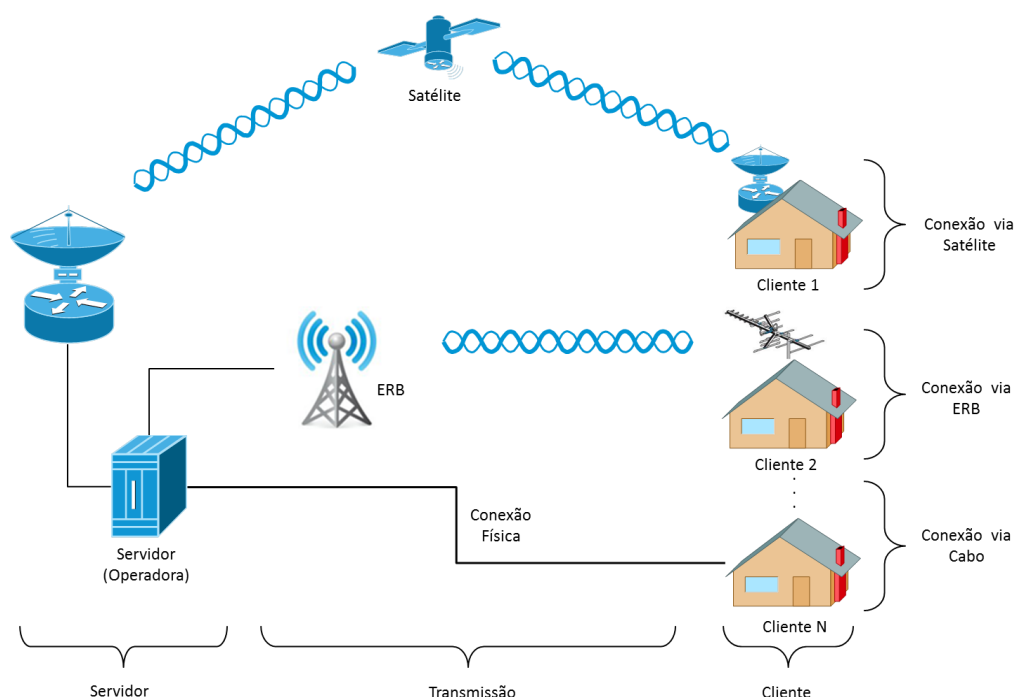


FIGURA 2 - TIPOS DE CONEXÃO DE TELEVISÃO POR ASSINATURA. SÃO DEMONSTRADOS TRÊS TIPOS DE CONEXÃO: VIA SATÉLITE, VIA ERB E VIA CABO.

programações de televisão. Para que o cliente possa assistir a sua televisão pelo IPTV é necessário que ele possua um *Home Gateway* (HG) em sua casa. O HG recebe o sinal e separa os serviços. O sinal de vídeo é transmitido para o *Set Top Box* (STB), que por sua vez transmite o sinal para a televisão. O STB é a interface do usuário para o serviço de televisão. Quando a conexão é estabelecida entre o cliente e o servidor, o cliente solicita o recebimento de um conteúdo de um canal. O servidor autentica o usuário e envia o sinal através de pacotes IP através da rede.

Dentro da rede IP os roteadores possuem o trabalho de encaminhar os pacotes para o local correto. O transporte dos pacotes realizado pelo protocolo UDP (*User Datagram Protocol*) definido na RFC.768 (*Request for Comments* número 768) (Postel, 1980) do IETF (*Internet Engineering Task Force*). O UDP cria um datagrama no pacote IP, esse datagrama consiste em um cabeçalho com as informações necessárias para o roteamento da origem ao destino do pacote, além de um campo com o tipo do pacote. No UDP a entrega do pacote IP, a hora de chegada ou a ordem não é garantida, caso essas garantias sejam necessárias, deve-se programar estruturas de controle. O UDP é um serviço sem conexão. Para aplicações como o IPTV é utilizado um protocolo de transporte em tempo real, o RTP (*Real-time*

Transport Protocol). Nesse protocolo, definido pela RFC 3550, é descrito o procedimento da fragmentação do fluxo transmitido, a sequência da informação e o tempo de referência para reprodução. O RTP utiliza o UDP como protocolo de transporte.

Estima-se que atualmente são assistidos aproximadamente quatro bilhões de horas de vídeo no *site* YouTube (Pingdom, 2013) e no *site* VIMEO existem mais de 200 *petabytes* de vídeos nos servidores. Estes números mostram o crescimento da quantidade de clientes conectados e o aumento da demanda solicitada da transmissão de vídeos. Já no caso do IPTV a situação se agrava, pois a sua transmissão envolve conteúdo com limite para atraso máximo de transmissão. O fluxo agregado de vídeo e picos de tráfego que ocorrem em do *codec* utilizado podem acarretar em perdas de pacotes. Existem duas alternativas para que se possa resolver o problema causado pela alta variabilidade do tráfego de vídeo: aumentar a capacidade dos equipamentos ou utilizar técnicas de suavização de tráfego de forma a limitar as rajadas geradas pelo *codec* MPEG, com possível redução de perdas, atraso médio e a variação de atraso, o que beneficia a QoE.

2.1 MPEG

O MPEG foi criado pelo ISO (*International Organization for Standardization*) para definir padrões para a compressão e transmissão de áudio e vídeo. Sua primeira versão foi lançada em 1988. O MPEG-4 parte 10 (ISO, 2010) está especificado na norma ISO/IEC 14496.

2.1.1 Compressão

Para transmitir o vídeo sem compactação (*cru*) em tempo real seria necessária uma taxa muito alta de transmissão. Por exemplo, um DVD (*Digital Versatile Disk*) comum suportaria apenas poucos segundos do vídeo em seu formato *cru* e com resolução e taxa de quadros utilizados no sistema televisivo de alta definição (Richardson, 2003).

Em cada quadro as cores são representadas por três números que são as proporções das três cores primárias aditivas da luz, o vermelho, o verde e o azul (RGB – *Red, Green and Blue*). Essas cores são chamadas de primárias, pois com a proporção certa elas podem se misturar e compor qualquer outra cor. O sistema

visual humano é mais sensível ao brilho (luminância) do que a cores (Richardson, 2003). Nas cores RGB cada uma das três cores possui a mesma importância, mas é possível representar uma cor mais eficientemente, separando a luminância da informação da cor. Uma maneira eficiente de representar as cores e o brilho de um vídeo é o $YCbCr$ (muitas vezes referenciado como formato YUV). Y define o grau de luminância do vídeo, Cb é a diferença entre as cores referenciando o azul e Cr é a diferença das cores, referenciando o vermelho.



FIGURA 3 - ANÁLISE DE REGIÕES HOMOGÊNEAS EM UMA IMAGEM
FONTE: IAIN E. G. RICHARDSON

O vídeo é composto por uma sequência de imagens. A taxa do vídeo é definida pelo número de imagens em um segundo, nomeada de quadros por segundo (fps). A compressão é realizada por meio da análise de redundâncias na informação entre os quadros. Segundo Richardson (2003), o algoritmo que opera na compressão de vídeo remove a redundância temporal, espacial e/ou no domínio da frequência. Por exemplo, em uma filmagem existem regiões que são estáticas entre quadros e/ou estáticas referentes a *pixels* associados. Na Figura 3 existem duas regiões marcadas, onde demonstram regiões homogêneas, ou seja, entre um quadro e outro, ela permanece inalterada. Essa região homogênea pode ser suprimida em alguns quadros, desde que no quadro de referência ela esteja lá e indique em quais quadros futuros ele deve estar também.

A codificação de vídeo é o procedimento de compressão e descompressão de um sinal de vídeo digital (Richardson, 2003). Para que a codificação seja possível é necessário criar parâmetros de análise. Em uma sequência capturada do mundo real é composta de múltiplos objetos, cada qual com a sua forma característica, profundidade, textura e iluminação. As cores e o brilho de um vídeo de uma cena natural contêm infinitas formas e constantes mudanças. A compressão espacial analisa características como a variação de textura dentro de uma imagem ou *frame*, a quantidade de objetos, seus formatos e cores. A compressão temporal verifica as alterações da imagem em quadros sucessivos, a movimentação dos objetos, as mudanças de iluminação, movimentação da câmera em uma cena, bem como, os pontos distintos de vista. Na Figura 4 ilustra a redundância temporal e espacial em uma sequência de quadros.

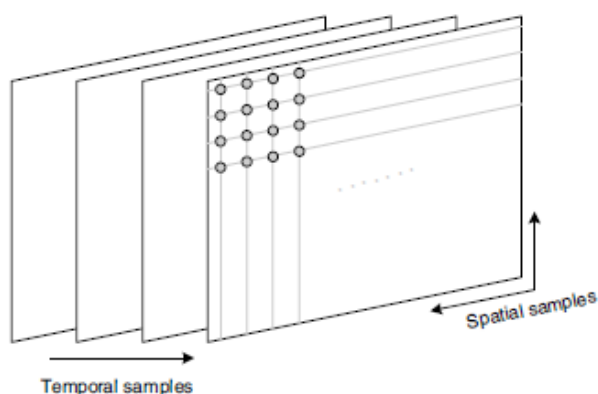


FIGURA 4 - REDUNDÂNCIA ESPACIAL E TEMPORAL EM UM VÍDEO
 FONTE: IAIN E. G. RICHARDSON

2.1.2 Funcionamento do MPEG

O MPEG é um padrão de representação visual (Richardson, 2003). Cada padrão é um documento que primordialmente define a representação codificada da informação visual, o método de decodificação e a sintaxe de reconstrução visual.

O MPEG é um grupo pertencente ao ISO e desenvolveu as diversas séries de padrões de vídeos também definidas de MPEG. O primeiro padrão criado, o MPEG-1, foi utilizado para compressão de vídeo e áudio para CD's (*Compact Disc*). A sua segunda versão, o MPEG-2, fez muito sucesso, ele foi utilizado para armazenamento e transmissão em *broadcast* de áudio e vídeo em para transmissões com qualidade para televisão. Dentro dos sistemas do MPEG-2, uma importante aplicação foi descoberta e é amplamente utilizada em áudios, o MPEG

Layer 3, ou mp3. O MPEG-4 é uma versão mais atual que trabalha somente com codificação de áudio e vídeo e será utilizada neste trabalho.

Para a construção das camadas do MPEG existem sintaxes importantes a serem analisadas para cada camada. Nesses parâmetros é importante o tamanho da imagem, o taxa de quadros, a taxa de *bits* e o tamanho de *buffers*. A estrutura do MPEG contém três tipos diferentes de quadros, I, P e B. Um arranjo regrado dessas imagens é denominado de GOP (*Group of Pictures*). Quanto maior o GOP, pior a qualidade do vídeo, menor a taxa de transmissão necessária e pior o atraso para ações interativas do usuário, como por exemplo, a troca de canal em um sistema IPTV. (Isnardi, 1999)

Os quadros I são codificados sem referência de outras imagens. A compressão é moderada, pois no quadro I são apenas retiradas as redundâncias espaciais, redundâncias temporais não são suprimidas. O quadro I é fundamental no sistema MPEG, pois ele serve de referência para todos os quadros pertencentes ao seu GOP.

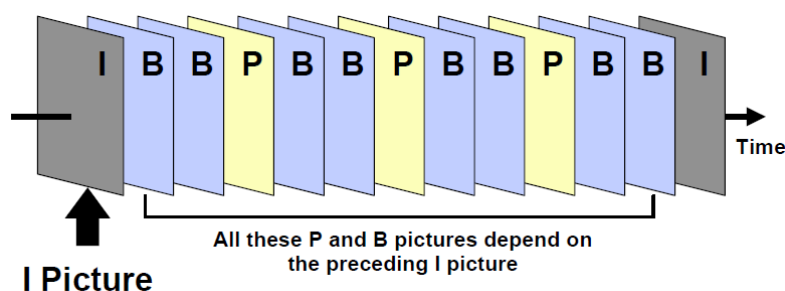


FIGURA 5 - GOP SINALIZANDO O QUADRO I E OS QUADROS P E B DEPENDENTES
 FONTE: MICHAEL ISNARDI, 1999

Os quadros P são comprimidos temporal e espacialmente. Essa compressão é realizada a partir da predição por compensação de movimento a partir de uma imagem I ou P passada. O quadro P se referencia a partir de um quadro P ou I passado e serve de referência para um quadro P futuro. Um quadro P típico é três vezes menor que um quadro I. Em relação ao quadro B, o quadro P é geralmente 50% maior (Isnardi, 1999).

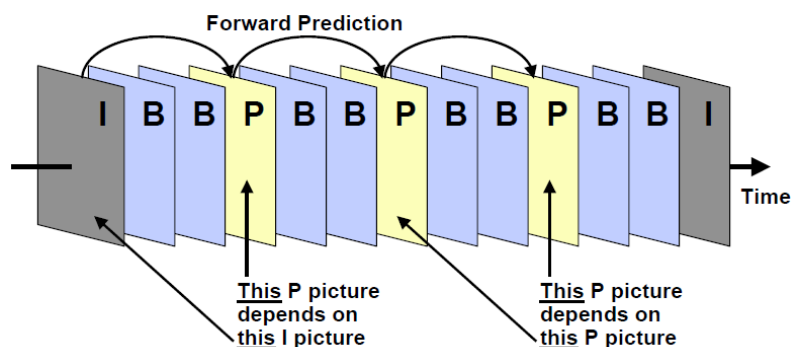


FIGURA 6 - GOP COM SINALIZAÇÃO DAS IMAGENS P E AS SUAS PREDIÇÕES FUTURAS
 FONTE: MICHAEL ISNARDI, 1999

O Quadro B é o tipo de quadro que recebe a maior compressão. A sua referência é bidirecional, pois ela se baseia em dois quadros (I ou P) que estejam um no passado e um no futuro. Nesse quadro são utilizadas redundâncias espaciais e temporais.

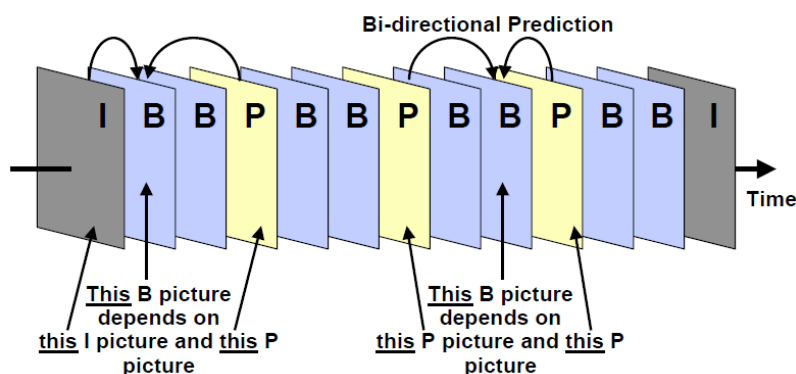


FIGURA 7 - GOP SINALIZANDO AS IMAGENS B COM PREDIÇÕES BIDIRECIONAIS
 FONTE: MICHAEL ISNARDI, 1999

Um conjunto ordenado e regrado de quadros I, P e B são definidos como GOP. Cada GOP deve conter os três tipos de quadros, I, P e B e conter obrigatoriamente ao menos um quadro I. O número de quadros P e I são livres, salvo a condição anterior. O número de quadros B é de livre escolha, mas deve existir sempre o mesmo número de quadros B entre as imagens I e P. O GOP em sua codificação deve iniciar com um quadro I. A estrutura do GOP pode ser descrita por dois parâmetros: N, o número de quadros no GOP e M, o número de quadros B entre quadros P e I. Na Figura 8 é ilustrada uma estrutura de GOP com, N igual a 12 e M igual a dois, ou (12,2).

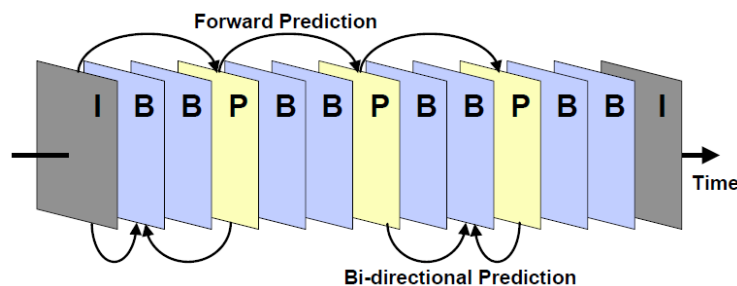


FIGURA 8 - GOP COM OS CONJUNTOS DE IMAGENS COM PREDIÇÕES FUTURAS E BIDIRECIONAIS.

FONTE: MICHAEL ISNARDI, 1999

2.2 PADRÕES DE QUALIDADE

Ecossistemas de comunicações abrangem uma grande área de problemas técnicos, pois essas lidam com trabalhos de modelos de resolução, de interação e do comportamento de pessoas com os aplicativos. Um dado aplicativo pode ter tanto relação com a rede, como uma relação com o usuário. Para que esse sistema funcione, é importante que essa relação, ou canal de comunicação, seja eficiente e que exista um padrão mínimo que assegure a qualidade da informação transmitida. Segundo Kilkki (2008) mecanismos de qualidade adicionados à rede garantem que o usuário final receberá um serviço melhor. Mas não somente será recebido com qualidade melhor, mas será garantido um estado mínimo necessário para que a comunicação seja dada como suficiente.

Dentro dessa qualidade existem dois conceitos chaves que são o QoS (*Quality of Service*) e o QoE (*Quality of Experience*). Em um dado sistema um QoS deve ser assegurado entre a aplicação e a rede e o QoE deve ser assegurado entre a aplicação e o usuário, conforme ilustrado na Figura 9.

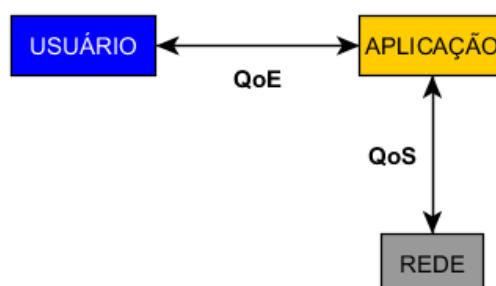


FIGURA 9 - QOE DEVE SER GARANTIDA ENTRE O USUÁRIO E APLICAÇÃO E QOS DEVE SER GARANTIDA ENTRE A APLICAÇÃO E A REDE

2.1.1 QoS

Toda aplicação consome recursos de rede e, dependendo da aplicação, é necessário garantir um nível essencial mínimo de recursos. Em uma conversa por VoIP (*Voice Over Internet Protocol*) é fundamental que a voz seja transmitida e recebida dentro de um limite máximo de atraso e de *jitter*, o que altera a qualidade da ligação. Já em uma conversa por meio de um *chat* de texto a mensagem enviada não possui a característica de ser sensível a atrasos.

O protocolo IP foi concebido com a regra do melhor esforço, o que não previa mecanismos de QoS, o que compromete a alocação de recursos da rede. No entanto, o IETF definiu diversos mecanismos para implementação de QoS para o IP. Desta forma o QoS deve garantir o desempenho, principalmente em gargalos da rede, em relação a determinadas métricas, principalmente:

- a) Taxa de transmissão;
- b) Atraso fim a fim;
- c) *Jitter*;
- d) Perda de Pacotes;

Com o QoS os pacotes são tratados de forma diferenciada, necessidades das aplicações, criando filas e prioridades distintas nos roteadores.

2.1.2 QoE

Para todo prestador de serviços de *Internet*, existe uma dependência do serviço prestado e da satisfação do cliente final com o serviço recebido. Para a ITU, o QoE é a aceitabilidade global de uma aplicação ou serviço, com um percebimento subjetivo de um usuário final. Buscando o sentido nas palavras temos para a qualidade: grau de perfeição, de precisão de conformidade a certo padrão, característica básica de alguma coisa. E para experiência: Conhecimento das coisas pela prática ou pela observação, conhecimento adquirido graças aos dados fornecidos pela própria vida. (Michaelis, 2013). Juntando as definições podemos definir o QoE como: características básicas ou naturais da participação ou observação direta de uma pessoa, neste caso, a sua participação na percepção de qualidade final da transmissão do vídeo.

O QoE refere-se sobre a percepção da qualidade percebidas pelo usuário, diferenciado claramente do QoS, que possui parâmetros facilmente mensuráveis e calculáveis. Tratando-se de vídeo pode-se definir que o QoE é a percepção do

usuário com qualidade resultante de uma transmissão afetada por um determinado nível de distorção. Distorção essa que pode ser gerada tanto na preparação do vídeo, na transmissão, na conversão, como também na decodificação do sinal. Dessa forma, além das métricas objetivas necessárias em uma rede (atrasos, perdas e latência), torna-se necessário considerar outros aspectos que tem impacto na qualidade de experiência de utilização, como: sincronismos entre áudio e vídeo, tempo de resposta a mudança de canal e fatores relacionados a própria qualidade de vídeo. (TELECO, 2008).

2.3 MOS

Para garantir a qualidade ao cliente, é importante que os prestadores de serviços mensurem o QoE de seus sistemas. Um dos métodos de medida é o MOS (*Mean Opinion Score*) que foi desenvolvido para a análise de testes, do ponto de vista do usuário, para ligações telefônicas. Segundo o ITU-R, na recomendação BT.500-13, o cálculo do MOS deve se dar nas seguintes condições:

O MOS não é utilizado somente para ligações telefônicas, mas também para áudios e vídeos. Ele disponibiliza uma indicação numérica para a qualidade recebida pelo usuário, de um vídeo, por exemplo, que sofreu compressão e foi transmitido. A nota de avaliação do MOS consiste de uma escala unitária começando pelo um e terminando no cinco, onde a nota mais baixa é o um, definindo uma qualidade muito ruim, e o cinco sendo a maior nota com uma qualidade excelente. Na Tabela 1 são apresentadas as possíveis avaliações de MOS e seu significado de qualidade.

Para avaliação das simulações realizadas nesse trabalho será necessário a realização do cálculo do MOS para cada situação utilizada na simulação. Será necessário também um número considerável de pessoas, fora a sala adequada, para que o levantamento seja estatisticamente aceitável, fugindo de incertezas causadas pela avaliação de apenas uma pessoa. No entanto, essa avaliação é inviável e será necessário utilizar uma ferramenta que possibilite essa avaliação automaticamente e numericamente.

TABELA 1 - TABELA DE AVALIAÇÃO MOS

MOS	Qualidade
5	Excelente
4	Bom
3	Razoável
2	Pobre
1	Ruim

Fonte: ITU P.800 – página 18

Para a transmissão alguns fatores são bem prejudiciais para a sua QoE, fatores esses como, perda de pacote, *delay* e *jitter*. No caso do vídeo esses fatores agravam, pois pôr o vídeo possuir compactação e compressão, se ele perder alguma informação que é referência de reconstrução de outra, será perdida a informação toda. Assim a redundância comprimida não poderá ser reconstruída, nesse caso um pacote perdido pode prejudicar a reprodução a da imagem durante um período considerável (Richardson, 2003).

3 SUAUIZAZÃO DE TRÁFEGO

3.1 MÉTODOS CONTROLADORES DE TRÁFEGO

Neste trabalho foram utilizados três métodos suavizadores, onde todos possuem suas premissas baseadas no algoritmo do Balde de Fichas.

3.1.1 Balde Furado

O Balde Furado consiste em converter um fluxo irregular de entrada em um fluxo regular de saída. Os pacotes transmitidos são inseridos no balde pelo transmissor. A seguir, ao saírem pelo fundo regulador do balde, os pacotes são transmitidos a uma taxa constante r [Bytes/s] para o receptor. O tamanho B [Bytes] do balde limita a fila de espera para a transmissão, caso o balde esteja cheio, os pacotes excedentes serão descartados. A taxa que determina o fluxo de transmissão é dada pela Expressão 1:

$$t = \frac{d}{r} \quad (1)$$

O d [Bytes] é o tamanho do pacote e o t [s] é o tempo de transmissão de um pacote d a uma taxa r , dado em Bytes por s. As unidades dos tamanhos d e r devem ser equivalentes. No balde furado um fluxo variável é obrigado a especificar um perfil correspondente a sua taxa máxima, o que pode levar a um uso ineficiente da rede. Para tais casos é indicada a utilização de uma derivação do Balde Furado, o Balde de Fichas.

3.1.2 Balde de Fichas

Para maior eficiência de utilização do algoritmo do Balde Furado, no Balde de Fichas foi acrescentado um *buffer* que espera o preenchimento do balde para transmissão de um pacote. O algoritmo consiste em regular um fluxo com um balde de tamanho B [Bytes], o balde é preenchido com fichas que são inseridas a uma taxa constante r [Bytes/s]. Caso não sejam utilizadas as fichas no balde, as fichas excedentes são descartadas.

No algoritmo do Balde Furado o pacote a ser transmitido entra no balde e sai a uma taxa constante, já no caso do Balde de Fichas o pacote com tamanho d [Bytes], é colocado em um *buffer*. Enquanto o pacote está no *buffer* é realizada uma comparação do tamanho do pacote com número de fichas existentes no balde. Caso haja um número de pacotes maior ou igual ao número de fichas, o pacote é transmitido e no balde são subtraídas d fichas. Caso não exista o número suficiente de fichas, o pacote espera o número de fichas se igualarem, nesse caso o pacote deve esperar a inserção de $d - B$ fichas no balde. O Balde de Fichas nunca descarta um pacote, pois cabe a atribuição do fluxo determinar o que fazer se não houver recursos de rede suficientes para atendê-lo.

O Balde de Fichas permite uma irregularidade no fluxo de dados, denominada de rajadas. O Balde Furado permite uma saída máxima de r Bytes por unidade de tempo e o Balde de Fichas garante que em um intervalo de tempo i , o fluxo máximo enviado não ultrapassa um limite dado pela Expressão 2.

$$n^{\circ} \text{ fichas de dados} = \frac{B+i}{r} \quad (2)$$

A fim de obter mais controle do método, foram criadas duas derivações do Balde de Fichas o SRTCM e TRTCM.

3.1.3 SRTCM (*Single Rate Three Color Marker*)

A técnica de suavização através do SRTCM é realizada através de dois baldes. Nessa técnica existem dois baldes com tamanhos independentes em série, Bc e Be. O Balde Bc é alimentado com fichas a uma taxa constante CIR [*Bytes/s*] (*Committed Information Rate*). O tamanho de Bc é definido pela rajada máxima do tráfego na taxa médias dada como CBS [*Bytes*] (*Committed Burst Size*). Já o tamanho do balde Be é dada pelo excesso de rajada ainda tolerável na transmissão, dada como EBS [*Bytes*] (*Excess Burst Size*).

No Balde de Fichas as fichas excedentes eram descartadas, no caso do SRTCM as fichas excedentes do Bc alimentam o balde Be. O pacote a ser analisado aguarda a verificação em um *buffer*. Para cada pacote de entrada existem duas possibilidades de estados em cada balde, totalizando três distintos estados de saída. O nome SRTCM, vem justamente que com uma taxa de entrada de fichas, existem três estados de saída, cada qual marcado com uma cor.

Com os parâmetros da técnica de suavização definida, é possível realizar uma análise do seu funcionamento. O pacote que está em espera no *buffer* sofre uma análise do primeiro balde. Se o tamanho do balde Bc for maior do que o tamanho do pacote B ele é transmitido e marcado com um pacote verde, nesse caso existe apenas um decremento apenas do balde Bc. Caso o tamanho do balde seja menor que o tamanho do pacote, ele passa a ser analisado pelo balde Be. No caso do tamanho do balde Be for maior que o tamanho B do pacote, ele é transmitido e marcado como amarelo. Nesse caso existe apenas o decremento do balde Be. O último estado possível para o SRTCM é de quando o pacote B é maior que o balde Bc e o balde Be, nesse caso ele é marcado como um pacote vermelho e não existe decremento de nenhum balde. Para o pacote ser transmitido, e evitar uma pico muito alto numa rajada de trafego, deve-se criar uma regra de transmissão. No caso desse trabalho, os pacotes vermelhos aguardam o balde Bc possuir um número equivalente de fichas no balde Bc para que seja feita a sua transmissão. Na Figura 10, é possível a visualização do método analisado.

O SRTCM permite rajadas de pacotes, mas com a dependência da entrada no número de pacotes. O decremento é realizado apenas em um balde e tamanho

da rajada é limitado pelo tamanho máximo do balde. Nessa técnica, um pacote de entrada que inicialmente é vermelho será sempre vermelho, um pacote amarelo, pode ser amarelo ou vermelho e um pacote verde pode continuar verde e poderá se tornar em amarelo ou vermelho, dependendo do estado que os baldes se encontram.

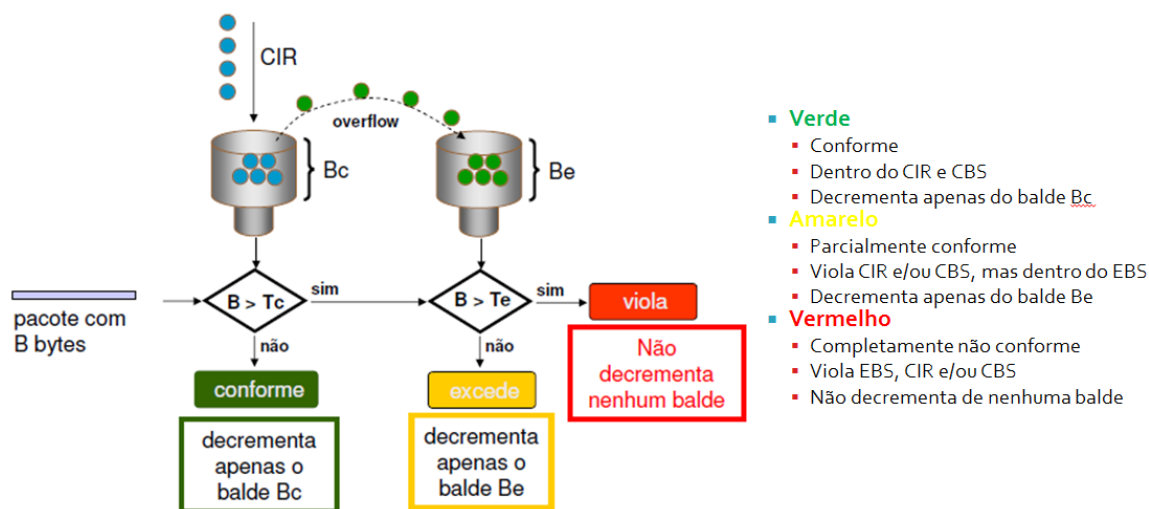


FIGURA 10 - MÉTODO DE SUAVIZAÇÃO SRTCM, EXISTE UMA ENTRADA DE FICHAS, AS FICHAS DE TRANSBORDO PASSAM PARA O SEGUNDO BALDE. EXISTEM TRÊS ESTADOS POSSÍVEIS DE SAÍDA.

3.1.4 TRTCM (*Two Rate Three Color Marker*)

O TRTCM como o SRTCM também é uma derivação do Balde de Fichas. O que dá o nome ao TRTCM e faz com que exista a diferença nos dois métodos, é de que as fichas de transbordo foram trocadas por uma entrada de fichas, ficando duas entradas de fichas. A marcação de cor ocorre de maneira diferenciada, mas ainda existem três estados de saída.

No TRTCM os baldes estão conectados em série, sendo que cada um possui uma entrada independente de fichas. No primeiro a entrada é definida pela taxa máxima de picos permitida PIR [*Bytes/s*] (*Peak Information Rate*), para o segundo balde a taxa de entrada das fichas é dada pela CIR [*Bytes/s*] (*Committed Information Rate*). Os tamanhos dos baldes são definidos pela rajada máxima permitida de pico e de tráfego médio. No primeiro balde o tamanho é definido por PBS [*Bytes*] (*Peak Burst Size*) e o tamanho do segundo balde é definido por CBS [*Bytes*] (*Committed Burst Size*).

Para análise do tráfego os baldes devem estar inicialmente cheios. O pacote a ser analisado é colocado em um *buffer* até ser realizada a sua análise. No caso do

tamanho do pacote B for maior que o tamanho do primeiro balde o pacote é considerado como vermelho e não feito nenhum decremento nos baldes. Como no caso do estado vermelho do SRTCM, no TRTCM, é realizada a espera do primeiro balde possuir o número de fichas para a sua transmissão, mas não é feito nenhum decremento no balde. Quando o pacote não for vermelho, ou seja, quando o tamanho do primeiro balde for maior do que o tamanho do pacote, são possíveis outros dois estados. No primeiro estado, quando o tamanho do pacote B é maior que o tamanho do segundo balde, o pacote é marcado amarelo, é feita a sua transmissão com decremento apenas no primeiro balde. Caso o tamanho B do pacote seja menor em ambos os baldes, ele é marcado como um pacote verde e a sua transmissão é realizada após o decremento em ambos os baldes. Na Figura 11 está representado o método TRTCM.

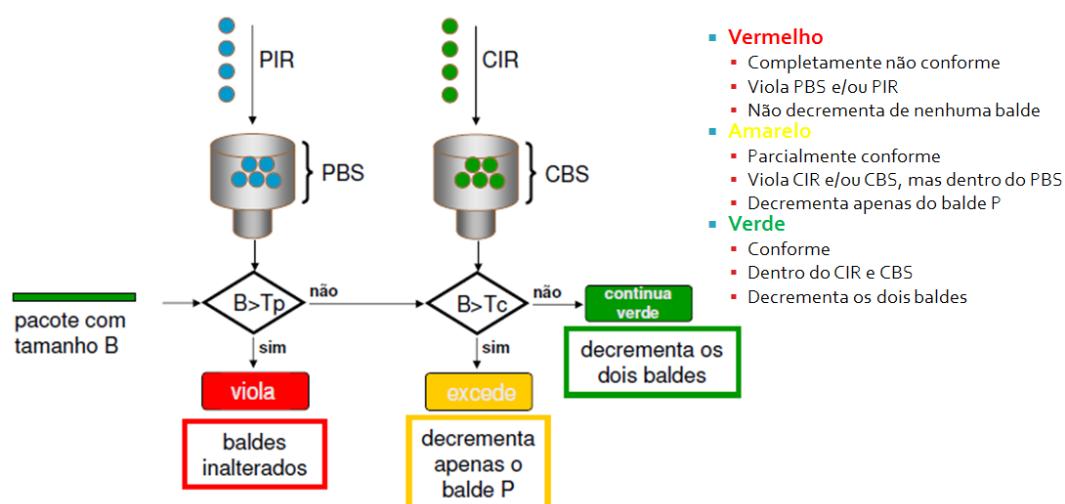


FIGURA 11 - MÉTODO DE SUAVIZAÇÃO TRTCM, EXISTE DUAS ENTRADAS DE FICHAS E TRÊS ESTADOS POSSÍVEIS DE SAÍDA.

4 TECNOLOGIAS UTILIZADAS

4.1 NS2 – NETWORK SIMULATOR 2

As simulações desenvolvidas no presente trabalho foram realizadas em um *software* chamado *Network Simulator* (NS). O NS é um *software* de código aberto desenvolvido pela Universidade de Berkley em parceria com companhias como:

DARPA, USC/ISI, Xerox Parc e LBNL. O simulador oferece suporte à simulação de um grande número de ambientes de redes com diferentes cenários baseados nos protocolos TCP e UDP. Em cada cenário, parâmetros podem ser alterados, como as políticas de fila, caracterização de tráfego, bem como realizar a transmissão de arquivos. O NS é desenvolvido na linguagem C++ e as configurações de topologias de simulação são escritas na linguagem OTCL. Neste presente trabalho foi utilizado o *Network Simulator 2 (NS2)*.

Como o NS é um *software* de código aberto, é possível criar, alterar ou até agregar códigos para o desempenho de uma determinada função. No NS a interface com o usuário é feita através da linguagem interpretada OTCL (*Object-oriented Tool Command Language*). O OTCL é uma linguagem *frontend* desenvolvido pela MIT (*Massachusetts Institute of Technology*). Uma linguagem *frontend* a linguagem utilizada para interação do *software* com o usuário. Através do OTCL são efetivamente descritas as simulações no NS. Nela são criados os enlaces, taxas de transmissão, atrasos, protocolos utilizados, tamanho de enfileiramento, bem como todas as necessidades de para a simulação de um ambiente, como a transmissão de um arquivo de vídeo.

4.2 EVALVID

O EvalVid é uma ferramenta desenvolvida pela Universidade Técnica de Berlim que calcula o valor do MOS de um vídeo que sofreu alguma transmissão. A premissa do cálculo do MOS é o cálculo do PSNR dos vídeos. Com a comparação do valor do PSNR do vídeo transmitido e do vídeo recebido é possível qualificar com o valor do MOS. Para a utilização do EvalVid na simulação não é necessário a transmissão do vídeo, e sim apenas a transmissão de pacotes que informam o tamanho e o tempo de cada parte do vídeo. No final da simulação é realizada uma comparação dos dados recebidos (com possíveis erros) com os dados do vídeo de referência. Em uma etapa do EvalVid é feita a retirada das informações recebidas erroneamente no vídeo de referência, resultando o vídeo que seria recebido na transmissão.

A avaliação de vídeo digital é baseada na percepção de qualidade do vídeo que está sendo recebido pelo usuário. Muitas pesquisas e recursos estão sendo investidos para o a automatização dos recursos de avaliação de vídeo. Mas uma das

maneiras encontradas para essa avaliação é o cálculo do pico da relação sinal ruído de cada imagem, o PSNR. O PSNR é uma derivação do SNR (*Signal Ratio Noise*) que é a relação do sinal de energia com a energia do erro. O PSNR compara os picos máximos da função SNR do sinal de energia e da energia do ruído. Como o PSNR é calculado *frame a frame*, e um vídeo pode conter milhares de *frames* o seu cálculo pode se tornar lento. O valor do PSNR é calculado na componente de luminância do vídeo. A comparação é feita quadro a quadro de uma imagem de fonte S para um uma imagem de destino D é dada pela seguinte expressão.

$$PSNR(n)_{dB} = 20 \log_{10} \left(\frac{V_{pico}}{\sqrt{\frac{1}{N_{coluna} N_{linha}} \sum_{i=0}^{N_{coluna}} \sum_{j=0}^{N_{linha}} [Y_S(n,i,j) - Y_D(n,i,j)]^2}} \right) \quad (3)$$

$$V_{pico} = 2^k - 1 \quad (4)$$

Onde k é o número de *bits* por *pixel*. A parte inferior da fração é o cálculo do erro médio quadrático (MSE – *Mean Square Error*), logo a expressão do PSNR pode ser simplificada para:

$$PSNR = 20 \log_{10} \left(\frac{V_{pico}}{MSE} \right) \quad (5)$$

Após testes no desenvolvimento dos cálculos envolvendo vídeos com o cálculos do MOS através do PSNR, chegou-se a um resultado final definido na Tabela 2.

TABELA 2 - TABELA DE AVALIAÇÃO MOS ATRAVÉS DO PSNR

PSNR [dB]	MOS
> 37	5 (Excelente)
31,01 – 37	4 (Bom)
25,01 – 31	3 (Razoável)
20 – 25	2 (Pobre)
< 20	1 (Ruim)

FONTE JIRKA KLAUE, 2003

O funcionamento do EvalVid está demonstrado na Figura 12, para facilitar o seu entendimento na imagem estão demonstrados os blocos de cada função. A ferramenta do EvalVid começa com a criação de vídeos de referência. O vídeo original que pode ser de qualquer fonte deve ser transformado em seu formato cru, (YUV), esse mesmo vídeo será codificado para o MPEG que servirá de base para o vídeo de resultado, com ou sem os erros causados pela simulação. Os cálculos das simulações de rede realizadas serão realizadas na etapa *Network*, do diagrama da Figura 12.

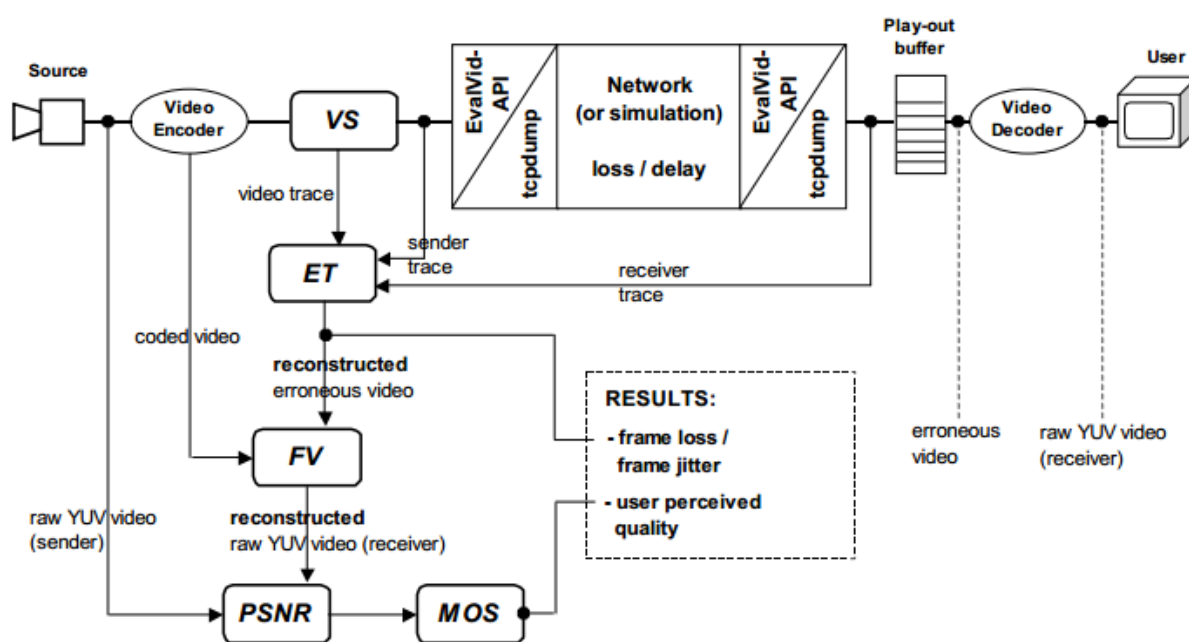


FIGURA 12 - FLUXOGRAMA DE FUNÇÕES PARA A EXECUÇÃO DO EVALVID
 FONTE: JIRKA KLAUE, 2003

4.2.1 VS – Video Sender

Uma vez que o vídeo é codificado, o VS pode criar o arquivo de *trace*. Esse arquivo contém todas as informações relevantes para que a ferramenta EvalVid possa trabalhar. Dois arquivos de *traces* são gerados, neles estão contidas informações sobre todos os quadros do arquivo de vídeo e todos os pacotes gerados para a transmissão. O quadro H é a marcação de início do vídeo. As Figuras 13 e 14 ilustram os dois arquivos de *trace* gerados pelo VS.

Frame Number	Frame Type	Frame Size	Number of UDP-packets	Sender Time
0	H	24	1 segm	40 ms
1	I	9379	10 segm	80 ms
2	P	2549	3 segm	120 ms
3	B	550	1 segm	160 ms
...				

FIGURA 13 - ARQUIVO DE TRACE GERADO, PELO VIDEO SENDER.
FONTE: JIRKA KLAUE, 2003

time stamp [s]	packet id	payload size
1029710404.014760	id 48946	udp 24
1029710404.048304	id 48947	udp 1024
1029710404.048376	id 48948	udp 1024
...		

FIGURA 14 - ARQUIVO DE TRACE GERADO COM O TEMPO DE CADA PACOTE, A SUA IDENTIFICAÇÃO E O TAMANHO DO PACOTE UDP.
FONTE: JIRKA KLAUE, 2003

Esses dois arquivos de *trace* juntos representam uma transmissão completa de vídeo, do lado da transmissão, e contém todas as informações necessárias para a avaliação futura do EvalVid. A rede causa atrasos e provavelmente perda e reordenação dos pacotes. Na recepção deve-se também criar um arquivo como o da geração dos *traces*.

4.2.2 ET – Evaluate Traces

O ET é considerado o coração da avaliação dos quadros no EvalVid. No ET são calculadas as perdas de pacotes e de quadros, além de realizar o cálculo do *delay* e do *jitter*. Para o cálculo da perda de pacotes é contabilizado cada tipo de quadro na entrada e na saída da simulação, sendo possível saber a porcentagem de perda de cada tipo de quadro. O ET realiza o cálculo a partir dos *traces* de entrada e de saída, tendo como o vídeo codificado de referência. No final do ET é criada uma cópia do vídeo com a alteração onde existiram erros na transmissão, sendo o vídeo recebido, como resultado final da transmissão.

PACKET LOSS			FRAME LOSS				
H:	1	0	0.0%	H:	1	0	0.0%
I:	2825	3	0.1%	I:	375	3	0.8%
P:	2210	45	2.0%	P:	1125	45	4.0%
B:	3266	166	5.1%	B:	2998	166	5.5%
ALL:	8302	214	2.6%	ALL:	4499	214	4.8%

FIGURA 15 - AVALIAÇÃO REALIZADA PELO ET. CÁLCULO DE PERDA DE PACOTES E DE QUADROS, SENDO DEMONSTRADO O TAMANHO TOTAL, O PERDIDO E A PERDA RESPECTIVA EM PORCENTAGEM. O QUADRO H É O QUADRO DE INÍCIO DE TRANSMISSÃO A SUA COMPACTAÇÃO É EQUIVALENTE AO QUADRO I.

FONTE: JIRKA KLAUE, 2003

4.2.3 FV – *Fix Video*

Em casos que os codificadores não conseguirem lidar com os quadros perdidos o FV é acionado para a correção. O FV também recebe o vídeo resultante da simulação e o constrói em seu formato cru.

4.2.4 PSNR

Na etapa do PSNR são calculados os valores do PSNR. São feitos dois cálculos, primeiro do vídeo de referência com o vídeo codificado. O segundo valor calculado é o do vídeo de referência com o valor do vídeo recebido na transmissão. A etapa que define o valor final do vídeo é realizada pela comparação dos dois valores de PSNR calculado. A primeira etapa foi implementada para reduzir a possibilidade de existirem valores iguais de *bits* nas imagens, o que levaria o valor de PSNR a zero, pela Equação 3. Com o valor do PSNR calculado é possível realizar a sua avaliação no MOS. Na Figura 16 está ilustrado o sinal de PSNR de um vídeo, onde na parte superior da figura é apresentado um sistema com poucas perdas e a segunda com altas taxas de perdas.

4.2.5 MOS

Com o valor do PSNR calculado a ferramenta do MOS realiza a comparação dos dois valores de PSNR e estipula através da Tabela 2 o valor final MOS. O resultado do MOS é agregado aos resultados do ET e são entregues os resultados finais da avaliação.

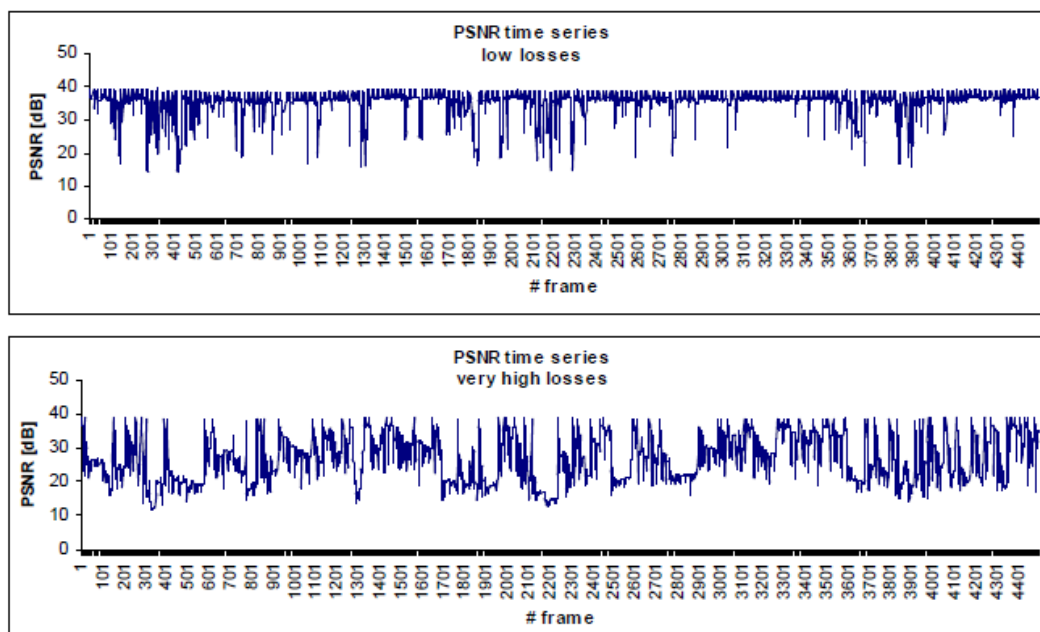


FIGURA 16 - VALORES CALCULADOS EM SIMULAÇÃO COM RESULTADO DE BAIXA PERDA E ALTAS PERDAS E SUAS INFLUÊNCIAS NO RESULTADO DO PSNR.
 FONTE: JIRKA KLAUE, 2003

4.3 TÉCNICAS DE SUAVIZAÇÃO

Todos os arquivos que foram simulados pelo programa NS2 tiveram que ser adaptados para que o programa pudesse entender e trabalhar com suas informações. Para cada arquivo de vídeo testado foi criado um arquivo de *trace* onde continham as informações de tamanho e tempo de cada pacote. A adaptação para o NS2 ocorre no momento que é criado um arquivo hexadecimal contendo apenas as informações de tempo e tamanho de cada pacote.

Os métodos suavizadores foram implementados na criação desse arquivo de transmissão. Para cada método suavizador foi criado um arquivo diferente, cada qual com as suas regras estabelecidas no Capítulo 3. A implementação desses métodos foi feita através da linguagem C++ e seus códigos encontram-se no Apêndice.

4.4 ORIGEM DOS DADOS

Para a simulação do sistema IPTV foram necessários à utilização de vídeos. Quando se trata de vídeo existe uma série de parâmetros que devem ser levados

em consideração. No caso da simulação, é buscada a solução de melhoria para o pior do caso, ou seja, vídeos com alta taxa de mudança de quadros.

Em um vídeo de um jogo de futebol, por exemplo, sempre existirá uma grande área verde pertencente ao conteúdo transmitindo, somente proporções menores, que seriam os jogadores correndo de um lado para o outro atrás da bola, pertencem ao grupo de constante mudança no vídeo. Em um noticiário, existem poucas mudanças bruscas, o vídeo é considerado um vídeo de imagens estáticas.

Para que seja possível explorar ambientes caóticos de transmissão, com alta taxa de ocupação, determinada pelo simulador, e com muitas transições nos vídeos, como mudança de cor, luminância, posição de imagem e constante mudança de cenário é necessária realizar a simulação com vídeos de filmes. Para uma melhor avaliação é interessante simular ambientes com pouco grau de controle de transmissão de conteúdo, podendo-se avaliar de uma maneira melhor os métodos a serem testados. Os vídeos testados são amplamente utilizados em pesquisas (TKN, 2003).

5 SIMULAÇÃO

5.1 PARÂMETROS DE SIMULAÇÃO

Na simulação foi utilizado um vídeo de um trecho de 30 segundos do filme Jurassic Park. Com o arquivo de *trace* é possível estimar o tamanho dos quadros, a sua quantidade e tempo. Na Tabela 3 estão os valores dos tamanhos de quadro, número de pacotes, taxa e tempo.

TABELA 3 - VALORES DE PARÂMETROS OBTIDOS A PARTIR DO TRACE DO VIDEO SIMULADO

Tamanho Total [B]	11006182
Tamanho Total [b]	88049456
Taxa [Bps]	354614,88
Taxa [bps]	2836919
Taxa Média [Mbps]	2,836919
Nº Pacotes Total	7789
Maior Pacote [B]	95846
Tempo Total [s]	31,037

A fim de trazer os resultados próximos da realidade através da simulação, a topologia utilizada na simulação é mostrada pela Figura 17. A topologia criada pelo NS2 está ilustrada na Figura 18.

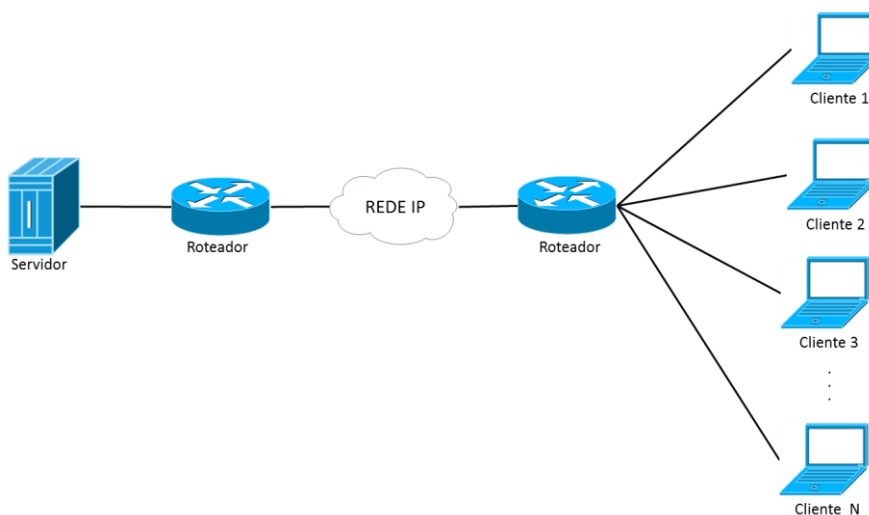


FIGURA 17 – TOPOLOGIA SIMPLIFICADA DO SISTEMA DE REDE DO IPTV

A taxa de transferência média do vídeo é dada pela soma do tamanho dos quadros dividida pelo tempo total do vídeo, dada pela Expressão 7:

$$\lambda = \frac{\sum \text{Quadros}}{\text{Tempo}} \left[\frac{B}{s} \right] \quad (6)$$

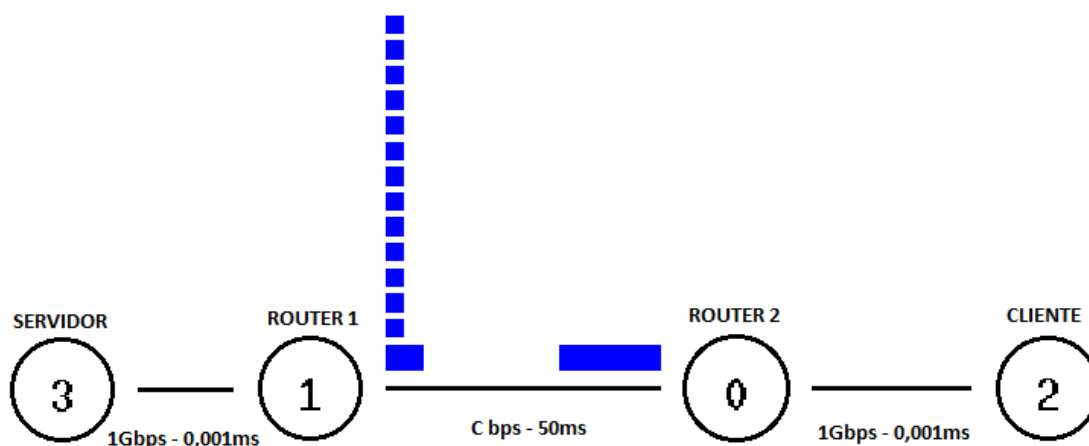


FIGURA 18 - TOPOLOGIA CRIADA PARA A SIMULAÇÃO DE TRANSMISSÃO DE IPTV NO SOFTWARE NS2

Para análise do comportamento da transmissão na simulação foram utilizadas diversas taxas de ocupação. A taxa de ocupação é uma grandeza

adimensional com tamanho máximo de um, onde a transmissão está ocupando a totalidade do canal disponível. Na simulação foi utilizada uma escala de taxa de ocupação contemplando as diversas possibilidades da realidade. A taxa de ocupação pode ser definida pela divisão da taxa de transferência do vídeo pela velocidade do canal, dada pela Expressão 8. Sabendo-se da taxa de ocupação desejada, é possível determinar o tamanho do canal da simulação.

$$\rho = \frac{\lambda}{\mu} \quad (7)$$

Considerando-se que a quantidade de informações do vídeo é de 11.006.182 *Bytes* e o tempo total 31,037 segundos e utilizando a Expressão 8, é possível calcular a capacidade do canal necessária para obter os diversos níveis de utilização desejados. Os resultados são listados na Tabela 4.

TABELA 4 - VALORES DE OCUPAÇÃO DE CANAL UTILIZADOS NA SIMULAÇÃO.

Ocupação [%]	Velocidade do Canal [Mbps]
10%	28,369
20%	14,185
30%	9,456
40%	7,092
50%	5,674
60%	4,728
70%	4,053
80%	3,546
90%	3,152
95%	2,986

Nas simulações realizadas o tamanho da fila no roteador foi limitado de modo a causar diversos níveis de perda de pacotes. Foram testados dois ambientes o com 1% de perda e a condição que o *buffer* do roteador possuía tamanho infinito. Dentro dos parâmetros de perda de simulação também foi inserido a condição que todo pacote com mais de 200ms de atraso ser considerado como perdido (DSL Fórum, 2006).

Em um algoritmo de suavização, para que os quadros transmitidos sejam de fato suavizados, é necessário que o tamanho do balde b seja menor que o tamanho do maior quadro. Caso contrário, nos quadros que geram pico poderiam ser transmitidos sem ser suavizados. Considerando que os quadros I são os maiores e Φ_I denotando o tamanho médio dos quadros I , então b deve ser menor que Φ_I .

$$b < \Phi_I \quad (8)$$

Para que o quadro suavizado não atrase os quadros subsequentes é importante estabelecer um tempo de transmissão para ele. Como a taxa de quadros é dada em f quadros por segundo, é possível definir que o quadro I deve ser transmitido no máximo até $1/f$. Nesse caso:

$$t_{\Phi_I} < \frac{1}{f} \quad (9)$$

No enlace de transmissão C representa a taxa de transmissão do canal em bps, com o tamanho do b do balde, inicialmente cheio, podemos definir que o tempo de transmissão de um quadro I será dado por:

$$t_{\Phi_I} = \frac{b}{C} + \frac{\Phi_I - b}{r} \quad (10)$$

Utilizando as Equações 9 e 10, é possível escrever:

$$\frac{b}{C} + \frac{\Phi_I - b}{r} < \frac{1}{f} \quad (11)$$

Isolando em função de b e simplificando a equação, obtém-se:

$$b < \frac{C(r - f\Phi_I)}{f(r - C)} \quad (12)$$

Utilizando a Equação 9 e a Equação 13 foi obtido o valor de b . A fim da busca de resultados otimizados, foram utilizadas duas condições onde b deve ser no

máximo Φ_1 e $\Phi/2$. Para os métodos SRTCM e TRTCM são utilizados dois baldes, a sua soma equivale ao balde utilizado no método de Balde de Fichas. Utilizando-se estas restrições obtêm-se os valores da Tabela 5. A taxa de entrada de fichas no balde é igual a r . No caso do TRTCM que são duas entradas, a entrada do segundo balde é igual duas vezes r .

TABELA 5 - CÁLCULO DO TAMANHO DO BALDE. MÉTODO BALDE E SRTCM E TRTCM. NO SRTCM E TRTCM SÃO UTILIZADOS DOIS BALDES IGUAIS, A SUA SOMA É EQUIVALENTE AO TAMANHO DO BALDE DE FICHAS. TAMANHO MÁXIMO LIMITADO PELA EXPRESSÃO 9

Ocupação	C [Bps]	r [Bps]	F [fps]	Φ [B]	Tamanho no método Balde de Fichas		Tamanho nos métodos SRTCM e TRTCM	
					$b = \phi$ [B]	$b = \phi/2$ [B]	$b/2 = \phi$ [B]	$b/4 = \phi$ [B]
10%	3546148,8	354614,9	30	38692	29857	8362	14929	4181
20%	1773074,4	354614,9	30	38692	33589	9407	16795	4703
30%	1182049,6	354614,9	30	38692	38388	10751	19194	5375
40%	886537,2	354614,9	30	38692	38692	12543	19346	6271
50%	709229,76	354614,9	30	38692	38692	15051	19346	7526
60%	591024,8	354614,9	30	38692	38692	18814	19346	9407
70%	506592,68	354614,9	30	38692	38692	19346	19346	9673
80%	443268,6	354614,9	30	38692	38692	19346	19346	9673
90%	394016,53	354614,9	30	38692	38692	19346	19346	9673
95%	373278,82	354614,9	30	38692	38692	19346	19346	9673

Ao todo foram realizadas 140 simulações separadas em dois grupos distintos, um com a sua porcentagem proposital de erro e a condição do *buffer* infinito. Em cada situação as 70 simulações foram realizadas com o mesmo tamanho de fila para cada porcentagem de ocupação. As simulações contemplaram a simulação sem método de suavização nenhum e os três métodos suavizadores testados.

6 RESULTADOS OBTIDOS

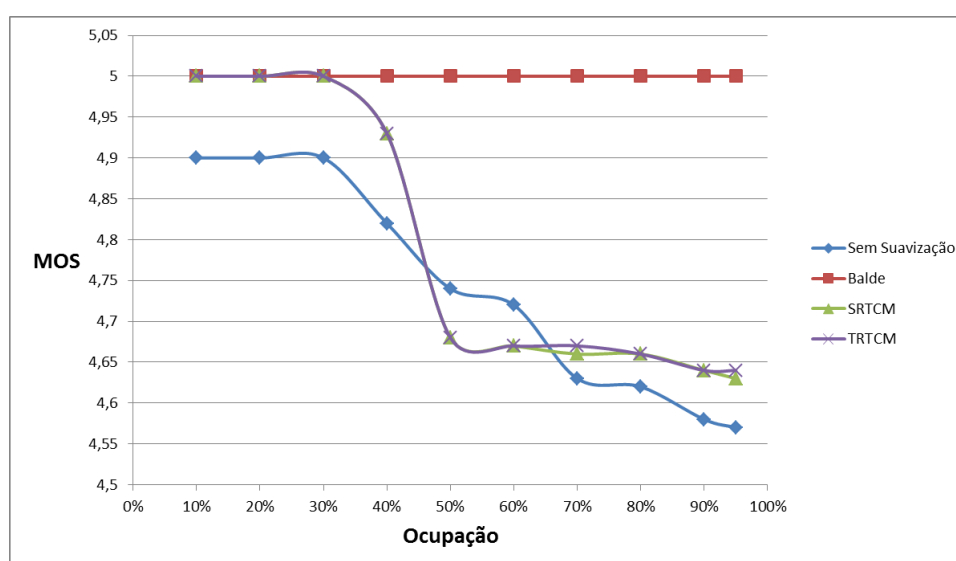
A partir dos resultados obtidos foram analisados o MOS, o *delay* e o *jitter*. Os resultados foram tabelados e graficados. A sua demonstração serão divididas pela porcentagem de erro inserido na simulação. Em cada ponto está demonstrada a tabela de resultados de simulação e os seus respectivos gráficos.

6.1 PERDA IGUAL A 1%

De modo a testar um ambiente com perdas, foi estabelecida uma perda fixa de 1% dos pacotes na transmissão. Esta perda composta por erros na transmissão e por pacotes que possuem tempo maior do que 200ms de transmissão. O erro foi configurado pelo tamanho do *buffer* do roteador, limitando o número de pacotes enfileirados. Para cada porcentagem de ocupação o tamanho de fila é diferente, o número utilizado foi o que permitisse a perda mais próxima de 1%. Mesmo com a tentativa de manter o 1%, nos casos de ocupações mais altas, não era possível, pois a perda real é maior.

TABELA 6 - RESULTADO DO MOS A PARTIR DA SIMULAÇÃO COM 1% DE PERDA

Ocupação	MOS		$b < \phi$			$b < \phi/2$		
	Sem Suavização	Balde	SRTCM	TRTCM	Balde	SRTCM	TRTCM	
10%	4,9	5	5	5	5	5	5	
20%	4,9	5	5	5	5	5	5	
30%	4,9	5	5	5	5	5	5	
40%	4,82	5	4,93	4,93	5	5	5	
50%	4,74	5	4,68	4,68	5	4,97	4,98	
60%	4,72	5	4,67	4,67	5	4,68	4,68	
70%	4,63	5	4,66	4,67	5	4,68	4,68	
80%	4,62	5	4,66	4,66	5	4,66	4,67	
90%	4,58	5	4,64	4,64	5	4,66	4,66	
95%	4,57	5	4,63	4,64	5	4,66	4,66	

FIGURA 19 - MOS CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A ϕ_1

Para a transmissão com perda de 1% e com o tamanho do balde igual a Φ_1 o balde de fichas obteve uma resposta de simulação sem nenhuma perda de qualidade de sinal, sendo seu MOS igual a 5 em todas as ocupações testadas. Os valores para o SRTCM e TRTCM foram muito parecidos, sendo que o valor de seu MOS divergiu pouco em apenas dois pontos, 70% e 99% de ocupação. Comparando ao valor simulado sem suavização o SRTCM e TRTCM obtiveram um acréscimo considerável nos casos com ocupação de até 45%. Entre 45% e 65% o seu valor foi inferior ao testado sem simulação. Nos casos acima de 65% obteve-se um ganho de MOS novamente.

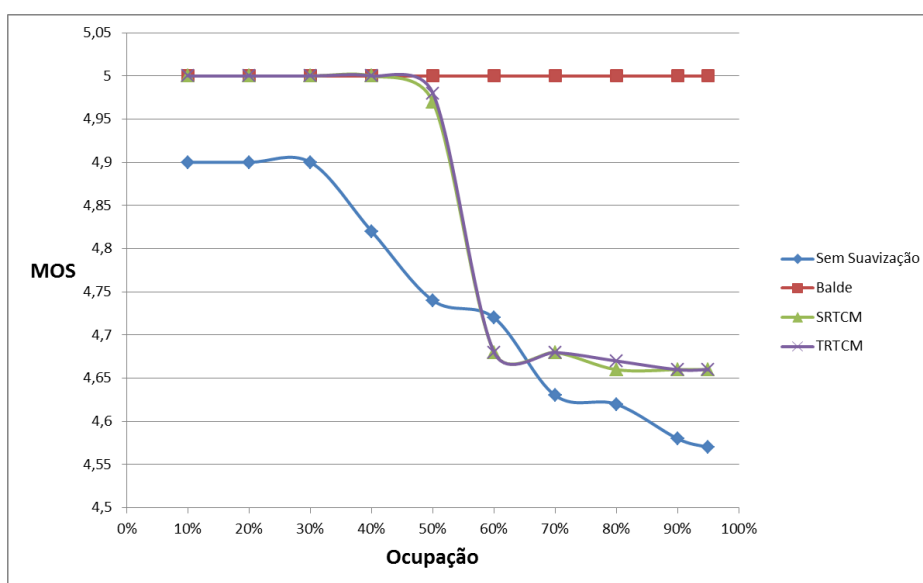
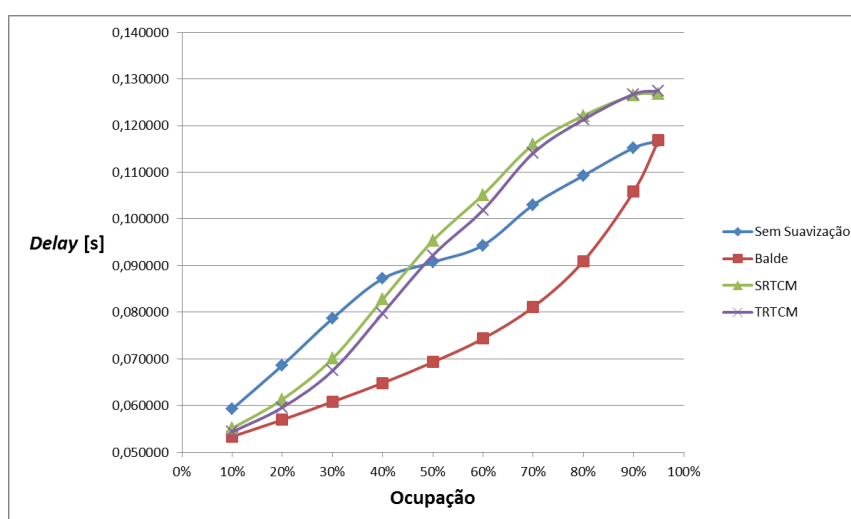


FIGURA 20 - MOS CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A $\Phi_1/2$

Os valores de MOS testados com o balde igual a $\Phi_1/2$ são semelhantes ao valor testado com o Φ_1 . O MOS do Balde de Fichas é constante igual a 5. No caso do TRTCM e SRTCM houve uma melhora, onde o valor do MOS começa a cair com uma porcentagem maior. Nesse caso o valor do MOS inferior ao testado sem nenhuma suavização é um pequeno intervalo perto dos 60% de ocupação.

TABELA 7 - RESULTADO DO DELAY A PARTIR DA SIMULAÇÃO COM 1% DE PERDA

Ocupação	DELAY		$b < \phi$			$b < \phi / 2$		
	Sem Suavização	Balde	SRTCM	TRTCM	Balde	SRTCM	TRTCM	
10%	59,3E-3	53,4E-3	55,1E-3	54,4E-3	53,1E-3	53,3E-3	53,3E-3	
20%	68,6E-3	57,0E-3	61,3E-3	59,6E-3	56,2E-3	57,2E-3	57,2E-3	
30%	78,6E-3	60,8E-3	70,1E-3	67,5E-3	59,6E-3	62,9E-3	62,8E-3	
40%	87,2E-3	64,8E-3	82,7E-3	79,8E-3	63,3E-3	72,5E-3	71,8E-3	
50%	90,7E-3	69,3E-3	95,3E-3	92,2E-3	67,7E-3	84,2E-3	83,4E-3	
60%	94,3E-3	74,4E-3	105,2E-3	101,9E-3	73,0E-3	103,3E-3	102,1E-3	
70%	103,0E-3	81,1E-3	115,9E-3	114,1E-3	79,7E-3	120,6E-3	119,5E-3	
80%	109,3E-3	90,9E-3	122,1E-3	121,3E-3	89,7E-3	128,7E-3	127,6E-3	
90%	115,2E-3	105,8E-3	126,5E-3	126,7E-3	104,8E-3	132,1E-3	130,8E-3	
95%	116,7E-3	116,9E-3	126,8E-3	127,4E-3	116,0E-3	131,8E-3	130,5E-3	

FIGURA 21 - DELAY CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A Φ_1

O *delay* do Balde de Fichas aumenta mais lentamente que os outros métodos suavizadores, havendo menor perda de qualidade. Já o SRTCM e o TRTCM possuem atrasos menores que o tráfego sem suavização até os 50% de ocupação, para valores superiores a 50% o atraso é maior.

Para o caso do balde igual a $\Phi_1/2$ o SRTCM e TRTCM possuem uma resposta igual e com um pequeno aumento referente ao Φ_1 . Já no caso do Balde de Fichas não houve mudança.

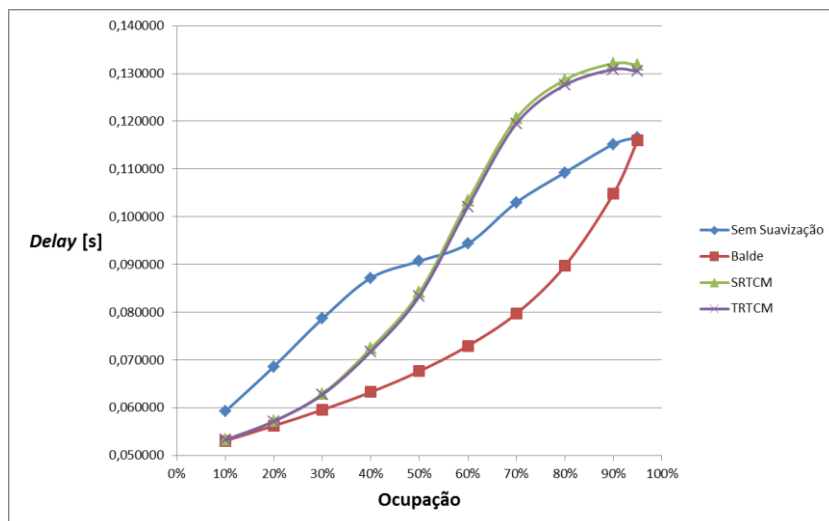
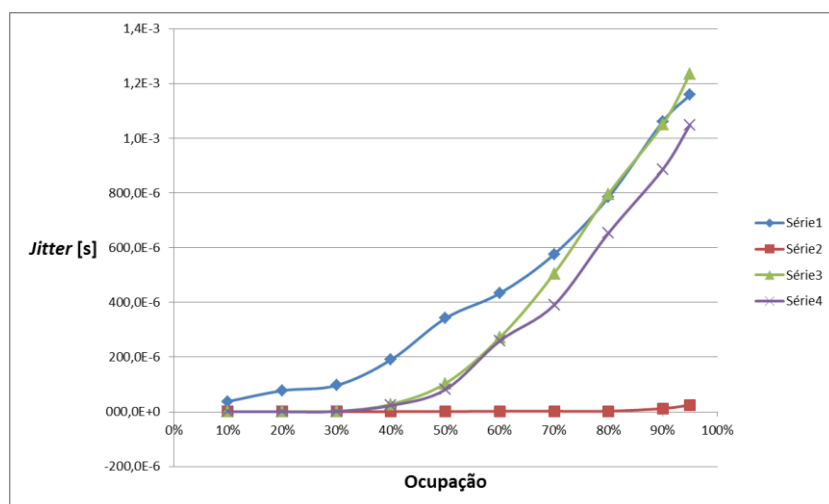
FIGURA 22 - DELAY CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A $\Phi/2$

TABELA 8 - RESULTADO DO JITTER A PARTIR DA SIMULAÇÃO COM 1% DE PERDA

JITTER		$b < \phi$			$b < \phi/2$		
Ocupação	Sem Suavização	Balde	SRTCM	TRTCM	Balde	SRTCM	TRTCM
10%	37,9E-6	207,4E-9	207,4E-9	207,4E-9	207,4E-9	207,4E-9	207,4E-9
20%	77,5E-6	415,0E-9	448,9E-9	448,9E-9	415,0E-9	415,0E-9	415,0E-9
30%	96,7E-6	622,6E-9	2,1E-6	1,3E-6	622,6E-9	1,5E-6	1,5E-6
40%	190,3E-6	830,3E-9	27,9E-6	23,1E-6	830,3E-9	6,8E-6	6,8E-6
50%	341,8E-6	1,0E-6	103,7E-6	81,3E-6	1,0E-6	10,5E-6	7,0E-6
60%	433,0E-6	2,2E-6	271,0E-6	259,0E-6	2,2E-6	130,3E-6	105,6E-6
70%	575,7E-6	2,4E-6	505,4E-6	390,4E-6	2,4E-6	413,5E-6	380,8E-6
80%	783,7E-6	2,5E-6	796,2E-6	653,2E-6	2,5E-6	885,4E-6	850,3E-6
90%	1,1E-3	12,3E-6	1,1E-3	886,4E-6	12,3E-6	1,2E-3	1,2E-3
95%	1,2E-3	24,4E-6	1,2E-3	1,0E-3	24,4E-6	1,4E-3	1,3E-3

FIGURA 23 - JITTER CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A Φ

Os valores de *jitter* calculados para 1% de perda e com o balde igual a Φ_1 são demonstrados na Figura 23. O valor da variação de atraso para o Balde de Fichas houve uma melhora considerável, pois o seu valor pouco aumentou. No caso do SRTCM e TRTCM, os valores acompanham a curva de crescimento da curva sem suavização. No caso do TRTCM o valor é inferior em todas as ocupações testadas. Já no caso do SRTCM os seus valores são significativamente menores em valores inferiores a 70% de ocupação.

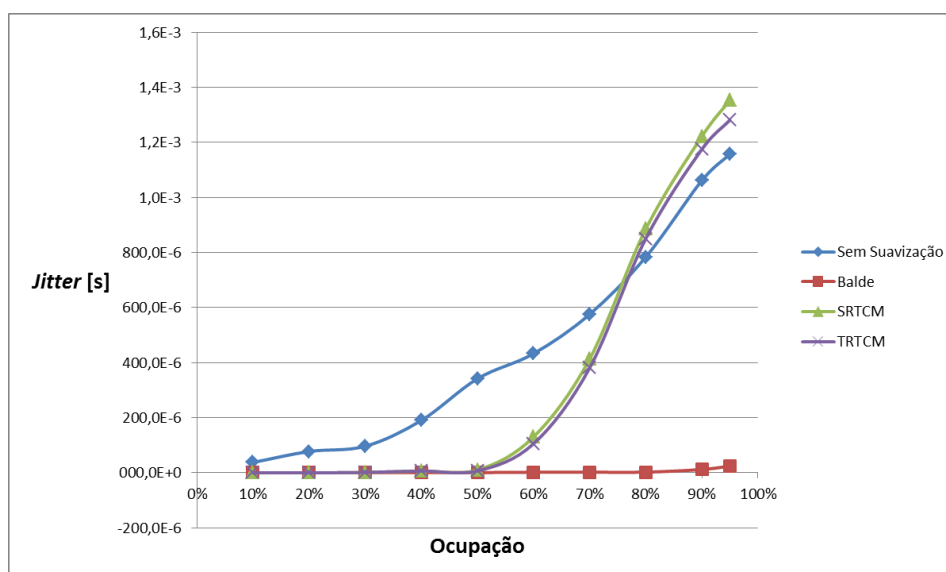


FIGURA 24 - *JITTER* CÁLCULADO COM 1% DE PERDA E BALDE IGUAL A $\Phi_1/2$

Referente aos valores simulados no Φ_1 , os valores de $\Phi_1/2$ são similares. A diferença é de que a curva do TRTCM acompanha o crescimento da curva do SRTCM.

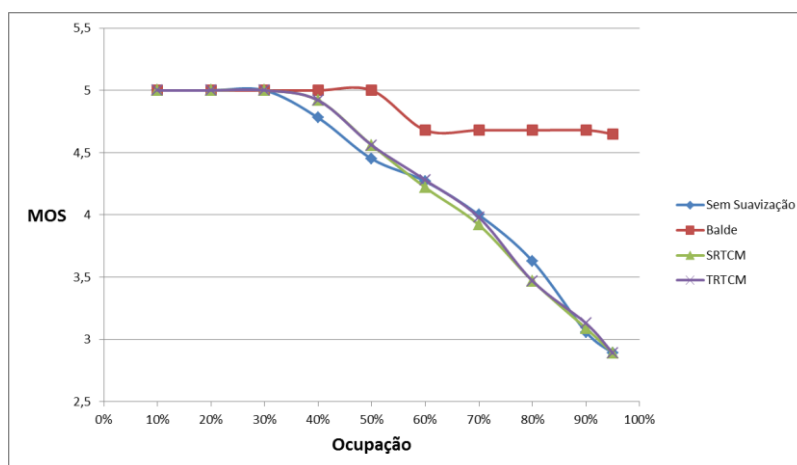


FIGURA 25 - *MOS* CÁLCULADO COM *BUFFER* INFINITO E BALDE IGUAL A Φ_1

6.2 BUFFER INFINITO

TABELA 9 - RESULTADO DO MOS A PARTIR DA SIMULAÇÃO COM *BUFFER* INFINITO

Ocupação	MOS		$b < \phi$			$b < \phi/2$		
	Sem Suavização	Balde	Balde	SRTCM	TRTCM	Balde	SRTCM	TRTCM
10%	5	5	5	5	5	5	5	5
20%	5	5	5	5	5	5	5	5
30%	5	5	5	5	5	5	5	5
40%	4,78	5	4,92	4,92	4,92	5	5	5
50%	4,45	5	4,56	4,56	4,56	5	4,98	4,98
60%	4,27	4,68	4,22	4,28	4,28	4,68	4,44	4,44
70%	4	4,68	3,92	3,98	3,98	4,68	3,93	4,26
80%	3,63	4,68	3,47	3,47	3,47	4,68	3,46	3,46
90%	3,06	4,68	3,09	3,13	3,13	4,68	2,9	2,9
95%	2,89	4,65	2,89	2,89	2,89	4,65	2,89	2,89

Para a simulação com o *buffer* infinito o Balde de Fichas também obteve a melhor resposta de simulação sem suavização. Quanto maior a ocupação do canal, mais significativos são os resultados obtidos com a utilização do Balde de Fichas. No caso do SRTCM e TRTCM, a melhora foi pouco considerável, pois em muitos casos o seu resultado foi inferior ao resultado sem suavização.

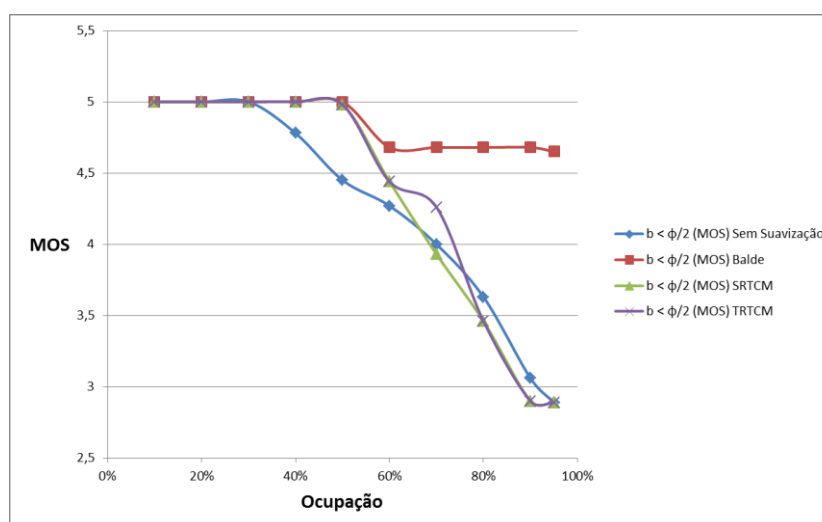


FIGURA 26 - MOS CÁLCULADO COM *BUFFER* INFINITO E BALDE IGUAL A $\Phi/2$

Os resultados da simulação do *buffer* infinito com o balde igual a $\Phi/2$ obtiveram uma melhora nos métodos do SRTCM e TRTCM, referente ao balde igual

a Φ . No caso da suavização com Balde de Fichas os valores foram exatamente os mesmos.

TABELA 10 - RESULTADO DO *DELAY* A PARTIR DA SIMULAÇÃO COM *BUFFER* INFINITO

Ocupação	<i>DELAY</i>		$b < \phi$			$b < \phi/2$		
	Sem Suavização	Balde	SRTCM	TRTCM	Balde	SRTCM	TRTCM	
10%	59,6E-3	53,4E-3	55,1E-3	54,4E-3	53,1E-3	53,3E-3	53,3E-3	
20%	69,3E-3	57,0E-3	61,3E-3	59,6E-3	56,2E-3	57,2E-3	57,2E-3	
30%	80,5E-3	60,8E-3	70,1E-3	67,5E-3	59,6E-3	62,9E-3	62,8E-3	
40%	101,5E-3	64,8E-3	85,1E-3	81,9E-3	63,3E-3	72,5E-3	71,8E-3	
50%	151,7E-3	69,3E-3	118,4E-3	114,8E-3	67,7E-3	85,0E-3	84,2E-3	
60%	259,6E-3	74,7E-3	228,7E-3	224,9E-3	73,3E-3	124,2E-3	122,6E-3	
70%	446,4E-3	81,8E-3	421,6E-3	417,7E-3	80,4E-3	263,0E-3	261,4E-3	
80%	715,3E-3	92,3E-3	706,8E-3	702,7E-3	91,1E-3	552,2E-3	550,4E-3	
90%	1,1E+0	109,7E-3	1,1E+0	1,1E+0	108,7E-3	1,0E+0	1,0E+0	
95%	1,6E+0	126,0E-3	1,6E+0	1,6E+0	125,1E-3	1,5E+0	1,5E+0	

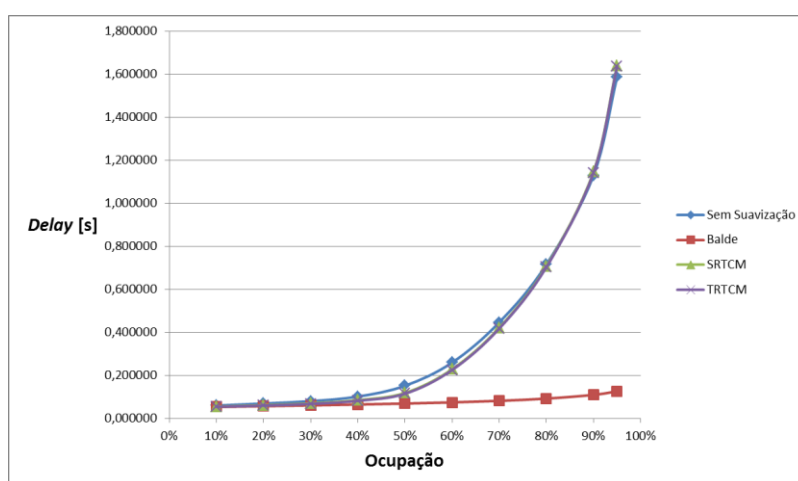


FIGURA 27 - *DELAY* CÁLCULADO COM *BUFFER* INFINITO E BALDE IGUAL A Φ

Os valores do *delay* do método de Balde de Fichas é muito inferior. Nesse caso em específico o valor dos atrasos pelos métodos do SRTCM e TRTCM são idênticos ao do simulado sem suavização.

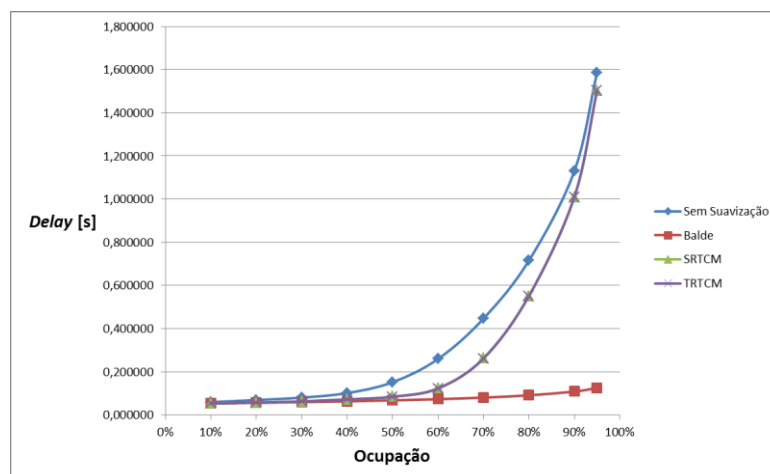


FIGURA 28 - DELAY CÁLCULADO COM *BUFFER* INFINITO E BALDE IGUAL A $\Phi/2$

No caso da simulação do atraso com o balde igual a $\Phi/2$ o método de Balde de Fichas existe um atraso muito inferior. Já referente ao SRTCM e TRTCM existe uma atraso inferior até os 90% de ocupação.

TABELA 11 - RESULTADO DO *JITTER* A PARTIR DA SIMULAÇÃO COM *BUFFER* INFINITO

<i>JITTER</i>		$b < \phi$			$b < \phi/2$		
Ocupação	Sem Suavização	Balde	SRTCM	TRTCM	Balde	SRTCM	TRTCM
10%	2,7E-6	207,4E-9	207,4E-9	207,4E-9	207,4E-9	207,4E-9	207,4E-9
20%	5,4E-6	415,0E-9	448,9E-9	448,9E-9	415,0E-9	415,0E-9	415,0E-9
30%	8,1E-6	622,6E-9	2,1E-6	1,3E-6	622,6E-9	1,5E-6	1,5E-6
40%	10,7E-6	830,3E-9	4,9E-6	2,5E-6	830,3E-9	6,8E-6	6,8E-6
50%	13,4E-6	1,0E-6	7,5E-6	5,2E-6	1,0E-6	4,3E-6	4,3E-6
60%	16,1E-6	1,2E-6	10,2E-6	7,9E-6	1,2E-6	9,0E-6	9,0E-6
70%	18,8E-6	1,5E-6	12,9E-6	10,6E-6	1,5E-3	11,9E-6	11,9E-6
80%	21,5E-6	1,7E-6	15,6E-6	13,3E-6	1,7E-6	14,6E-6	14,6E-6
90%	46,9E-6	2,5E-6	52,2E-6	49,9E-6	2,5E-6	56,4E-3	56,4E-6
95%	197,1E-6	3,4E-6	210,9E-6	208,6E-6	3,4E-6	215,1E-6	215,1E-6

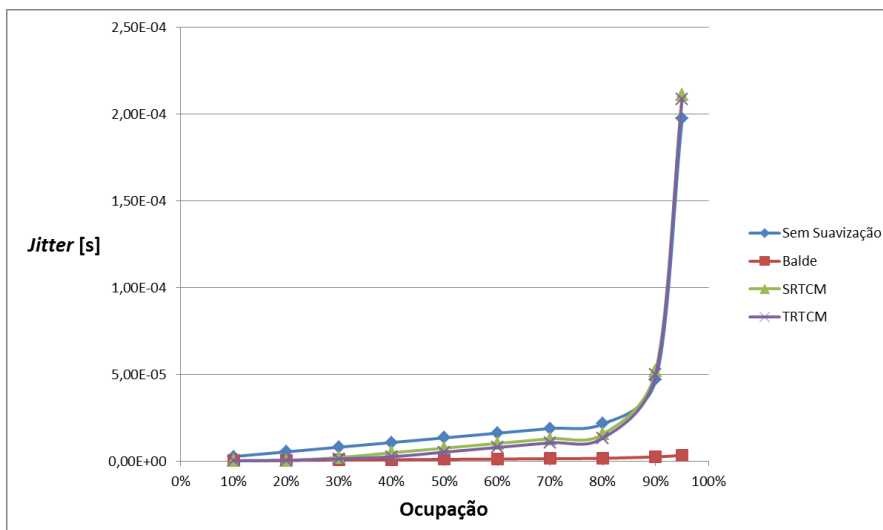


FIGURA 29 - *JITTER* CÁLCULADO COM *BUFFER* INFINITO E BALDE IGUAL A Φ_1

Para o cálculo do *jitter* o valor do Balde de Fichas foi inferior em todos os casos, tanto para o Φ_1 como para o $\Phi_1/2$. Já os valores para os métodos do SRTCM e TRTCM acompanham a curva do sem crescimento, mas com um valor inferior até os 90%.

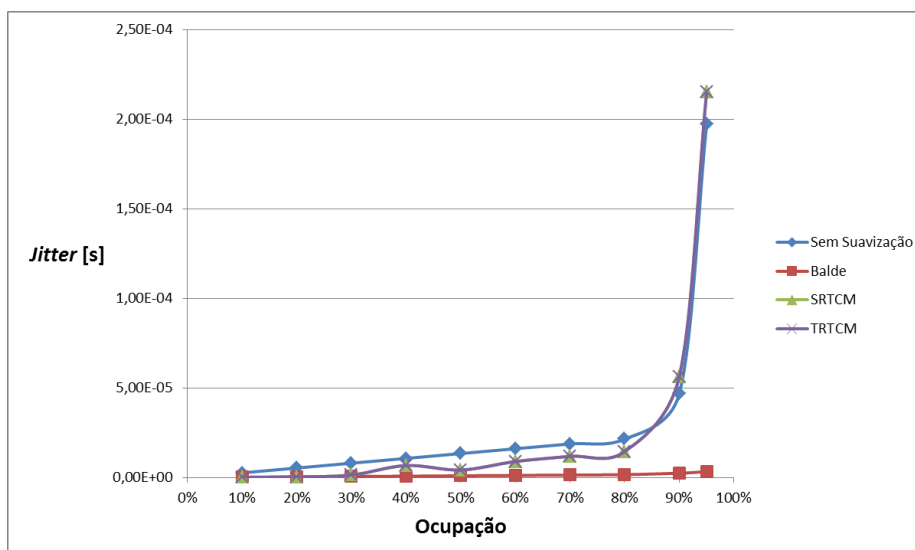


FIGURA 30 - *JITTER* CÁLCULADO COM *BUFFER* INFINITO E BALDE IGUAL A $\Phi_1/2$

7 DISCUSSÃO

Com os resultados simulados é possível analisar o comportamento do sistema em relação a cada parâmetro. O MOS naturalmente diminui com o aumento da ocupação do canal e o *jitter* e o *delay* aumentam com a ocupação do canal. Com o uso dos métodos suavizadores foi possível observar que o MOS foi o maior em comparação ao sistema que não aplica suavização de tráfego, bem como o *jitter* e o atraso médio foram reduzidos.

Para a simulação com o método suavizador do Balde de Fichas foi observada uma melhoria significativa do MOS em todos os casos simulados. Em relação ao *delay* o Balde de Fichas levou a uma redução significativa do atraso médio observado. No caso do *buffer* infinito, a suavização apresentou atraso bem inferior principalmente em valores com ocupação acima de 60%. Em relação ao *jitter* o resultado do método Balde de Fichas trouxe uma melhora considerável, principalmente para valores com mais de 50% e ocupação.

Nos casos que foram simulados com o tamanho do balde igual ao tamanho médio do quadro I, para a suavização utilizando O SRTCM e TRTCM, o valor do MOS começou a decrescer com a ocupação em 40%, em 50% e 60% o seu valor era abaixo do valor sem suavização. Acima desses valores existiu uma melhora do valor de MOS avaliado. Em alguns pontos específicos o valor do TRTCM foi superior ao valor do SRTCM simulado. Quando o valor do balde era reduzido a sua metade, os valores resultantes do MOS melhoraram, sendo que a melhoria do MOS começa a ser observada em 50% de utilização. Em relação ao *delay*, o valor do MOS observado com o uso do SRTCM e do TRTCM é menor em ocupações inferiores a 50%. Acima desse valor existe um acréscimo considerável. Com relação ao *jitter* o valor para $b < \Phi/2$ existe um acréscimo do valor para valores acima de 80%, o que não ocorre na simulação $b < \Phi_1$. Neste caso o TRTCM possui um valor inferior ao SRTCM simulado.

Desta maneira, dentre os três métodos simulados, o Balde de Fichas obteve o melhor desempenho, tanto no resultado do MOS, quanto na sua característica em relação ao *delay* e ao *jitter* da transmissão do vídeo no canal. Possivelmente isso se deve ao uso consecutivo de dois baldes de fichas, o que permite que mais rajadas sejam transmitidas na rede.

7 CONCLUSÕES E TRABALHOS FUTUROS

Após a análise dos resultados o método suavizador que trouxe uma melhora considerável é o método do Balde de Fichas, sendo observado significativa melhora no MOS e uma melhora considerável no *delay* e *jitter*. No caso do TRTCM e SRTCM ocorreu uma melhora no MOS observado com desempenho melhor do algoritmo quando o tamanho do balde foi configurado com tamanho igual a metade do tamanho médio do quadro I.

Desta forma, os testes realizados indicam que é possível obter uma melhora no MOS aplicando métodos de suavização ao tráfego de vídeo. Também foi sugerido um método para configuração dos parâmetros do algoritmo do Balde de Fichas, relacionando r e b com o tamanho médio do quadro I e com a taxa de transmissão de quadros do vídeo.

Como trabalhos futuros, será necessário analisar uma quantidade maior de vídeos, com diferentes configurações de qualidade, geometria e GOP, para comprovar os resultados obtidos. Também é interessante realizar a transmissão do tráfego do vídeo agregado, com e sem suavização, para observar os efeitos da aplicação do método nestes casos.

REFERÊNCIAS

- ASU, Arizona State University. **Video Traces**. Disponível em: <<http://trace.eas.asu.edu>>. Acesso em: 08/09/2012
- DSL Forum, **Triple-play Services Quality of Experience Requirements**. [2006] Disponível em: <<http://www.broadband-forum.org/technical/download/TR-126.pdf>> Acesso em: 09/03/2013
- IETF. **The Internet Engineering Task Force**. Disponível em: <<http://tools.ietf.org/html/rfc768>> Acesso em: 16/01/2013
- IETF. **The Internet Engineering Task Force**. Disponível em: <<http://tools.ietf.org/html/rfc3550>> Acesso em: 16/01/2013
- Information Sciences Institute. **NS**. [Califórnia, 2002]. Disponível em: <<http://www.isi.edu/nsnam/ns/>> Acesso em: 01/11/2012
- International Telecommunication Union, **Methods for Objective and Subjective Assessment of Quality**. 1996
- ISNARDI, M. **MPEG-2 Video Compression**, Nova Jersey, 1999
- ISO. **International Organisation for Standardisation** [2010] Disponível em: <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=56538> Acesso em: 10/12/2012
- KLAUE, J. **Manual Evalvid**. [Berlim, 2003]. Disponível em: <<http://www2.tkn.tu-berlin.de/research/evalvid/EvalVid/docevalvid.html>> Acesso em: 10/12/2012
- KLAUE, J. **Manual Evalvid**. [Berlim, 2003]. Disponível em: <http://www.tkn.tu-berlin.de/fileadmin/fg112/Hard_Software_Components/Software/evalvid.pdf > Acesso em: 10/12/2012
- PEDROSO, C. M. **QoS**. [Curitiba, 2009]. Disponível em: <<http://www.eletrica.ufpr.br/pedroso/2009/TE090/Aulas/QoS.pdf> qos> Acesso em: 21/11/2012
- POSTEL, J. **User Datagram Protocol**. [28/08/1980] Disponível em: <<http://www.ietf.org/rfc/rfc768.txt> > Acesso em: 21/11/2012
- RICHARDSON, I. E. G. **H 264 And Mpeg-4 Video Compression - Video Coding For Next Generation Multimedia**. Aberdeen, 2003
- Royal Pingdom. **Internet 2012 in numbers**. [Suécia, 2013] Disponível em: <<http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/>> Acesso em: 17/01/2013
- SANTOS, A. P. S. **Qualidade de Serviço na Internet**. [Rio de Janeiro, 1999]. Disponível em: <<http://www.rnp.br/newsgen/9911/qos.html>> Acesso em: 21/11/2012

APÊNDICE

TRANSFORMA – CRIAÇÃO DO ARQUIVO HEXADECIMAL PARA O NS2

```

#include <stdio.h>
#include <stdlib.h>
/* reverseInt é necessário para transformar o formato para Little Indian para gravação */
int reverseInt (int i) {
    unsigned char c1, c2, c3, c4;

    c1 = i & 255;
    c2 = (i >> 8) & 255;
    c3 = (i >> 16) & 255;
    c4 = (i >> 24) & 255;

    return ((int)c1 << 24) + ((int)c2 << 16) + ((int)c3 << 8) + c4;
}

int main(int argc, char *argv[])
{
    int i;
    unsigned int size, q, n, intervalo;
    unsigned char t;
    float time, novotime, old=0;
    FILE *arqe, *arqs;
    if (argc!=3) {
        printf("\nUso: %s <nome-arquivo-entrada> <nome-arquivo-saida>\n\n", argv[0]);
        exit(0);
    }
    arqe=fopen(argv[1], "r");
    if (arqe==NULL) {printf("\nErro de abertura %s\n", argv[1]); exit(0);}

    arqs=fopen(argv[2], "w");
    if (arqs==NULL) {printf("\nErro de abertura %s\n", argv[2]); exit(0);}

    while(!feof(arqe))
    {
        fscanf(arqe,"%d %c %d %d %f", &n, &t, &size, &q, &time);

    if (feof(arqe))
        break;

    novotime=time;
    time = time - old;
    intervalo = time * 1000000.0;
    printf("%d %c %d %d %f %d\n", n, t, size, q, time, intervalo);

    intervalo=reverseInt(intervalo);
    size=reverseInt(size);
    fwrite(&intervalo, sizeof(int), 1, arqs);
    fwrite(&size, sizeof(int), 1,arqs);
    old=novotime;
    }
    fclose(arqe);
    fclose(arqs);
}

```


BALDE DE FICHAS

```

#include <stdio.h>
#include <stdlib.h>
/* reverseInt é necessário para transformar o formato para Little Indian para gravação */
int reverseInt (int i)
{
    unsigned char c1, c2, c3, c4;
    c1 = i & 255;
    c2 = (i >> 8) & 255;
    c3 = (i >> 16) & 255;
    c4 = (i >> 24) & 255;
    return ((int)c1 << 24) + ((int)c2 << 16) + ((int)c3 << 8) + c4;
}
int main(int argc, char *argv[])
{
    int i;
    unsigned int size, q, n, intervalo;
    unsigned char t;
    float time, global_time, old=0;
    float r, bmax, b;
    FILE *arqe, *arqs;
    if (argc!=5)
    {
        printf("\nUso: %s <nome-arquivo-entrada> <nome-arquivo-saida> <r> <b>\n\n", argv[0]);
        exit(0);
    }
    arqe=fopen(argv[1], "r");
    if (arqe==NULL) {printf("\nErro de abertura %s\n", argv[1]); exit(0);}
    arqs=fopen(argv[2], "w");
    if (arqs==NULL) {printf("\nErro de abertura %s\n", argv[2]); exit(0);}
    r=atof(argv[3]);
    bmax=atof(argv[4]);
    b=bmax;
    while(!feof(arqe))
    {
        fscanf(arqe,"%d %c %d %d %f", &n, &t, &size, &q, &time);
        if (time > global_time)
            global_time=time;
        else
            global_time=time=old;
        printf ("\n -----\n Tempo global=%f",global_time);
        // atualização do numero de fichas no balde
        b = b + r*(global_time-old);
        printf("\n VAlor de b será atualizado b = %f, globaltime= %f, old= %f",b, global_time, old);
        if (b>bmax) b=bmax;
        if (b<size) // se nao existem fichas no balde o pacote espera até que tenham fichas no balde
        {
            printf("\nNão tem fichas no balde ... b=%f recalculando globaltime", b);
            global_time=time=global_time+(size-b)/r;
            b=size;
            printf("\nsize=%d, b=%f, r=%f, globaltime=%f",size, b, r, global_time);
        }
        b=b-size;
        printf("\natualizando o valor de b=%f",b);
        time = time - old;
        intervalo = time * 1000000.0;
        printf("\n%d %c %d %d %f %f\n", n, t, size, q, global_time, b);
        intervalo=reverseInt(intervalo);
        size=reverseInt(size);
        fwrite(&intervalo, sizeof(int), 1, arqs);
    }
}

```

```

        fwrite(&size, sizeof(int), 1,arqs);
        old=global_time;
    }
    fclose(arqe);
    fclose(arqs);
}

```

SRTCM

```

#include <stdio.h>
#include <stdlib.h>
/* reverseInt é necessário para transformar o trace para o formato de gravação para a transmissão no
NS2 */
int reverseInt (int i)
{
    unsigned char c1, c2, c3, c4;
    c1 = i & 255;
    c2 = (i >> 8) & 255;
    c3 = (i >> 16) & 255;
    c4 = (i >> 24) & 255;
    return ((int)c1 << 24) + ((int)c2 << 16) + ((int)c3 << 8) + c4;
}
int main(int argc, char *argv[])
{
    int i;
    unsigned int size, q, n, intervalo;
    unsigned char tipo;
    float time, global_time, old=0;
    float rc, bemax, be, bcmax, bc;
    int vm=0, vd=0, am=0;
    FILE *arqe, *arqs;
    if (argc!=6) {
        printf("\nUso: %s <nome-arquivo-entrada> <nome-arquivo-saida> <rc> <bc> <be>\n\n",
argv[0]);
        exit(0);
    }
    arqe=fopen(argv[1], "r");
    if (arqe==NULL) {printf("\nErro de abertura %s\n", argv[1]); exit(0);}
    arqs=fopen(argv[2], "w");
    if (arqs==NULL) {printf("\nErro de abertura %s\n", argv[2]); exit(0);}
    rc=atof(argv[3]);
    bcmax=atof(argv[4]);
    bemax=atof(argv[5]);
    bc=bcmax;
    be=bemax;
    while(!feof(arqe))
    {
        fscanf(arqe,"%d %c %d %d %f", &n, &tipo, &size, &q, &time);
        if (feof(arqe))
            break;
        if (time > global_time)
            global_time=time;
        else
            global_time=time=old;
        bc = bc + rc*(global_time-old);
        if (bc>bcmax)
        {
            be=be+bc-bcmax;
            if (be>bemax)
                be=bemax;
        }
    }
}

```

```

        bc=bcmax;
    }
    printf("\nValor Atualizado dos Baldes: Bc=%f e Be=%f", bc, be);
    if (size>bc) // se nao existem fichas no balde o pacote espera até que tenham fichas
no balde
    {
        if (size>be)
        {
            //Pacote VERMELHO (Não decrementa)
            printf("\nPacote VERMELHO");
            printf("\nPacote Vermelho ... recalculando globaltime");
            global_time=time=global_time+(size-bc)/rc;
            printf("\nglobaltime=%f",global_time);
            bc=size;
            vm++;
        }
        else
        {
            //Pacote AMARELO (Decrementa de Be)
            printf("\nPacote AMARELO");
            be=be-size;
            printf("\nglobaltime=%f",global_time);
            am++;
        }
    }
    else
    {
        //Pacote VERDE (Decrementa de Bc)
        printf("\nPacote VERDE");
        bc=bc-size;
        printf("\nglobaltime=%f",global_time);
        vd++;
    }
    printf("\n%d %c %d %d %f %f %f\n", n, tipo, size, q, global_time, bc, be);
    time = time - old;
    intervalo = time * 1000000.0;
    printf("vd=%d, am=%d, vm=%d\n", vd, am, vm);
    intervalo=reverselnt(intervalo);
    size=reverselnt(size);
    fwrite(&intervalo, sizeof(int), 1, arqs);
    fwrite(&size, sizeof(int), 1,arqs);
    old=global_time;
}
fclose(arqe);
fclose(arqs);
}

```

TRTCM

```

#include <stdio.h>
#include <stdlib.h>
/* reverselnt é necessário para transformar o trace para o formato de gravação para a transmissão no
NS2 */
int reverselnt (int i)
{
    unsigned char c1, c2, c3, c4;
    c1 = i & 255;
    c2 = (i >> 8) & 255;
    c3 = (i >> 16) & 255;
    c4 = (i >> 24) & 255;
    return ((int)c1 << 24) + ((int)c2 << 16) + ((int)c3 << 8) + c4;
}

```

```

}
int main(int argc, char *argv[])
{
    int i;
    unsigned int size, q, n, intervalo;
    unsigned char t;
    float time, global_time, old=0;
    float re, bemax, be, rc, bcmax, bc;
    int vm=0, vd=0, am=0;
    FILE *arqe, *arqs;
    if (argc!=7)
    {
        printf("\nUso: %s <nome-arquivo-entrada> <nome-arquivo-saida> <rc> <re> <bc> <be>\n\n",
argv[0]);
        exit(0);
    }
    arqe=fopen(argv[1], "r");
    if (arqe==NULL) {printf("\nErro de abertura %s\n", argv[1]); exit(0);}
    arqs=fopen(argv[2], "w");
    if (arqs==NULL) {printf("\nErro de abertura %s\n", argv[2]); exit(0);}
    rc=atof(argv[3]);
    re=atof(argv[4]);
    bcmax=atof(argv[5]);
    bemax=atof(argv[6]);
    bc=bcmax;
    be=bemax;
    while(!feof(arqe))
    {
        fscanf(arqe,"%d %c %d %d %f", &n, &t, &size, &q, &time);
        if (feof(arqe))
            break;
        if (time > global_time)
            global_time=time;
        else
            global_time=time=old;
        bc = bc + rc*(global_time-old);
        be = be + re*(global_time-old);
        if (bc>bcmax)
            bc=bcmax;
        if (be>bemax)
            be=bemax;
        printf("Valor Atualizado dos Baldes: Bc=%f e Be=%f", bc, be);
        if (size>bc)
        {
            //Pacote VERMELHO (NÃO DECREMENTA)
            // se nao existem fichas no balde o pacote espera até que tenham fichas no
            balde
            printf("\nPacote VERMELHO");
            printf("\nPacote Vermelho ... recalculando globaltime");
            global_time=time=global_time+(size-bc)/rc;
            printf("\nglobaltime=%f", global_time);
            be=be+re*((size-bc)/rc);
            bc=size;
            vm++;
        }
        else{
            if(size>be){
                //Pacote AMARELO (DECREMENTA DE BC)
                printf("\nPacote AMARELO");
            }
        }
    }
}

```

```

        printf("\nglobaltime=%f",global_time);
        am++;
        bc=bc-size;
    }
    else{
        //Pacote VERDE (DECREMENTA DOS DOIS BALDES)
        printf("\nPacote VERDE");
        printf("\nglobaltime=%f",global_time);
        vd++;
        be=be-size;
        bc=bc-size;
    }}
    printf("\n%d %c %d %d %f %f %f\n", n, t, size, q, global_time, bc, be);
    time = time - old;
    intervalo = time * 1000000.0;
    printf("vd=%d, am=%d, vm=%d\n\n", vd, am, vm);
    intervalo=reverseInt(intervalo);
    size=reverseInt(size);

    fwrite(&intervalo, sizeof(int), 1, arqs);
    fwrite(&size, sizeof(int), 1,arqs);
    old=global_time;
}
    fclose(arqe);
    fclose(arqs);
}

```

CRIAÇÃO DE ARQUIVOS DE TRACE

```

#!/bin/bash
#-----#
# Shell Script para Criar arquivo de Trace para o NS2
#-----#
#-----#
#./ffmpeg -r 30 -b 64000 -bt 3200 -g 30 -i $video.yuv -vcodec mpeg4 -bf 2 -sameq
$video.m4v -an
video=trace1 #NOME DO VÍDEO
caminho_video=/home/danielmklassen/Documentos/TCC
caminho_prog=/home/danielmklassen/Documentos/TCC/Programas
#-----#
cd Traces_Originais
cp $video.mp4 $caminho_video
cd ..
$caminho_prog/ffmpeg -i $video.mp4 $video_"original.yuv
$caminho_prog/ffmpeg -s 720x480 -r 30 -b 64000 -bt 3200 -g 12 -i
$video_"original.yuv -vcodec mpeg4 -bf 2 -sameq $video_"ref.m4v
$caminho_prog/MP4Box -hint -mtu 1500 -fps 30 -add $video_"ref.m4v
$video_"ref.mp4
$caminho_prog/ffmpeg -i $video_"ref.mp4 $video_"ref.yuv
$caminho_prog/psnr 720 480 420 $video_"original.yuv $video_"ref.yuv >
psnr_"ref_"$video.txt
$caminho_prog/mp4trace -f -s 192.168.0.2 12346 $video_"ref.mp4 > trace_"$video
$caminho_prog/transforma trace_"$video $video_"NS2
./token-bucket-suaviza trace_"$video $video_"balde 354615 38692

```

```
./srtcm trace"_"$video $video"_"srtcm 354615 19346 19346
./trtcm trace"_"$video $video"_"trtcm 354615 709230 19346 19346
mkdir $video
mv $video.mp4 $video"_"original.yuv $video"_"ref.m4v $video"_"ref.mp4
$video"_"ref.yuv psnr"_"ref"_"$video.txt trace"_"$video $video"_"NS2
$caminho_video/$video
```

SIMULAÇÃO

```
#!/bin/bash
# Shell Script para Rodar a Avaliação da Simulação do Video no NS2 #
#-----#
# Aluno: Daniel Mauricio Klassen
# Trabalho de Conclusão de Curso
#
# Implementação de um Suavizador de Tráfego para Melhoria da Qualidade de
# Experiência em Sistemas IPTV
#
# UNIVERSIDADE FEDERAL DO PARANÁ
# Orientador: Carlos Marcelo Pedroso
#-----#
caminho_video=/home/danielmklassen/Documentos/TCC
caminho_prog=/home/danielmklassen/Documentos/TCC/Programas
#-----#
video=trace1 #Nome do Vídeo Transmitido
NS=balde_b2
#-----#
#-----#
ocup=10 # Porcentagem de Ocupação do Canal
#-----#
#-----#
# Simulação no NS2
cd $caminho_video/$video
cp $NS $caminho_video
cd ..
ns NS2.tcl
rm -R $NS
awk '$12 >= 0 && $12 <= 7788' out.tr > out.txt #Filtra as informações do video
#-----#
#-----#
# Cálculo Delay e Jitter
cd $caminho_prog
cp pacotes_perdidos.awk $caminho_video
cp delay.awk $caminho_video
cp jitter.awk $caminho_video
cp pacotes.awk $caminho_video
cp perdidos.awk $caminho_video
cd ..
gawk -f pacotes_perdidos.awk out.txt > pacotes_perdidos.txt
gawk -f delay.awk out.txt > delay.txt
```

```

gawk -f jitter.awk out.txt > jitter.txt
#-----
#-----
#Cálculo TX id
awk '$1 == "r" && $3 == 3 && $4 == 1 && $8 == 1' out.txt > tx.txt
awk '{print $2"\t","id "$12"\t",$5,"\t"$6}' tx.txt > tx
#Cálculo RX id
awk '$1 == "r" && $3 == 0 && $4 == 2 && $8 == 1' out.txt > rx.txt
awk '{print $2"\t","id "$12"\t",$5,"\t"$6}' rx.txt > rx
#-----
#-----
awk '$1 == "d" && $12 >= 0 && S12 <= 7788' out.txt > dropados.txt
awk '{print "id "$11}' dropados.txt > indrop.txt
awk '{print $2"\t","id "$11"\t",$5,"\t"$6,"\t"$12}' tx.txt > txid ;
awk '{print $2"\t","id "$11"\t",$5,"\t"$6,"\t"$12}' rx.txt > rxid ;
awk '{$3=int($3)}1' delay.txt > inddelay.txt
awk '{ print $2, $3 }' inddelay.txt > delayid.txt
awk 'NR==FNR{a[$2]=$1;next}$6 in a{print $0,a[$6]}' delayid.txt rxid > rxcomdelay.txt
awk '$7 < 0.200 ' rxcomdelay.txt > rx2
awk '$7 > 0.200 ' rxcomdelay.txt > rx_excluidos
awk '{print $1"\t","id "$3"\t",$4,"\t"$5}' rx2 > rx_final
awk '{sum+=$2} END { print "delay = ",sum/NR}' delay.txt > Delay.txt
awk '{sum+=$2} END { print "jitter = ",sum/NR}' jitter.txt > Jitter.txt
gawk -f pacotes.awk out.txt > pacotes_perdidos.txt
wc -l rx_excluidos > excluidos.txt
awk '{print $1}' excluidos.txt > teste.txt
cat teste.txt pacotes_perdidos.txt > cat.txt
cat cat.txt | awk '{ sum+=$1} END {print sum}' > pacotes_perdidos_total.txt
gawk -f perdidos.awk out.txt > pacotes_total.txt
cat pacotes_total.txt pacotes_perdidos_total.txt > test.txt
awk '{printf("%s ", $0)}' test.txt > tes.txt
awk '{print "\n","Total de Pacotes = "$1"\n","Pacotes Perdidos = "$2"\n","Porcentagem = "$2/$1*100"\n"}' tes.txt > Pacotes_perdidos_final.txt
cat Delay.txt >> Pacotes_perdidos_final.txt
cat Jitter.txt >> Pacotes_perdidos_final.txt
rm -R rx.txt tx.txt cat.txt delay.txt delayid.txt Delay.txt dropados.txt excluidos.txt
inddelay.txt indrop.txt jitter.txt Jitter.txt pacotes_perdidos.txt
pacotes_perdidos_total.txt pacotes_total.txt txid rx rx2 rxcomdelay.txt rx_excluidos
rxid test.txt teste.txt pacotes.awk perdidos.awk pacotes_perdidos.awk delay.awk
jitter.awk out.tr tes.txt
#-----
#-----
cd $video
cp $video_"ref.mp4 $caminho_video
cp trace_"$video $caminho_video
cd ..
$caminho_prog/etmp4 -f -0 tx rx_final trace_"$video $video_"ref.mp4
$video_"sender
rm -R $video_"sender.m4v
rm -R $video_"ref.mp4

```

```

rm -R trace"_"$video
$caminho_prog/ffmpeg -i $video"_"sender.mp4 $video"_"sender.yuv
mkdir work
mkdir arquivos
mv out.txt rate_r"_"$video"_"sender.txt rate_s"_"$video"_"sender.txt rx_final tx
delay"_"$video"_"sender.txt loss"_"$video"_"sender.txt $caminho_video/arquivos
rm -R out.nam
echo Copiando arquivo yuv ...
cd $video
cp $video"_"original.yuv $caminho_video
cd ..
echo Calculando PSNR ...
$caminho_prog/psnr 720 480 420 $video"_"original.yuv $video"_"sender.yuv >
psnr"_"sender"_"$video.txt
rm -R $video"_"original.yuv
rm -R $video"_"sender.yuv
cd $caminho_video/$video
cp psnr"_"ref"_"$video.txt $caminho_video/work
cp psnr"_"ref"_"$video.txt $caminho_video
cd ..
cp psnr"_"sender"_"$video.txt $caminho_video/work
rm -R psnr"_"sender"_"$video.txt
$caminho_prog/mos $caminho_video/work psnr"_"ref"_"$video.txt 30 > mos.txt
rm -R psnr"_"ref"_"$video.txt
#rm -R trace"_"$video
mkdir Resultado
mv mos.txt $caminho_video/Resultado
mv Pacotes_perdidos_final.txt $caminho_video/Resultado
mv $video"_"sender.mp4 $caminho_video/Resultado
cd $caminho_video/arquivos
awk '{print "Quadros Perdidos:\n"l = "$1"\nP = "$2"\nB = "$3"\nMédia = "$4}'
loss"_"$video"_"sender.txt > loss.txt
mv loss.txt $caminho_video/Resultado
rm -R loss"_"$video"_"sender.txt delay"_"$video"_"sender.txt
cd ..
mkdir $video"_"$ocup"%"
mv $caminho_video/arquivos $caminho_video/$video"_"$ocup"%"
mv $caminho_video/Resultado $caminho_video/$video"_"$ocup"%"
mv $caminho_video/work $caminho_video/$video"_"$ocup"%"
cd $video"_"$ocup"%"
cd Resultado
cat loss.txt >> Pacotes_perdidos_final.txt
cat Pacotes_perdidos_final.txt >> mos.txt
awk '{ print } NR == 7 { print "" }' mos.txt > teste.txt
awk '{ print } NR == 10 { print "" }' teste.txt > MOS.txt
rm -R loss.txt Pacotes_perdidos_final.txt teste.txt mos.txt
cd ..
cd ..
#-----

```


CÁLCULO DO JITTER

```

BEGIN { highest_packet_id=0; }
{
action = $1;
time = $2;
from = $3;
to = $4;
type = $5;
pktsize = $6;
flow_id = $8;
src = $9;
dst = $10;
seq_no = $11;
packet_id = $12;
if ( packet_id > highest_packet_id )
highest_packet_id = packet_id;
if ( start_time[packet_id] == 0 )
{ pkt_seqno[packet_id] = seq_no;
start_time[packet_id] = time; }
if ( flow_id == 1 && action != "d" )
{ if( action == "r" ) { end_time[packet_id] = time; }
else { end_time[packet_id] = -1; }
}
}
}
END {
last_seqno = 0;
last_delay = 0;
seqno_diff = 0;
for(packet_id=0; packet_id < highest_packet_id; packet_id++)
{
start = start_time[packet_id];
end = end_time[packet_id];
packet_duration = end-start;
if( start < end )
{
seqno_diff = pkt_seqno[packet_id]-last_seqno;
delay_diff = packet_duration - last_delay;
if ( seqno_diff == 0 )
{ jitter = 0; }
else { jitter = delay_diff/seqno_diff; }
printf("%f %f\n",start, jitter);
last_seqno = pkt_seqno[packet_id];
last_delay = packet_duration;
}}}

```

CÁLCULO DO DELAY

```

BEGIN { highest_packet_id=0; }
{
action = $1;
time = $2;

```

```

from = $3;
to = $4;
type = $5;
pktsize = $6;
flow_id = $8;
src = $9;
dst = $10;
seq_no = $11;
packet_id = $12;
if ( packet_id > highest_packet_id )
highest_packet_id = packet_id;
if ( start_time[packet_id] == 0 )
start_time[packet_id] = time;
if ( flow_id == 1 && action != "d" )
{
if( action == "r" )
{ end_time[packet_id] = time; }
else
{ end_time[packet_id] = -1; }
}
}
END {
#for(packet_id=0; packet_id < highest_packet_id; packet_id++)
for(packet_id=0; packet_id <= 7788; packet_id++)
{
#if (action == "r" && from == 3 && to == 1)
start = start_time[packet_id];
#if (action == "r" && from == 1 && to == 0)
end = end_time[packet_id];
packet_duration = end-start;
if( start < end )
{
printf("%f %f %f\n",start, packet_duration, packet_id);
}}}

```

CÁLCULO DE PERDA DE PACOTES

```

BEGIN {
# Initialization. Set two variables. fsDrops: packets drop. numFs: packets sent
    fsDrops = 0;
    numFs = 0;
    Percentual = 0;
}
{
    action = $1;
    time = $2;
    from = $3;
    to = $4;
    type = $5;
    pktsize = $6;
    flow_id = $8;

```

```

src = $9;
dst = $10;
seq_no = $11;
packet_id = $12;
  if (from==1 && to==0 && action == "+")
  {
    if (packet_id > 0 && packet_id <= 7788)
      numFs++;
  }
  if (flow_id==1 && action == "d") #from==1 && to==0 &&
  {
    if (packet_id > 0 && packet_id <= 7788)
      fsDrops++;
  }
}
END {
  Percentual = 100*fsDrops/numFs;
  printf("\nPacotes Enviados = %d\nPacotes Perdidos = %d\nPercentual =
%f\n\n", numFs, fsDrops, Percentual);
}

```

CÓDIGO DE SIMULAÇÃO DO NS2

Programa para uso de transmissão de arquivo de trace NS2

22/03/2013

=====

Aluno: Daniel Mauricio Klassen

Orientador: Carlos Marcelo Pedroso

Universidade Federal do Paraná

Departamento de Engenharia Elétrica

Centro Politécnico

CEP: 81531-990 - Curitiba - PR - Brasil

set stoptime 100.0

set plottime 101.0

set ns [new Simulator]

Nós n0 e n1 são roteadores

set n0 [\$ns node]

set n1 [\$ns node]

Nó n2 a n5 são clientes

set n2 [\$ns node]

set n3 [\$ns node]

O nó nserveridor representa o servidor

set nserveridor [\$ns node]

set nserveridorftp [\$ns node]

set f [open out.tr w]

\$ns trace-all \$f

set nf [open out.nam w]

\$ns namtrace-all \$nf

Para diferenciar na animação

\$ns color 1 Blue

Link entre n0 e n1 é o bottleneck

```

$ns duplex-link $n0 $n1 28.369Mb 50ms DropTail
# Os clientes estão do lado do n0
$ns duplex-link $n0 $n2 1000Mb 0.001ms DropTail
$ns duplex-link $n0 $n3 1000Mb 0.001ms DropTail
# O servidor está do outro lado
$ns duplex-link $n1 $nservidor 1000Mb 0.001ms DropTail
$ns duplex-link $n1 $nservidorftp 1000Mb 0.001ms DropTail
$ns queue-limit $nservidor $n1 1000
#Orientação para Animação
$ns duplex-link-op $n1 $nservidor orient left
$ns duplex-link-op $n0 $n1 orient left
$ns duplex-link-op $n0 $n2 orient right
$ns duplex-link-op $n0 $n1 queuePos 0.5
$ns queue-limit $n0 $n1 100000
$ns queue-limit $n1 $n0 100000
$ns queue-limit $n0 $n2 100000
$ns queue-limit $n2 $n0 100000
$ns queue-limit $n0 $n3 100000
$ns queue-limit $n3 $n0 100000
$ns queue-limit $nservidor $n1 100000
$ns queue-limit $n1 $nservidor 100000
$ns queue-limit $nservidorftp $n1 100000
$ns queue-limit $n1 $nservidorftp 100000
#####
## Trace files
set tfile1 [new Tracefile]
$tfile1 filename trace1_NS2
#####
## Fonte 1
set s1 [new Agent/UDP]
$ns attach-agent $nservidor $s1
$s1 set fid_ 1
$s1 set packetSize_ 1500
set null1 [new Agent/Null]
$ns attach-agent $n2 $null1
$ns connect $s1 $null1
set trace1 [new Application/Traffic/Trace]
$trace1 attach-tracefile $tfile1
$trace1 attach-agent $s1
#####
## Background traffic
#Setup a TCP connection
#
set agent_tcp [new Agent/TCP]
#Attach TCP Agent to source node nservidorftp
$ns attach-agent $nservidorftp $agent_tcp
set agent_sink [new Agent/TCPSink]
#Attach a TCPSink Agent to destination node n3
$ns attach-agent $n3 $agent_sink
#Connect TCP Agent with TCPSink Agent

```

```
$ns connect $agent_tcp $agent_sink
#Flow Identity for TCP
$agent_tcp set fid_ 2
#Setup a FTP traffic over TCP connection
set traf_ftp [new Application/FTP]
$traf_ftp attach-agent $agent_tcp
#####
# Agenda tarefas
$ns at 0.1 "$trace1 start"
$ns at 0.1 "$traf_ftp start"
$ns at $stoptime "$trace1 stop"
$ns at $stoptime "$traf_ftp stop"
$ns at $plottime "close $f"
$ns at $plottime "finish tg"
proc finish file {
    puts "running nam..."
}
$ns run
exit
```