



## Testbed para análise de QoS/QoV sobre redes ip

Carlos Soares Noletto<sup>1</sup>, Jean Nunes Ribeiro Araujo<sup>2</sup> e Claudio de Castro Monteiro<sup>3</sup>

<sup>1</sup>Pós Graduando do Programa de Pós-Graduação em Telemática do Instituto Federal do Tocantins. e-mail: carlosnoletto@gmail.com

<sup>2</sup>Pós Graduando do Programa de Pós-Graduação em Telemática do Instituto Federal do Tocantins. e-mail: jean.to@gmail.com

<sup>3</sup>Prof. Doutor do Instituto Federal do Tocantins. e-mail: ccm@ifto.edu.br

**Resumo:** O aumento da demanda de serviços multimídia sobre redes de computadores tem despertado a atenção da comunidade de pesquisa nos últimos anos. As transmissões de conteúdo utilizando áudio e vídeo vêm ganhando espaço nas universidades, empresas, sistemas de ensino a distância, governo, além de alcançar a todos que possuem acesso a Internet. A grande aceitação e, muitas vezes, a preferência dos usuários, evidencia o crescimento do consumo deste tipo de conteúdo na busca por entretenimento e educação. No entanto, as transmissões multimídia acrescentam maior grau de sobrecarga em uma rede, principalmente quando se trata das populares redes WLAN, devido à maior probabilidade de interferências. Esta realidade justifica as diversas iniciativas de estudos e pesquisas que procuram desenvolver e propor melhorias em termos de qualidade de serviço e qualidade de vídeo nas transmissões de tráfego multimídia em redes IP. Geralmente, as pesquisas possuem características experimentais, onde os pesquisadores desenvolvem rotinas próprias a fim de otimizar as coletas de dados que possam ser usados em suas propostas. Porém, muito do que se faz em um ambiente de transmissão multimídia, caracteriza-se por tarefas repetitivas e manuais, principalmente nas etapas de transmissão de conteúdo, coleta, sintetização e análise dos dados. Este trabalho apresenta uma proposta de testbed reutilizável, que automatiza as tarefas repetitivas para testes desta natureza, proporcionando ao pesquisador maior dedicação a proposta da pesquisa e a análise dos dados coletados durante as seqüências de transmissão.

**Palavras-chave:** testbed, multimídia, QoS, QoV.

## 1. INTRODUÇÃO

Existem inúmeras pesquisas relacionadas à qualidade de vídeo em redes de computadores, por conseguinte, diversos artigos vêm sendo publicados apresentando contribuições para esta ampla e promissora área de conhecimento. Neste contexto, é natural que haja diversos caminhos que se possa seguir a fim de propor melhorias à comunidade (Migliorini, 2011) (Kim, 2011) (Mingardi, 2009).

Em relação à transmissão de conteúdo multimídia, algumas contribuições que tem sido propostas utilizam-se das redes sem fio WLAN devido à mobilidade proporcionada. Acessar um vídeo, disponível na Internet ou em qualquer ambiente compartilhado, sem necessidade de cabos e conectores contribui, sem dúvida, para um satisfatório aumento da conectividade dos usuários. Porém, essa tendência não reside apenas na pretensão de unir o útil ao agradável, mas também em viabilizar, na prática, esta realidade. É conhecido que o tráfego de áudio e vídeo, dependendo do estado da rede, adiciona maior sobrecarga no meio, o que pode provocar perda de pacotes e prejuízo na qualidade de visualização da mídia no dispositivo cliente.

O presente artigo apresenta um *testbed* eficiente que padroniza as rotinas de transmissão de *streams* de vídeo e de coleta de dados em ambientes de rede. A proposta é otimizar os processos de requisição de conteúdo multimídia, medir a qualidade da rede baseado em métricas de QoS, calcular a qualidade do vídeo recebido pelo cliente baseado em métricas de QoV e armazenar de maneira coerente os dados levantados para posterior utilização do pesquisador. O objetivo principal é facilitar os processos de configuração do ambiente e realização dos experimentos práticos, permitindo maior dedicação à análise dos dados coletados e à implementação da proposta.

Na sessão 2 apresentamos alguns trabalhos desenvolvidos que realizam experimentos de transmissão multimídia. Na sessão 3 descrevemos o ambiente utilizado e algumas configurações importantes para realização dos experimentos. Na sessão 4 são apresentados os algoritmos que



compõe o *testbed*. Na sessão 5 demonstra-se algumas estratégias de análise dos dados coletados nos experimentos. Finalizamos na sessão 6 com a conclusão e trabalhos futuros.

## 2. TRANSMISSÃO DE VÍDEO EM REDES IP

Existem fatores importantes que direcionam os pesquisadores no estudo de propostas que viabilizem a utilização de conteúdo multimídia em redes IP. Destes, podemos destacar como essenciais a maior oferta de banda larga, o aumento da demanda de serviços multimídia na Internet, a evolução dos *codecs* de vídeo e a padronização do protocolo IP. No entanto, algumas pesquisas, possuem características experimentais, levando os pesquisadores a desenvolver seus *testbeds* visando tornar mais rápidas as coletas de dados que possam ser usados em suas propostas. Estes ambientes, normalmente demoram a ser montados, consumindo tempo da pesquisa. Os trabalhos descritos nesta seção são exemplos da necessidade de montagem de ambientes experimentais de testes envolvendo transmissão de vídeo em redes IP, fato que justifica a proposta de um *testbed* reutilizável, com características gerais para testes desta natureza.

Em Kim et al. (2011) o autor apresenta uma proposta que objetiva melhorar o desempenho fim a fim de transmissões de vídeo em tempo real no modelo *broadcast* sobre redes sem fio WLAN 802.11. O esquema tem por definição o uso do padrão WLAN 802.11 e do protocolo HCCA além da utilização de compressão de vídeo através da tecnologia H.264/SVC. Mackenzie et al. (2009) apresenta um estudo para melhoria da qualidade de serviço em redes WLAN na distribuição de arquivos multimídia. O autor utiliza-se do padrão de compressão de vídeo H.264 e do protocolo aprimorado de acesso ao canal distribuído (EDCA) do 802.11e. Portanto, o esquema proposto faz um mapeamento da influência que cada *slice* (pedaço) de vídeo exerce sobre a qualidade final e atribui prioridades de transmissão.

Em Fan et al. (2009) a proposta apresentada demonstra um ambiente de transmissão de vídeo em tempo real sobre redes WLAN. O trabalho descreve que a latência pode ser gerada por vários fatores intermediários: na transmissão, na codificação e na decodificação do vídeo. Portanto, a maior contribuição da proposta é a aplicação de métodos que procurem reduzir a latência nos processos intermediários. Em Feng et al. (2009), utilizando o protocolo IEEE 802.11e, o autor procura apresentar uma estratégia para oferecer qualidade de serviço em transmissões multimídia sobre redes WLAN. A proposta consiste, em síntese, na aplicação de um algoritmo que configura dinamicamente o valor do  $TXOP_{limit}$  de cada estação baseado nas condições atuais de tráfego.

Como observado, existem diversos trabalhos relacionados à garantia de QoS e QoV em ambientes de transmissão de vídeo. Notamos também que os autores apresentam propostas de melhoria, e as desenvolvem em diferentes frentes de pesquisa. É possível deduzir que, para alcançar os resultados mostrados, os autores se ocuparam em: configurar o ambiente de testes, realizar as transmissões multimídia, coletar e analisar os dados, aferir e comparar os resultados a fim de fazer as conclusões necessárias. Este ciclo em uma pesquisa dessa natureza pode se repetir diversas vezes.

## 3. A PROPOSTA

Portanto, a motivação deste trabalho é proporcionar ao pesquisador maior dedicação à proposta, à análise, à comparação e à conclusão referentes aos dados coletados. O *testbed* teria a função de realizar as seqüências de transmissão e as coletas dos dados, fases que consomem bastante tempo. A configuração da quantidade de experimentos que serão realizados em um ciclo e as limitações do ambiente proposto serão alguns dos parâmetros de entrada do *testbed*. Dentro das coletas de dados são armazenadas informações das métricas de QoS (atraso, jitter e perda) e das métricas de QoV (PSNR e BLUR) (Huynh-thu, 2012) (Crete, 2007). Com estes dados é possível analisar detalhadamente os resultados obtidos.

O ambiente de desenvolvimento dos experimentos é baseado na proposta de quem está realizando a pesquisa. Portanto, a infra-estrutura física pode possuir algumas diferenças dependendo do trabalho a ser realizado. Utilizaremos um ambiente infra-estruturado baseado na disposição física apresentada pela recomendação G. 1082 (ITU-T, 2009), e que define bem a maioria dos ambientes utilizados em pesquisas desta natureza como mostra a Figura 1.

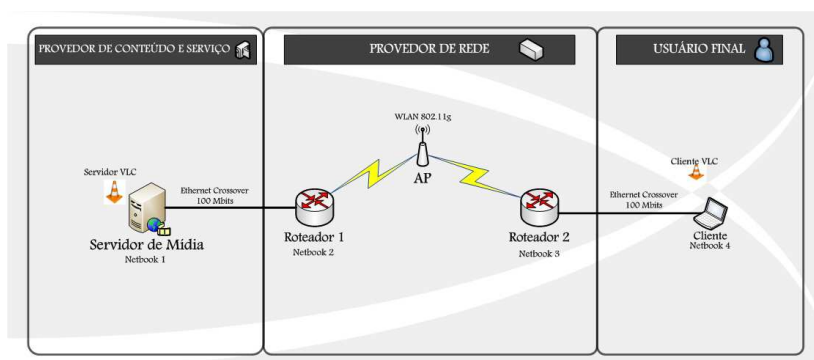


Figura 1. Descrição do ambiente

No ambiente temos um servidor multimídia que disponibiliza o conteúdo sob demanda aos clientes. Os vídeos são codificados no formato H264/MPEG4, utilizando *bitrate* de 300k e GOP 12. O conteúdo multimídia é disponibilizado utilizando o servidor vlc, e acessado através do cliente vlc no domínio do cliente. A recomendação G. 1082 apresenta o domínio de codificação e armazenamento dos *streams* de vídeo como o Provedor de Conteúdo e Serviço.

Seguindo o modelo, visualizamos o núcleo da rede que é formado por dois roteadores que se comunicam entre si através de uma WLAN 802.11. É neste domínio que podem ocorrer maior incidência de perda e atraso em razão das limitações das redes sem fio na transmissão de conteúdo em tempo real. Este domínio é conhecido na recomendação G. 1082 como Provedor de Rede.

No domínio do usuário, através da requisição de um dispositivo, o conteúdo é transmitido sob demanda. Os *streams* de vídeo armazenados no Provedor de Conteúdo e Serviço são enviados através do Provedor de Rede e visualizados pelo Usuário Final. No domínio do usuário, é implantado o *testbed* alvo deste trabalho, a fim de otimizar e automatizar as requisições de mídia e a coleta dos dados de transmissão referentes a rede (atraso, jitter e perda) e referentes ao vídeo (PSNR e BLUR).

Os *hosts* utilizados no ambiente da rede executam o sistema operacional Ubuntu 11.04. Os demais programas e componentes usados para programação e execução dos experimentos foram: ffmpeg (codificação dos vídeos), vlc (servidor e cliente de mídias), cbq shaper (limitador de banda das interfaces), octave (executa programa de cálculo de BLUR das imagens), psnr (realiza cálculo de PSNR), wine (executa programa de cálculo de psnr), dos2unix (converte arquivos gerados pelo psnr para o formato unix), bc (realiza cálculos diversos) e python (Linguagem de programação usada no script de coleta de dados na rede). As *Interfaces* de comunicação, endereços IP e rotas de cada *interface* foram definidas no arquivo de configuração de rede */etc/rc.local* de cada máquina do ambiente.

Durante os experimentos, muitas vezes é necessário simular uma sobrecarga na rede a fim de verificar a qualidade das transmissões em todas as possíveis situações. Esta sobrecarga pode ser diminuída ou aumentada através de estratégias de limitação de banda. Uma maneira interessante e eficiente de limitar a banda das *interfaces* é através do *cbq shaper*. Ao instalar o *cbq* é necessário elaborar um plano que permita executar as limitações de acordo com o propósito do trabalho desenvolvido. Em nosso modelo limitamos as *interfaces* de rede dos roteadores, afinal, nosso objetivo era verificar o comportamento do Provedor de Rede na transmissão do conteúdo. Para otimizar o processo de limitação das *interfaces* dos roteadores para tráfego multimídia foi desenvolvido um *script* para cada limitação desejada: 50Kb, 100Kb, 300Kb, 500Kb e 1000Kb.

Com ambiente configurado, rotas funcionando e arquivos de limitação de banda criados, colocamos em execução o servidor de mídia. Para isso, criamos um arquivo com extensão *.vlm* e um *script* que inicia o servidor de mídia. O arquivo *.vlm* é reconhecido pelo vlc e tem por função listar todos os vídeos que estarão disponíveis aos clientes. Após adicionar os vídeos no arquivo *.vlm* executamos o *script* de inicialização do vlc server.

### 3.1 Algoritmos de otimização

O objetivo do *testbed* é automatizar as transmissões multimídia analisando o comportamento da





rede durante os períodos de transmissão de cada *stream*. Portanto, faz-se necessário acompanhar e armazenar informações que permitam calcular a perda, o atraso e o jitter em cada seqüência. Para isto utilizamos um programa em *python* que transmite datagramas ICMP e captura o tempo de ida e volta de cada datagrama, de acordo com as condições da rede, tornando possível o levantamento das variáveis de QoS (Monteiro, 2012). Após a execução do programa, as informações são guardadas em um arquivo com a média de cada métrica de QoS citada acima.

A partir destas premissas, criamos os algoritmos de automação/otimização, que realizam seqüencialmente as solicitações dos vídeos no cliente vlc, baseado em uma estrutura de repetição com  $n$  transmissões para cada mídia. Além disso, o programa *python* faz a análise em tempo real do comportamento da rede. Portanto, utilizamos dois processos que executam paralelamente, enquanto o primeiro realiza as requisições vlc, o outro analisa o estado do meio. Ao encerrar cada transmissão é gerado um arquivo de *log* contendo o valor das métricas de QoS para cada seqüência.

O *testbed* foi desenvolvido e refinado para atender as demandas do ambiente e sua posterior reutilização em diferentes tipos de cenários. O *testbed* é composto por componentes separados, permitindo que o pesquisador decida o que deseja utilizar nas transmissões, como mostra a Tabela 1.

Componente	cliente.sh	qos.py	psnr.exe	blur.m
Função	Onde são implementados os algoritmos de requisição dos conteúdos multimídia.	Script desenvolvido em <i>python</i> para analisar e coletar informações sobre o estado da rede.	Calcula o PSNR dos frames de vídeos.	Programa em <i>octave</i> que calcula a métrica de borramento dos frames.

Tabela 1. Componentes do *testbed*

Os algoritmos foram desenvolvidos com objetivo de automatizar os processos de requisição do conteúdo multimídia, aferir a qualidade da rede através das métricas de QoS, calcular a qualidade do vídeo recebido pelo cliente utilizando métricas de QoV e armazenar de maneira coerente os dados obtidos. Apresentaremos adiante as funções desenvolvidas, as partes importantes dos códigos e as saídas que deverão ser analisados pelos pesquisadores.

Inicialmente, o *testbed* possui algumas dependências para seu correto funcionamento. Para evitar falhas na execução, foi desenvolvida uma função de verificação destas dependências. Assim, caso algum requisito não esteja sendo atendido, o programa apresenta uma mensagem informando sobre as dependências existentes e então encerra a execução. Com esta informação o usuário deverá proceder com a instalação dos programas solicitados. Após avaliação das dependências é feita a verificação das entradas do *script*. Caso haja algum erro será apresentada mensagem de acordo com a inconsistência encontrada, como demonstra a Tabela 2. Após realizar as validações, é preciso identificar quais os vídeos que serão utilizados nas transmissões. Para tanto, foi criada uma função que identifica a quantidade e o nome dos vídeos dentro de um diretório que pode ser definido no *script*. É importante observar que para os cálculos de PSNR, os vídeos precisam estar no formato *.yuv*.

Erro	Mensagem
Parâmetros obrigatórios não informados.	A utilização deste script exige que seja informado o número de testes desejado, a velocidade e o tipo de teste respectivamente. Exemplo - <code>./cliente.sh 10 300K 1</code>
Número de testes inválido.	O número de testes deve ser um número inteiro.
Velocidade do link inválido.	A velocidade do link deve ser informada em Kbits. Exemplo: 300K
Tipo de teste inválido.	O tipo de teste precisa ser um número entre 1 e 3. Tipo de Teste: (1 – PSNR, 2 – BLUR, 3 – PSNR E BLUR)

Tabela 2. Validações e mensagens de erro do *script* cliente.sh

#### A. Aquisição do conteúdo multimídia

Os arquivos multimídia são solicitados, transmitidos e armazenados através de um processo que percorre a relação de vídeos que constam no diretório definido no *script*. Dentro desta seqüência o

algoritmo realiza a quantidade de requisições por vídeo informadas pelo usuário durante a chamada do programa *cliente.sh*. Desta forma, é possível transmitir diversos *streams* de vídeo diferentes, quantas vezes o pesquisador achar necessário, realizando apenas uma chamada ao *testbed*.

Para cada sequência de vídeo transmitida, é criado um diretório onde se armazena todas as informações referentes aos testes, como: os arquivos *.mp4* recebidos no cliente, os arquivos de QoS capturados na rede para cada *stream*, os dados de PSNR e BLUR conforme o tipo de métrica escolhida. Dentro do *cliente.sh* consta o código de coleta de informações de QoS da rede e aquisição dos vídeos, como mostra a Tabela 3. É importante frisar que o endereço do servidor e a porta deverão ser alterados no *script* conforme demanda de cada ambiente. Os nomes dos vídeos que estão localizados no diretório definido no *script* devem ser exatamente iguais aos disponibilizados no servidor.

Função para aquisição dos vídeos	
01	<i>for i in \$videos</i>
02	<i>do</i>
03	<i>j=0</i>
04	<i>for ((j=0; j&lt;\$num_testes;j++))</i>
05	<i>do</i>
06	<i>python qos.py \$servidor \$i \$j&amp;</i>
07	<i>cvlc rtsp://\$servidor:\$porta/\$i --sout file/mp4:mp4/\$i/\$i\$j.mp4 vlc://quit</i>
08	<i>killall -9 ping</i>
09	<i>done</i>
10	<i>done</i>

Tabela 3. Algoritmo para aquisição de vídeos

Depois de finalizadas as transmissões das sequências de todos os vídeos, os cálculos de QoV são executados através de algumas funções conforme as opções selecionadas na chamada do sistema.

#### B. Calcula PSNR

A métrica PSNR é alcançada através de uma fórmula que compara dois frames de vídeo (original e transmitido) a fim de encontrar o valor de ruído sofrido pelo frame transmitido, como mostra a Tabela 4.

Função para cálculo de PSNR utilizando o programa <i>psnr.exe</i>	
01	<i>for i in \$videos</i>
02	<i>do</i>
03	<i>j=0</i>
04	<i>for ((j=0; j&lt;\$num_testes;j++))</i>
05	<i>do</i>
06	<i>ffmpeg -i mp4/\$i/\$i\$j.mp4 -b 300k mp4/\$i/\$i\$j.yuv</i>
07	<i>wine psnr.exe 352 288 420 yuv/\$i.yuv mp4/\$i/\$i\$j.yuv &gt; mp4/\$i/\$i\$j.dat</i>
08	<i>done</i>
09	<i>done</i>

Tabela 4. Algoritmo para cálculo de PSNR

Conforme apresentado no código, antes da realização dos cálculos do PSNR, o vídeo recebido no formato *.mp4* deve ser convertido novamente para o formato *.yuv*. Após a conversão é chamado o programa *psnr.exe* através do *wine*, a fim de comparar os frames do vídeo recebido com os frames do vídeo original. Isto é necessário porque o PSNR é uma métrica que precisa de uma referência. Os resultados são colocados em um arquivo *.dat* com o mesmo nome do vídeo avaliado.

#### C. Calcula BLUR



A métrica BLUR não necessita de referência, portanto, a forma de cálculo é diferente. Inicialmente, devemos converter os vídeos recebidos em imagens .png, para serem avaliadas. As imagens corrompidas são removidas e as remanescentes são reordenadas de forma que possamos realizar um *loop* no diretório até o total de arquivos, a fim de avaliar imagem por imagem. Após a realização do tratamento dos dados, é preciso executar os testes que calculam o grau de borramento, comparando a imagem atual com a imagem anterior, como mostra a Tabela 5 (Crete, 2007).

Função para cálculo blur	
01	<i>for i in \$videos</i>
02	<i>do</i>
03	<i>j=0</i>
04	<i>cont_frames=\$(ls mp4/\$i/\$i\$j/   wc -l)</i>
05	<i>sed -i "/video=/c video=\$video" blur.m</i>
06	<i>sed -i "/for u=/c for u=0:\$num_teste" blur.m</i>
07	<i>sed -i "/for i=/c for i=0:\$cont_frames" blur.m</i>
08	<i>octave blur.m # chama função que calcula o blur através do octave</i>
09	<i>done</i>

Tabela 5. Algoritmo para cálculo de blur

Na função de cálculo do BLUR, existem dois loops que percorrem as pastas onde foram guardadas as imagens de cada vídeo, realizando os cálculos e armazenando em um arquivo chamado *blurmetric.txt*. Como os vídeos possuem tamanhos distintos, esta função tem o objetivo de alterar o arquivo de cálculo de borramento *blur.m* adicionando nome do vídeo, número de testes e número de frames de cada vídeo. Após realizadas as modificações necessárias o *script* invoca o programa *octave*.

Conforme pode ser visto, as funções trabalham de forma independente, sendo possível a realização dos cálculos de PSNR, BLUR ou mesmo dos dois simultaneamente, sem interferências nos resultados um do outro. Os arquivos gerados nas transmissões e nos cálculos de QoV são relacionados a cada teste realizado. Caso sejam realizados diversos experimentos, serão gerados pelo menos dois arquivos por teste: um contendo os valores referentes ao QoS e o outro contendo os valores de QoV. O *testbed* então consolida estas informações em apenas 5 arquivos, onde cada arquivo conterá as médias de QoS (atraso, jitter e perda) e QoV (PSNR e BLUR) para cada sequência transmitida.

#### 4. APLICAÇÃO DO TESTBED

Com o ambiente em funcionamento é possível desenvolver o planejamento dos experimentos baseado no que se deseja testar ou analisar. Em nosso modelo, planejamos uma série de transmissões de mídia sobre o ambiente proposto. Naturalmente, em seqüências de testes, muitas tarefas são repetíveis, neste contexto um *testbed* que executa estas rotinas constantes se torna bastante útil.

Para demonstrar a funcionalidade dos algoritmos desenvolvidos, propomos um esquema de experimentos, adotando três seqüências de vídeo com características diferentes (*elephant\_dream*, *highway* e *rush\_hour*), a fim de analisar o comportamento da rede para cada *stream* (Xiph, 2012). Em cada seqüência aplicamos as seguintes limitações de banda para tráfego multimídia configuradas via *cbq*: 1000Kbps, 500Kbps, 300Kbps, 100Kbps e 50Kbps. Definimos a amostra modelo de 10 transmissões para cada seqüência transmitidas para as 5 limitações de banda, como descrito em (1):

$$\begin{aligned}
 total_t &= Q_{seq} * Q_{test} * Q_{limit} \\
 total_t &= 3 * 10 * 5 \\
 total_t &= 150
 \end{aligned}
 \tag{1}$$

onde,

$total_t$  – Total de transmissões multimídia a serem realizadas.

$Q_{seq}$  – Quantidade de seqüências de vídeo.

$Q_{test}$  – Quantidade de testes por limitação de banda.

$Q_{limit}$  – Quantidade de limitações de banda impostas a cada seqüência de vídeo.

Temos assim, um total de 150 requisições de transmissão, o que torna demasiadamente trabalhoso realizar manualmente as rotinas de solicitação de cada sequência de vídeo. Após as transmissões, o *testbed* sintetiza os dados obtidos de cada métrica de QoS e QoV, como mostra a Tabela 6.

highway - 300Kbps					
Transmissão	Jitter (ms)	Atraso (ms)	Perda (ms)	PSNR (0-100)	BLUR (0-1)
01	32,046	270,058	1	25,508	0,67623
02	33,060	273,520	1	25,490	0,67792
03	39,003	280,270	1	25,529	0,68048
04	35,513	269,118	1	25,709	0,68019
05	30,274	273,131	0	25,854	0,67866

Tabela 6. Sintetização dos dados feita pelo *testbed* em uma limitação de 300Kbps da sequência highway

Finalizados os testes, podemos então verificar o ganho de tempo que foi obtido com a utilização do *testbed*, o mesmo pode ser representado pela seguinte equação (2):

$$tempo_t = \sum_{i=0}^n t0_i + t1_i + t2_i + t3_i + t4_i + t5_i \quad (2)$$

onde,

$tempo_t$  – Tempo total para a execução dos testes.

$t0_i$  – Tempo para iniciar o monitoramento da rede.

$t1_i$  – Tempo para iniciar o processo de transmissão multimídia.

$t2_i$  – Tempo de transmissão de cada sequência.

$t3_i$  – Tempo para finalizar o monitoramento da rede.

$t4_i$  – Tempo para iniciar os cálculos de qualidade de vídeo (PSNR e/ou BLUR).

$t5_i$  – Tempo para a organização e sintetização dos dados obtidos.

$n$  – Número de testes a serem realizados.

Através desta fórmula, podemos então visualizar os tempos envolvidos na realização dos testes. A única variável que não é afetada pela utilização do *testbed* é a  $t2_i$ , que depende unicamente da duração do vídeo que está sendo transmitido. Todos os demais tempos tendem a zero, já que não é necessária a intervenção humana em nenhum destes procedimentos, o que torna os dados mais confiáveis e menos susceptíveis a erros.

## 5. CONCLUSÃO

Em um ambiente de testes que utilize transmissões de vídeo em qualquer infra-estrutura de rede, o pesquisador deverá realizar uma série de procedimentos a fim de obter os dados para análise e conclusões. Estes procedimentos normalmente são repetitivos, manuais e consomem bastante tempo.

Em um cenário definido em uma pesquisa específica que se exija transmissão de arquivos multimídia e análise do comportamento da rede, é necessário pelo menos realizar os seguintes procedimentos: I – aquisição dos vídeos em um modelo cliente/servidor; II – coleta das informações referentes ao estado da rede no momento das transmissões; III – sincronização entre a transmissão de vídeo e coleta de variáveis de QoS. IV – armazenamento dos dados de forma organizada; V – sintetização das informações coletadas manualmente a fim de proceder com as ações de análise. Este ciclo, em uma pesquisa dessa natureza, pode se repetir diversas vezes. Neste contexto, o ideal é que o pesquisador consuma maior tempo na análise dos dados, ao invés de se ater a estes procedimentos repetitivos e manuais que, inclusive, podem gerar inconsistências devido a algum erro humano.





Portanto, a proposta atua dentro desta realidade, realizando as seqüências de transmissões e as coletas dos dados automaticamente, de acordo com os parâmetros definidos pelo usuário. Como entrada é possível definir a quantidade de experimentos que serão realizados em um ciclo e as limitações de rede. As informações das coletas são armazenadas e consolidadas em arquivos organizados por cada métrica de qualidade, permitindo a síntese e a análise detalhada dos resultados.

Concluimos que o *testbed* proporciona um ganho satisfatório de tempo nas tarefas propostas a otimizar, evitando que o pesquisador seja obrigado a estar sempre no ambiente de testes realizando procedimentos repetitivos durante os experimentos. Essas funcionalidades proporcionam a possibilidade de dedicar-se a compreensão do problema e a proposta de solução, direcionando a pesquisa de acordo com a necessidade e complexidade do cenário utilizado.

Como trabalhos futuros podemos citar, a implementação de funções que realizem automaticamente os cálculos estatísticos dos dados obtidos, e retornem como saída informações de desvio padrão da amostra, intervalo de confiança, entre outras informações estatísticas relevantes a pesquisa.

## REFERÊNCIAS

- CRETE, F.; DOLMIERE, T.; LADRET, P.; NICOLAS, M. **The Blur Effect: Perpection and Estimation with a New No-Reference Perceptual Blur Metric**. SPIE Electronic Imaging Symposium Conf Human Vision and Electronic Imaging – 2007.
- DJAMA, I.; AHMED, T. **A Cross-Layer Interworking of DVB-T and WLAN for Mobile IPTV Service Delivery**. IEEE Transactions on Broadcasting, vol. 53 – 2007.
- FAN, B.; SHEN, L.; SONG, T. **The Design and Implementation of a Wireless Real-time Video Transmission System over WLAN**. IEEE Communications Society – 2009.
- FENG, Z.; WEN, G.; ZOU, Z.; GAO, F. **Red-txop Scheme for Video Trasmission In IEEE 802.11E EDCA WLAN**. IEEE Communications Society – 2009.
- FORCHHAMMER, S.; Li, H.; ANDERSEN, J. D. **No-reference analysis of decoded MPEG images for PSNR estimation and post-processing**. Jornal of Visual Communication and Image Representation, pg. 313-324 – 2011.
- HUYNH-THU, Q.; GHANBARI, M. **The accuracy of PSNR in predicting video quality for different video scenes and frame rates**. Telecommunication Systems – Vol. 49 Pg. 35-48 – 2012.
- ITU-T, Recommendation G. 1082. **Measurement-based methods for improving the robustness of IPTV performance** – 2009.
- KIM, Y. H.; PARK, H. R.; SUK, J. B. **A Scheme of H.264/SVC Video Broadcasting Over IEEE 802.11 WLAN**. IEEE Communications Society – 2011.
- MACKENZIE, R.; HANDS, D.; O'FARREL, T. **QoS of Video Delivered over 802.11e WLANs**. IEEE Communications Society – 2009.
- MIGLIORINI, D.; MINGOZZI, E.; VALLATI, C. **Performance evaluation of H.264/SVC video streaming over mobile WiMAX**. Computer Networks – Vol. 55 Pg. 3578–3591 – 2011.
- MONTEIRO, C. C. **Um ambiente para apoio à integração de redes sem fio heterogêneas**. Tese (Doutorado em Engenharia Elétrica) - Universidade de Brasília, Brasília – 2012.
- LIANG, J. Y.; YANG, B.; NAHRSTEDT, K. **A Testbed for Future Internet-Based TV Broadcasting**. Department of Computer Science, University of Illinois at Urbana – Champaign – 2007.
- MINGARDI, C.; BRUNNER, M. **IPTV Quality of Service Management in Home**. IEEE Communications Society – 2009.
- SINGH, H. K.; KIM, C. Y.; NGO, S. S. **IPTV over WirelessLAN: Promises and Challenges**. IEEE Communications Society – 2008.
- XIPH. <http://media.xiph.org/video/derf/> - 2012.