

UM ESTUDO COMPARATIVO SOBRE A QUALIDADE DA TRANSMISSÃO DE VÍDEOS EM REDES SEGMENTADAS

Leonardo Ferreira Borges¹; Fernando Eduardo Rezende Mattioli²; Eduardo Fernandes Saad³; Diovani dos Santos Milhorim⁴

^{1, 2, 3, 4} Faculdade de Talentos Humanos - FACTHUS, Uberaba (MG), Brasil

²Universidade Federal de Uberlândia - UFU, Uberlândia (MG), Brasil

³Centro Brasileiro de Pesquisas Físicas - CBPF, Rio de Janeiro (RJ), Brasil

leofbrgs@gmail.com, fernando.mattioli@facthus.edu.br, eduardo.saad@facthus.edu.br, diovani.milhorim@facthus.edu.br

RESUMO: O tráfego de rede por vezes se torna lento, um dos motivos é a grande quantidade de transmissão em um único bloco de comunicação, apontando para soluções como segmentação de rede. Este artigo demonstra a avaliação da qualidade da transmissão de um vídeo, através de uma rede corporativa de produção antes e após a segmentação por VLAN, utilizando métricas de avaliação mundialmente difundidas para testes em redes de computadores. Foi utilizada a rede corporativa de uma empresa do segmento agroindustrial, que já é uma rede segmentada, criando um novo segmento para os testes serem efetuados sem prejudicar o andamento das operações da empresa. Foi gerado a transmissão do vídeo utilizando dois *laptops* em dois cenários, onde um completamente isolado de qualquer tráfego e outro simulando uma rede com tráfego de dados intenso, para avaliar a qualidade final dos vídeos mediante os cenários. Para a criação do tráfego concorrente e avaliação da rede foi utilizada a ferramenta *IPERF3*. As métricas utilizadas para a avaliação da qualidade dos vídeos requerem que haja pares para comparação, a marcação dos frames, que foi necessária para identificar se há uma correspondência entre os vídeos comparados e após a transmissão em uma comparação quadro a quadro. Por fim, os resultados apontam que uma rede com menor tráfego concorrente, apresenta maior desempenho em relação à rede com maior tráfego, com as variações de banda e atraso (*Jitter*) influenciando diretamente na qualidade.

PALAVRAS CHAVE: Avaliação de qualidade, métrica objetiva, transmissão de vídeo, redes de computadores, segmentação de redes, VLAN, LAN

A COMPARATIVE STUDY ABOUT QUALITY OF TRANSMISSION OF VIDEOS IN SEGMENTED NETWORKS

ABSTRACT: Sometimes the network traffic become slow and one of the reasons is the amount of traffic in one communication segment, indicating that the solution is divide the network in segments. This article demonstrates the valuation of quality video transmission through a production corporative network before and after the segmetation by VLAN, using valuation metrics worldwide widespread to network tests. Also valuating the final quality of this video after the transmission in those two enviroments, using two objective valuation metrics. It was used the corporate network of a company in the agroindustrial segment, which is already a segmented network, creating a new segment for the tests to be carried out without harming the progress of the company's operations. It was generated the video transmission using two laptops in two scenarios, where one completely isolated from any traffic and another simulating a network with intense data traffic, to evaluate the final quality of the videos through the scenarios. The *IPERF3* tool was used to create concurrent traffic and network evaluation. The metrics used to evaluate the quality of the videos require that there be pairs for comparison, the marking of the frames, which was necessary to identify if there is a correspondence between the videos compared and after the transmission in a frame-by-frame comparison. Finally the results indicate that one network with minor concurrent traffic shows a better performance in relation of network with larger traffic, with variations of band and delay (*Jitter*) directly influencing the quality.

KEY WORDS: Quality valuation, Objective metric, Video transmission, computer networks, network segmentation, VLAN, LAN

INTRODUÇÃO

Redes de dados ou redes de computadores podem ser definidas como um composto de computadores independentes, interconectados por meio de uma única

tecnologia. (TANENBAUM, 2002). Quanto maior o número de pessoas e equipamentos trafegando dados nestas redes, maior o volume de frames de broadcast em trânsito, isto conseqüentemente diminui seu desempenho.

Partindo de uma rede local, interligação de uma rede privada limitada a um espaço geográfico conhecido entre entidades como LAN - *Local Area Network*, tem-se um domínio de propagação, *broadcast*, onde um dispositivo conectado envia um quadro na rede que acaba disseminando para todos os outros dispositivos de rede criando tráfego desnecessário. Assim, o conceito de rede local virtual (VLANs) surge como uma resposta para estes problemas. (FILIPPETTI, 2015).

Este artigo tem por objetivo a avaliação da qualidade de transmissão de vídeo, através de uma rede de computadores com concorrência de tráfego similar à de uma rede corporativa não segmentada, em seguida verificar após a segmentação por uma rede virtual (Vlan) utilizando métricas objetivas.

Segmentação de Redes

A (VLAN), *Virtual Local Area Network*, consiste na segmentação de redes como objetivo de diminuir a disseminação quadros no broadcast em redes locais, possibilitando que várias redes locais virtuais sejam independentes através de uma única infraestrutura física de rede, melhorando assim o desempenho no tráfego e dados. [ELIAS e LOBATO 2012].

A operação de uma VLAN só é possível em uma rede comutada. O componente de rede *Switch* permite que o administrador de rede configure como sendo uma única entidade interligada. Todavia, são asseguradas aos usuários a conectividade e a privacidade que são esperadas como se existissem múltiplas redes separadas. (DANTAS, 2002).

A transmissão de vídeo em redes.

A transmissão de vídeo através das redes de computadores tem se tornado algo cada vez mais rotineiro. Para que as transmissões tenham uma qualidade aceitável, é muito importante que estas redes sejam bem projetadas, pois influenciará diretamente na qualidade final da transmissão. Vídeos digitais estão vulneráveis a diversas formas de interferências que podem ocorrer durante os processos de aquisição, processamento, compressão, armazenamento e transmissão, resultando em perda da característica visual. A avaliação da qualidade de vídeo é notável para definir métricas e propor a redução da deterioração. Existem duas formas para efetuar a avaliação da qualidade de vídeo, sendo a subjetiva e objetiva, porém a avaliação objetiva é mais rápida e tem custo menor. (REGIS, 2013).

A Linguagem Python.

Python é uma linguagem de programação multiparadigma. Lançada inicialmente por Guido van Rossum em 1991, e atualmente é aberta e com gerenciamento pela Python Software Foundation. Além disto foi utilizada as bibliotecas OpenCV que originalmente foi desenvolvida pela Intel para

desenvolvimento de aplicativos em Visão Computacional, sua licença segue o modelo BSD Intel, Numpy que possui uma grande coleção de funções matemáticas para trabalhar com arrays e matrizes em Python e de licença BSD. Também Scikit-image que é uma biblioteca de processamento de imagens para Python de licença BSD e foi desenvolvida por Stéfan van der Walt.

O IPERF

De acordo com Mota Filho (2013) o *IPERF* é uma ferramenta pensada com o objetivo de medir o tráfego gerado por uma rede e seu desempenho. Para isto, é necessário ter uma entidade de rede operando como cliente e outra entidade funcionando como servidor, com o objetivo de realizar a verificação das extremidades do caminho da rede. O RFC 6349 (2011), mostra que o *IPERF* é a ferramenta mais utilizada na medição de redes. Existem vários parâmetros, dentre eles a taxa de transferência direcional ou bidirecional máxima entre dois pontos de uma rede. Os testes devem ser realizados em vários momentos do dia e com duração superior a 30.

O Protocolo TCP

O TCP (*Transmission Control Protocol*) que tem sua tradução para Protocolo de Controle de Transmissão, é um protocolo de nível da camada 4, a de transporte onde se contem a maioria dos protocolos de aplicações, da *World Wide Web* como o SSH, FTP, HTTP. Este Protocolo de transmissão vem com prerrogativa de confiabilidade, pois entrega na sequência correta e processa a verificação de erros dos pacotes de dados entre as diferentes entidades de rede.

O Protocolo UDP

O UDP (*User Datagram Protocol*) é um protocolo descrito na RFC 768 da camada de transporte que permite o envio de um datagrama sem qualquer tipo de garantia que o pacote chegue corretamente, sendo considerado um protocolo não confiável. Cada datagrama UDP tem um tamanho diferente e pode ser considerado como um registro indivisível e sem orientação a fluxo de bytes, sem conexão permanente entre entidades. Um exemplo é descrito podendo um servidor ler datagramas vindos de diversos clientes, usando um único socket.

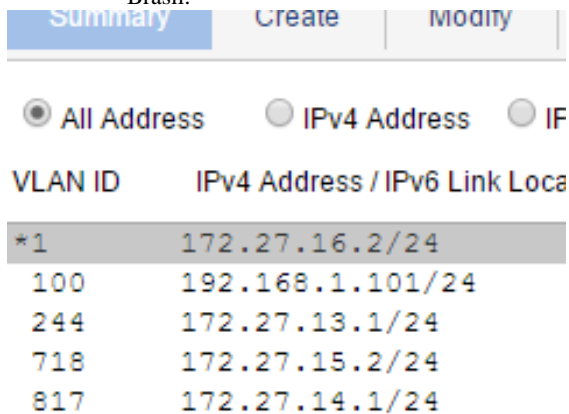
MÉTODOS

Para este estudo foi utilizada a rede corporativa de uma empresa do segmento agroindustrial de origem indiana, esta já é uma rede segmentada, conforme imagem retirada do *Switch Core* (Figura 1).

Foi adicionado um novo segmento para os testes serem efetuados sem prejudicar o andamento das operações da empresa.

Esta rede utiliza para comunicação *Switches* de Camada 2 e Camada 3 do modelo OSI, em que *Switches* de Camada 3 possibilitam que exista tráfego entre VLANs equipamentos conectem redes diferentes, isto é, domínios de broadcast diferentes, o que é tradicionalmente feito por um router entre dois *Switches* Camada 2 (TORRES, 2014). Estes equipamentos estão interligados por uma fibra optica monomodo, com módulos SFP LX para tráfego entre os nós da rede. Na distribuição é utilizado cabo par trançado CAT5E. A configuração das VLANs é apresentada conforme Tabela 1.

Figura 1 – Configuração da segmentação de rede UPL do Brasil.



VLAN ID	IPv4 Address / IPv6 Link Local
*1	172.27.16.2/24
100	192.168.1.101/24
244	172.27.13.1/24
718	172.27.15.2/24
817	172.27.14.1/24

Fonte: Os autores, 2017.

Tabela 1 – Funções das VLANs.

VLAN ID	Função
1	Vlan padrão, tráfego de dados pela rede cabeada de todos os sistemas e ferramentas
244	Telefonia VoIP
100	Banda Larga
718	WiFi Corporativa
817	WiFi Visitantes

Fonte: Os autores, 2017.

A modalidade de configuração da VLAN utilizado neste cenário é o de VLAN por porta, que trabalha na camada 1 do modelo OSI definido pelo padrão IEEE 802.1Q, este é o método mais utilizado, por ser de rápida e fácil configuração, conforme Moraes, (2002).

Para a implementação dos testes foi criada uma rede virtual (VLAN), paralela às redes virtuais do ambiente corporativo para que não houvesse interrupções das atividades rotineiras da empresa, pois a geração de tráfego concorrente poderia colapsar a rede em questão. A criação da VLAN120, que é a nova segmentação da rede exclusivamente para os testes, foi implementada inicialmente no *Switch Core* (Camada 3), utilizando a

interface gráfica do equipamento. Posteriormente foi criada esta VLAN nos *Switches* de distribuição, implementada com da mesma forma do *Switch Core*. As portas que conectam os *Switches* entre si são marcadas como portas tronco, esta configuração é utilizada para envio e recebimento de tráfego Ethernet de inúmeras VLANs. De acordo com Dias (2005), por padrão estas portas pertencem a todas as VLANs. Já as portas onde os computadores de fato se conectaram, foram configuradas em modo de acesso com a VLAN 120 como *tagged*, isto permitiu a comunicação entre os computadores.

Foi gerado a transmissão do vídeo em dois cenários, onde um completamente isolado de qualquer tráfego e outro simulando uma rede com tráfego de dados intenso, para avaliar a qualidade final dos vídeos mediante os cenários.

Para gerar a transmissão do vídeo foram utilizados dois laptops, o servidor com um processador i7, 8 Gigabytes de memória RAM e 500 Gigabytes de disco rígido, já o cliente com um processador i3, 4 Gigabytes de memória RAM e 500 Gigabytes de disco rígido, ambos conectados via cabo CAT5E. Com o servidor conectado a um Switch de distribuição e o cliente no Core, para gerar o tráfego concorrente no ambiente, foram utilizados 2 laptops ambos com a seguinte configuração: Processador Core i3, 4 Gigabytes de Memória e 500 Gigabtes de Disco Rígido, foi adotado o player e transmissor de multimídia VLC Player para efetuar a transmissão de um vídeo de domínio público com duração de 40 segundos.

As métricas utilizadas para a avaliação da qualidade dos vídeos requerem que haja pares para comparação, isto é, o vídeo original e o transmitido. A marcação dos frames, foi necessária identificar se há uma correspondência entre os vídeos comparados e após a transmissão, poderemos realinhar os quadros para uma comparação quadro a quadro. A avaliação da qualidade do vídeo para os dois ambientes foi realizada medindo-se as métricas SSIM (*Structural Similarity*), e PSNR (*Peak Signal Noise Ratio*) frame a frame, por meio de um script implementado na linguagem Python, gerando um arquivo CSV que contem os valores da comparação utilizando as métricas, onde cada célula do arquivo gerado equivalente ao valor obtido na comparação. A equação utilizada para calcular o PSNR é utilizado o calculo de MSE (Métodos de Suavização Exponencial) que é, segundo Sarmento (2012), definido como o valor médio dos erros quadráticos entre pixels, do quadro original e do quadro codificado. O MSE é representado pela fórmula:

$$MSE = \frac{1}{P} \sum_{i=0}^{K-1} \sum_{x,y} (f(x, y, i) - h(x, y, i))^2 \quad (1)$$

onde P é o numero total de pixels do quadro, x e y são os indices das linhas e colunas, K é o numero de frames, f e h o quadro original e o modificado. De acordo com Bezago (2011), a avaliação por PSNR é uma maneira para definir a relação entre a máxima energia de um sinal

e ruído que afeta sua reprodução, esta métrica é definida pela equação:

$$\text{PSNR} = 10\log[255^2/\text{MSE}] \quad (2)$$

Os valores encontrados variam de 0 a 100, onde 100 é o melhor valor.

A métrica objetiva SSIM compara cada frame das séries de vídeo original e transmitido, para medir a degradação ocorrida estabelecendo a similaridade entre os vídeos (BEZAGO, 2011). Segundo ROMANI (2015), esta utiliza componente de luminância dos vídeos, onde Sinal x de referência e y do transmitido, isto resulta na média de igualdade. As comparações de luminância, contraste e estrutura são dadas pelas seguintes equações:

$$l(f, x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (3)$$

$$c(f, x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (4)$$

$$s(f, x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \quad (5)$$

onde μ_x e μ_y são as médias das entradas, σ_x e σ_y a variância e σ_{xy} a covariância cruzada de x e y .

A seguinte equação representa esta métrica:

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (6)$$

Em que, segundo ESTRADA (2011), α , β e γ são razões que definem a importância proporcional de luminância, contraste e estrutura. Se $\alpha = \beta = \gamma = 1$, a equação acima se reduz a:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (7)$$

Em que 1 o é melhor valor possível e 0 o pior.

Para a criação do tráfego concorrente e avaliação da rede foi utilizada a ferramenta *IPERF3*. Esta é uma ferramenta cliente/servidor que integra em um único software o relatório para a análise das métricas como largura de banda máxima, atraso na entrega de pacotes e a perda. Atualmente o *IPERF3* é mantido e desenvolvido principalmente pelo ESnet / Lawrence Berkeley National Laboratory sob licença BSD (*IPERF*, 2017).

Transmissão do Vídeo

Para que fosse iniciada a transmissão do vídeo, foi utilizada a forma EasyStream do VLC, seguindo as orientações da documentação disponível no Wiki da

ferramenta. Adicionou-se um vídeo de formato AVI para fazer a transmissão através da rede, utilizando o protocolo RTSP, especificado o IP e porta do servidor, e foi utilizado o método de encapsulamento H.265 e MP# (*VIDEOLAN*, 2017).

Gravação do Vídeo Transmitido

Para efetuar a gravação da transmissão foi utilizado também a ferramenta VLC, utilizando o método de salvamento direto sem nenhum tratamento, isto é, o descarregamento bruto, para que não houvesse nenhuma interferência na qualidade do vídeo final (*VIDEOLAN*, 2017).

Avaliação da Rede

Conexões Ethernet devem ser avaliadas para que exista a garantia de que elas estão funcionando de forma correta e com o desempenho esperado. Os testes que devem ser executados para que exista essa garantia são: vazão ou *throughput*, que pode ser definido como a quantidade máxima de dados que pode ser trafegado do ponto de origem até o destino, latência que é definido como o tempo total que um quadro gasta para percorrer o caminho da origem até o destino, a variação da latência (*jitter*), definido como o atraso entre chegadas de quadros até o destino, e a perda de quadros, definida pela média de perda de quadros que ocorrem durante uma transmissão (MORAES, 2002).

As avaliações seguiram as métricas apontadas pela recomendação RFC 2544 do IETF que determina os testes para que se obtenha resultados reais a respeito de uma rede de computadores, somente não foi utilizada a métrica o tamanho dos quadros e a quantidade de testes (RFC 2544, 1999). Durante uma transmissão de vídeo por streaming, estes fatores podem interferir de forma positiva ou negativa, isto vai variar de acordo com a qualidade da rede.

Na ferramenta *IPERF3* foi utilizada a configuração padrão, onde foi somente alterado o protocolo de transmissão de TCP para UDP, onde foi possível criar fluxos UDP para medir a variação da latência (*Jitter*) e medir a perda de pacotes (SILVA; JÚNIOR, 2017). Para efetuar a medição da vazão foi utilizado o *IPERF3* no modo de transmissão TCP, com as configurações padrões, pois com isto ele utiliza a vazão dos hosts. Esta alteração foi importante para a medição, pois, segundo Tanenbaum (2003), o protocolo TCP faz um tratamento a respeito da transmissão dos pacotes, utilizando métricas para verificação da entrega, sendo assim, se há alguma perda ao longo do tráfego ele retransmite o pacote, ocorrendo a garantia de entrega. Ao contrário do UDP que continua transmitindo os pacotes independente da perda.

Por isto foi utilizado o UDP para verificação dos parâmetros de atraso e perda, e TCP para vazão. Para gerar tráfego concorrente, a ferramenta foi executada no modo UDP com 5 transmissões em paralelo durante o

período de 600 segundos, simulando o que ocorre com uma rede de computadores quando se tem uma quantidade de equipamentos transitando pacotes UDP na rede. A medição do tráfego foi feita pelo período de 100 segundos para todas variantes, obtendo no fim da avaliação a média de cada uma. Para executar o *IPERF3* são utilizados parâmetros de configuração para a transmissão do tráfego e análise do mesmo que estão descritos na tabela a seguir:

Tabela 2 – Funções dos parâmetros do *IPERF*

Parâmetro	Função
-s	Executar como servidor
-c	Executar como cliente
-i	Intervalo de tempo, em segundos, que os relatórios são gerados

-u	Executa o teste no utilizando o protocolo UDP
-P	Quantidade de conexões simultâneas
-f	Formato da impressão do teste
-t	Tempo de transmissão
-p	Porta utilizada para conexão cliente/servidor

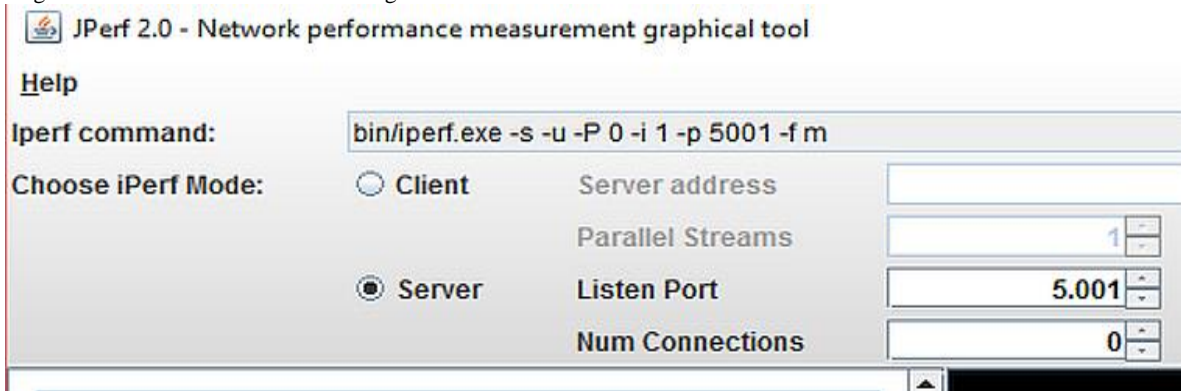
Fonte: *IPERF*, 2017.

Para efetuar estes testes, foi executado o *IPERF* no computador servidor, como é observado na figura 2.

Do lado cliente o *IPERF3* foi conforme a figura 3 a seguir.

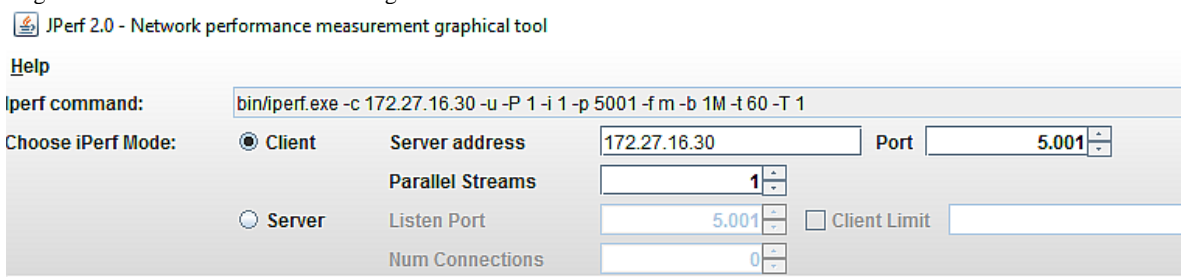
O script para tal procedimento é mostrado no anexo 1 deste trabalho.

Figura 2 – *IPERF* executado no modo gráfico como servidor.



Fonte: Os autores, 2017.

Figura 3 – *IPERF* executado no modo gráfico como cliente.



Fonte: Os autores, 2017.

Avaliação da qualidade do vídeo final

Conforme descrito anteriormente, foi feita o realinhamento dos frames de ambos os ambientes para que o script pudesse fazer a comparação frame a frame, evitando que fosse comparado frames diferentes e os valores fossem corrompidos. Após isto foi adicionado o vídeo original, o transmitido pela VLAN1 e VLAN120, todos eles já alinhados. Os resultados da avaliação foram escritos em uma planilha para que fossem gerados gráficos e médias.

RESULTADOS

Pode-se determinar, de acordo com a Tabela 3, que houve diferenças significativas nos ambientes avaliados, ainda que na avaliação das redes não foi observado perda de pacotes. Ocorreu uma perda significativa de frames, diretamente influenciados pelo *Jitter* e *Vazão* apresentadas. Além disto, as métricas apontaram que as proximidades são muito distintas.

Na Tabela 3 pode-se observar e comparar os índices avaliados nos dois ambientes, em que para a VLAN120 obtivemos valores muito próximos do ideal,

como na métrica SSIM onde o ambiente VLAN120 obteve um valor de 89% de proximidade com o original, o índice varia entre 0 para ruim e 1 para ótimo. Para os dois ambientes os vídeos avaliados na métrica PSNR não apresentaram similaridade próxima ao valor máximo, que é 100, apesar de haver diferenças nos dois ambientes. Além disto, houve perda de frames nos dois ambientes, porém pode-se constatar que no ambiente VLAN120, a perda foi menor, com uma taxa de 10,7% de perda.

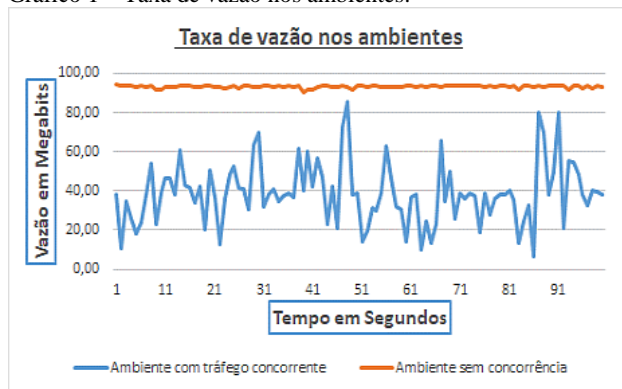
Tabela 3 - Resultados comparativos

Parâmetros	VLAN1	VLAN 120
PSNR (%)	22,5	30,3
SSIM (%)	62	89
Jitter (ms)	2,204	0,143
Largura de Banda (Mbits/s)	23,6	93,3
Perda de Pacotes (%)	0	0
Perda de Frames (%)	13,7	10,7

Fonte: Os autores, 2017.

O Gráfico 1 apresenta a comparação nos dois ambientes a respeito da taxa de vazão observada. Pode-se concluir que quando há tráfego concorrente, a taxa de vazão oscila e diminui em relação ao ambiente sem tráfego concorrente. Isto quer dizer que aplicações que necessitam de uma largura de banda mínima podem sofrer uma drástica queda na sua qualidade, podendo até não iniciar ou finalizar a sua tarefa.

Gráfico 1 – Taxa de vazão nos ambientes.



Fonte: Os autores, 2017.

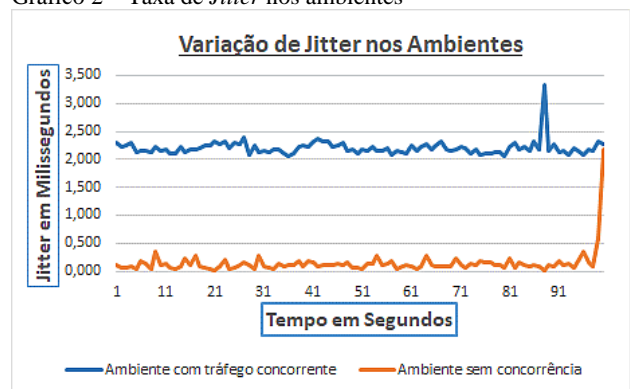
O Gráfico 2 apresenta a diferença do *Jitter* nos dois ambientes, onde o ambiente com uma taxa de tráfego concorrente menor apresenta menores taxas. Aplicações de streaming sofrem influência direta deste quesito, pois utilizam o protocolo UDP para transmissão de pacotes, onde não há garantia alguma da entrega, por este motivo pode haver perda de pacotes ao longo da transmissão, o que causa uma perda de qualidade. Esta diferença influenciou diretamente na qualidade final dos vídeos transmitidos, causando perda de frames e qualidade.

O Gráfico 3 apresenta a comparação da avaliação utilizando a métrica PSNR para os vídeos transmitidos, onde pode-se observar que há uma variação de qualidade no ambiente VLAN1 e uma maior constância no

ambiente VLAN120. Além da constância a VLAN120 apresenta maior taxa de similaridade com o vídeo original. Observa-se picos de qualidade no ambiente VLAN1, causado por oscilações na vazão. Além disto pode-se verificar que a partir do frame 1036 não há mais transmissão para o ambiente VLAN1.

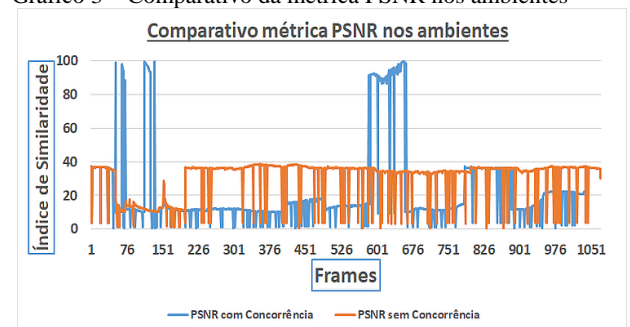
O Gráfico 4 apresenta a comparação da avaliação utilizando a métrica SSIM para os dois ambientes, onde podemos observar que inicialmente há uma similaridade nos dois vídeos, porém a partir do frame 226 o ambiente da VLAN1 mantém a qualidade baixa enquanto VLAN120 eleva a qualidade e mantém uma constância até o fim da transmissão. Esta diferença está ligada totalmente ao alto valor de *Jitter* para o ambiente VLAN1.

Gráfico 2 – Taxa de *Jitter* nos ambientes



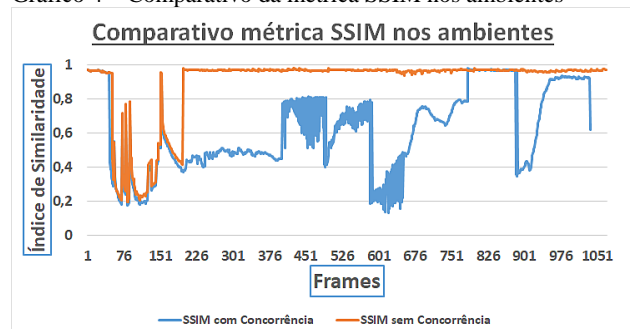
Fonte: Os autores, 2017.

Gráfico 3 – Comparativo da métrica PSNR nos ambientes



Fonte: Os autores, 2017.

Gráfico 4 – Comparativo da métrica SSIM nos ambientes



Fonte: Os autores, 2017.

CONCLUSÃO

Chegou-se à conclusão de que um ambiente segmentado, com menor número de equipamentos conectados à rede tem um desempenho superior quando se trata de transmissão de vídeo. Com as variações de banda e atraso (*Jitter*) influenciando diretamente na qualidade. Um ambiente bem gerenciado pode afetar em transmissões de vídeo positivamente, trazendo maior qualidade para as comunicações. Entretanto, a segmentação de uma rede de computadores deve ser feita de forma consciente, avaliando a quantidade, tipo e necessidades do tráfego que irá passar por um dado segmento.

Somente dividir a rede por si não aumenta o desempenho de aplicação alguma, a avaliação e o planejamento são indispensáveis para a qualidade e escalabilidade de uma rede de computadores. Isto ainda se tratando de tráfego de streaming pode ser evidenciado a olho nu, criando um ambiente completamente inadequado.

REFERÊNCIAS

BEZAGO, Dante Coaquira. **Avaliação de qualidade de vídeo sobre uma rede IP usando métricas objetivas**. 2011. 5 f. TCC (Graduação) - Curso de Engenharia de Sistemas Eletrônicos, Universidade de São Paulo, São Paulo, 2011.

BRADNER, S.; MCQUAID, J. **Benchmarking Terminology for Network Interconnection Devices**: RFC 2544. [S.I]: Internet Engineering Task Force, Network Working Group, 1999.

DANTAS, Mario. **Tecnologia de Redes de Comunicação e Computadores**. Rio de Janeiro: Axcel Books, 2002.

DIAS, Diego. **Perguntas e Respostas: Portas Access/Trunk/Híbrida, LACP e STP**. 2015. Disponível em: <<http://www.comutadores.com.br/perguntas-e-respostas-portas-access-trunk-hibrida-lACP-e-stp/>>. Acesso em: 12 maio 2017.

ESTRADA, Cássius Rodrigo Duque. **AVALIAÇÃO AUTOMÁTICA DE QUALIDADE DE VIDEOCONFERÊNCIAS DE ALTA DEFINIÇÃO**. 2011. 81 f. Dissertação (Mestrado) - Curso de Pós-graduação em Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

FILIPPETI, Marco Aurélio. **CCNA 5.0 Guia completo de estudo**. Florianópolis: Visual Books, 2014. 544 p.

IPERF. **IPERF - The ultimate speed test tool for TCP, UDP and SCTP**. 2017. Disponível em: <<https://iperf.fr/>>. Acesso em: 27 maio 2017.

MACEDO, Diego. **Introdução ao Wireshark: Detecção e captura de tráfego em redes**. 2016. Disponível em: <<http://www.diegomacedo.com.br/introducao-ao-wireshark->

[deteccao-e-captura-de-trafego-em-redes/](#)>. Acesso em: 27 maio 2017.

MORAES, Igor Monteiro. **VLANS - Virtual Local Area Networks**. 2002. 12 f. TCC (Graduação) - Curso de Redes de Computadores, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2002. Disponível em: <https://www.gta.ufrj.br/grad/02_2/vlans/referencias.html>. Acesso em: 17 maio 2017.

MOTA FILHO, João Eriberto. **Análise de tráfego em Redes TCP/IP**. São Paulo: Novatec Editora, 2013.

RFC 6349. **Framework for TCP Throughput Testing**. 2011. Disponível em: <<https://tools.ietf.org/html/rfc6349>>. Acesso em: 17 maio 2016.

REGIS, Carlos Danilo Miranda. **Métrica de Avaliação Objetiva de Vídeo Usando a Informação Espacial, a Temporal e a Disparidade**. 2013. 138 f. Tese (Doutorado) - Curso de Pós-graduação em Engenharia Elétrica, Universidade Federal de Campina Grande, Campina Grande, 2013. Disponível em: <http://www.difusaoacientifica.com.br/dt/Tese_Danilo_Regis.pdf>. Acesso em: 20 maio 2017

ROMANI, Eduardo. **AVALIAÇÃO DE QUALIDADE DE VÍDEO UTILIZANDO MODELO DE ATENÇÃO VISUAL BASEADO EM SALIÊNCIA**. 2015. 65 f. Dissertação (Mestrado) - Curso de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná, Curitiba, 2015.

SARMENTO, Camila Freitas; REGIS, Danilo; TAYNNAN, Michael. **Avaliação do Ganho de Qualidade de Vídeos Transmítidos com Codificação de Canal**. In: CONGRESSO NORTE NORDESTE DE PESQUISA E INOVAÇÃO, VII., 2012, Palmas.

SILVA, Pedro Henrique Diniz da; ALVES JÚNIOR, Nilton. **Ferramenta IPERF: geração e medição de Tráfego TCP e UDP**. *Cbpf*, Rio de Janeiro, v. 2, n. 4, Quadrimestral. Disponível em: <<http://revistas.cbpf.br/index.php/nt/article/view/75>>. Acesso em: 20 maio 2017.

TANENBAUM, Andrew S.. **Computer Networks**. 4. ed. Amsterdam: Campus, 2003. 968 p.

TORRES, Gabriel. **Redes de Computadores**. 2. ed. Rio de Janeiro: Novaterra, 2014. 955 p.

VIDEOLAN. **VideoLAN's Documentation**: The reference documentation on VideoLAN's projects. 2017. Disponível em: <<https://wiki.videolan.org>>. Acesso em: 12 maio 2017.

ANEXO 1

```
#!/usr/bin/python
# -*- coding:utf-8 -*-

import cv2
import numpy as np

from skimage.measure import compare_ssim, compare_psnr

REFERENCE = "output-features.avi"
TARGET = "SEM-CONCORRENCIA.mp4"
FILENAME = "SEM-CONCORRENCIA.csv"

class frame:
    def __init__(self, bincode, data, ssim, psnr):
        self.bincode = bincode
        self.data = data
        self.ssim = ssim
        self.psnr = psnr

def read_bin_line(img):

    code = []

    for k in range(0, 640, 80):
        b = 0
        w = 0

        for i in range(460, 470):
            for j in range(k, k + 80):
                if np.mean(img[i, j]) > 127:
                    w += 1
                elif np.mean(img[i, j]) < 127:
                    b += 1

            if w > b:
                code.append(1)
            elif w < b:
                code.append(0)
            else:
                code.append(-1)

    for k in range(0, 640, 80):
        b = 0
        w = 0

        for i in range(470, 480):
            for j in range(k, k + 80):
                if np.mean(img[i, j]) > 127:
                    w += 1
                elif np.mean(img[i, j]) < 127:
                    b += 1

            if w > b:
```



```
        code.append(1)
    elif w < b:
        code.append(0)
    else:
        code.append(-1)

    try:
        bincode = int(''.join(map(str, code)), 2)
    except ValueError:
        bincode = 0

    return bincode

if __name__ == "__main__":
    cap = cv2.VideoCapture(REFERENCE)

    referenceFrames = {}

    print 'Reading reference frames...'

    found = False

    while (True):
        ret, img = cap.read()

        if ret == True:
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            bincode = read_bin_line(img)

            if (bincode % 100 == 0):
                print "Adding frame ", bincode

            if bincode == 1:
                if found:
                    break
                else:
                    found = True

            referenceFrames[bincode] = img

        else:
            break

    cap.release()

    cap = cv2.VideoCapture(TARGET)

    targetFrames = []

    print 'Reading target frames...'

    while (True):
        ret, img = cap.read()

        if ret == True:
```

```
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
bincode = read_bin_line(img)

if bincode % 100 == 0:
    print 'Processing frame ', bincode

if bincode in referenceFrames:
    ssim = compare_ssim(referenceFrames[bincode], img,
multichannel=True)
    psnr = compare_psnr(referenceFrames[bincode], img)

    targetFrames.append(frame(bincode, img, ssim, psnr))

else:
    break

cap.release()

lines = list()

lines.append("bincode;ssim;psnr\n")

for f in targetFrames:
    lines.append("{};{};{}\n".format(f.bincode, f.ssim, f.psnr))

fid = open(FILENAME, "w")
fid.writelines(lines)
fid.close()

print 'Done.'
```