

ANÁLISE DE QoS EM REDES DEFINIDAS POR SOFTWARE E REDES TRADICIONAIS

Arich Andrade Rocha¹, Claudio de Castro Monteiro¹

¹Instituto Federal de Educação, Ciências e Tecnologia do Tocantins – IFTO – Campus Palmas, e-mail: <arichandrade@gmail.com, ccm@ifto.edu.br>

Resumo: O presente trabalho proporciona comparações e conclusões de dados realizados por meio de testbed. Foram realizadas sequências de requisições e as coletas dos dados, nos quais são armazenadas informações de métricas Q.o.S (Jitter, Atraso e Perda). Com esses dados foram possíveis analisar os resultados obtidos que serão apresentados no decorrer deste trabalho. Concluiu-se que, a qualidade de serviço está ligada diretamente com a qualidade de transmissão de arquivos pela rede.

Palavras-chave: Q.o.S, testbed, Qualidade de Serviço

1 INTRODUÇÃO

Com o avanço tecnológico, o usuário passou a ser mais exigente na velocidade das informações obtidas na internet ou até mesmo na intranet. Com a grande massa de dados espalhados na internet, têm-se disponível uma gama de conteúdos, onde os mesmos podem variar de tamanhos.

O meio pelo qual os conteúdos percorrem é o fator determinante para o rápido acesso a esses dados. A Qualidade de Serviço (*Quality of Service - Q.o.S*) requer melhores condições do tráfego, devido à velocidade de acesso dos mesmos. Os parâmetros (*Jitter*, Atraso, Perda) são usados para medir as condições do tráfego na rede. Através dos valores obtidos é possível determinar se o serviço fornecido ao cliente tem uma qualidade aceitável.

Este artigo propõe avaliar os resultados de um estudo realizado em três ambientes diferentes. A proposta é realizar uma sequência de transmissões de arquivos em uma rede virtual, em uma rede virtual integrada com uma rede física e somente em rede física. Foi montado um teste capaz de avaliar a variação dos parâmetros da qualidade de serviço enquanto era realizada transferência de arquivos, requisitados de cliente-servidor. Os resultados visa concluir os impactos sofridos de acordo com a topologia usada.

O texto está organizado da seguinte forma: Na seção 2, os trabalhos relacionados são discutidos. A seção 3 apresenta a Proposta. A Seção 4 é apresentado a Metodologia de Trabalho. Na seção 5 Resultados e Discussões. Por fim, na seção 6 as Considerações Finais.

2 TRABALHOS RELACIONADOS

No trabalho [Ribeiro Araujo] os autores propõem um *testbed* que tem a função de realizar sequências de transmissões e coletas de dados para que se possam medir métricas de QoS e QoV. O objetivo é automatizar as transmissões multimídia para analisar o comportamento da rede durante o período de transmissão de cada stream. Assim 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.

Em [Ferreira] realiza experimento em transmissão de vídeo em redes 802.11, mais com o objetivo de analisar a qualidade do vídeo transmitido. São analisados os parâmetros de qualidade de serviço *Q.o.S*, qualidade de experiência *Q.o.E* com seguintes requisitos Throughput, Atraso, *Jitter*, Porcentagem de pacotes perdidos e PSNR. Chegando à conclusão que qualidade de serviço está diretamente proporcional ligada à qualidade do vídeo, isso porque ao analisar os resultados

obtidos se observou que a qualidade do vídeo degradava, proporcionalmente, as métricas de qualidade de serviço.

Em [Egilméz] propõe demonstrar o desempenho de OpenQoS em uma rede real. O OpenQoS se baseia em um esquema de priorização de roteamento, ou seja, OpenQoS pode garantir a entrega de vídeo com perfeita, pouca ou nenhuma perturbação. Tenho melhor satisfação pelos usuários finais, mesmo usando um protocolo de transporte não confiável, como *UDP*, e um protocolo de transporte confiável, como o *TCP*, o OpenQoS pode garantir a qualidade do vídeo completo. Como resultado o OpenQoS se mostrou com bons resultados nas transmissões de vídeos.

Em [Kamienski] define o componente para a determinar um modelo *Q.o.S*, onde o tipo de tráfego que as aplicações gerem e ao tipo de comportamento esperado da rede, para que elas funcionem corretamente. As aplicações podem ser classificadas das seguintes formas:

a) Aplicações elásticas (não tempo real), a recepção correta dos dados é mais importante, ou seja, são completamente transferidas e só depois reproduzidas. Exemplos de aplicações elásticas são correio eletrônico, página Web, transferência de arquivos, tais como *File Transfer Protocol (FTP)* e compartilhamento de arquivos online.

Esses tipos de aplicações não são sensíveis aos fatores como, variação dos atrasos, variação de largura de banda, perdas, entre outros. Apesar dos mesmos afetarem o seu desempenho.

b) Aplicações de tempo real: Garantia do isocronismo dos dados gerados, ou seja, o fluxo de dados é transportado pela rede até seu destino, onde é reproduzido pela aplicação receptora antes de serem completamente transferidas. Exemplos de aplicações de tempo real são *stream* de reprodução de vídeos sobre redes IP.

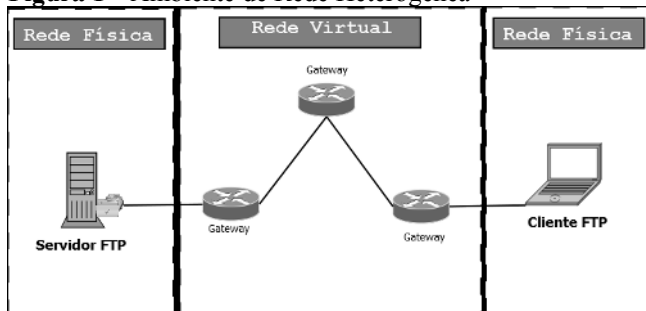
Esse tipo de aplicação é extremamente sensível à variação dos atrasos, perdas de pacotes, largura de banda, etc.

3 PROPOSTA

A motivação deste trabalho é proporcionar ao pesquisador a comparação e a conclusão dos diferentes dados coletados. O *testbed* teria a função de realizar as sequências de requisições e as coletas dos dados, fase que consomem bastante tempo. Dentro das coletas de dados são armazenadas informações de métricas *Q.o.S* (*Jitter*, Atraso e Perda). Com esses 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 infraestrutura física possui algumas diferenças dependendo do trabalho realizado.

Figura 1 – Ambiente de Rede Heterogênea



Fonte: Próprio Autor.

No ambiente temos um servidor de arquivos que disponibiliza 3 conteúdos. Os conteúdos são de tamanhos diferentes, primeiro de 10Mb, o segundo de 32Mb e o terceiro de 72Mb. O conteúdo é disponibilizado utilizando um servidor de arquivos (*FTP*).

Nos experimentos foram criados 4 redes. No primeiro *testbed* foram criados

exclusivamente na ferramenta NetKit, onde disponibiliza um ambiente para experimentos com redes de computadores em que se enfatizam configurações básicas de rede, criação de infraestrutura WAN com MPLS, e roteamento. Ele é composto de máquinas virtuais Linux (implementadas com kernel Linux UML – *User Mode Linux*), que funciona como roteadores ou computadores, switches Ethernet virtuais para interligar as mesmas. O Netkit proporciona máquinas virtuais como máquinas reais, possuindo uma ou mais interfaces de rede. Através destes recursos é possível criar redes de configurações arbitrárias para estudar mecanismos e protocolos de comunicação.

Já no segundo *testbed* foi criado uma junção entre rede virtual e uma rede física. O cliente realizava uma requisição e o mesmo partia de uma rede física, passando por uma rede virtual criada no NetKit e retornava para rede física onde o servidor FTP estava disponível. A partir daí nasceu o conceito de Redes Definidas por Software (*Software Defined Network – SDN*). A mesclagem de redes virtuais e rede física.

E por último foi realizado o *testbed* em uma rede integralmente física. Os demais programas que auxiliaram a execução dos experimentos foram: Script de requisição e captura dos dados feitos em linguagem em *Python*, *GNU PLOT* para criação dos gráficos a partir dos resultados obtidos. As interfaces de comunicação, endereços *IP's* e rotas foram definidas manualmente de cada máquina do ambiente.

Os *testbed* foram realizados em sequência. Primeiramente o *host* cliente realizava a requisição do arquivo de menor tamanho, em seguida do arquivo superior a ele e por último o arquivo de maior tamanho. Todas as solicitações feitas pelo *host* cliente para o servidor *FTP*, foram de três saltos, ou seja, passando pelos três gateway's.

4 METODOLOGIA DE TRABALHO

O objetivo do *testbed* é verificar o desempenho da rede através das métricas *Q.o.S* (*Jitter*, Atraso e Perda) durante a transmissão de cada arquivo requisitado. Portanto, deve-se armazenar as informações que permitam calcular os mesmos de cada sequência de requisições. Considerando uma população infinita, foi definido o intervalo de confiança de 95% com uma sequência de 100 repetições, assim podendo definir uma amostra consideravelmente grande, acarretando resultados confiáveis.

Foram utilizados ao total 5 hosts físicos para a realização do *testbed*. Todos eles possuem as seguintes configurações.

Tabela 1 – Descrição dos hosts utilizados no *testbed*

| Marca | Modelo | Sistema Operacional | Configurações |
|---------|---------|---------------------|---|
| Itautec | Desktop | Ubuntu 16.04.2 LTS | Proc.: Intel Core i5, Memória: 4GB RAM, HD: 1Tb |

Fonte: Próprio Autor.

Com o ambiente em funcionamento, foi possível planejar, testar ou analisar a ordem dos experimentos proposto. No modelo apresentado, planejamos uma sequência de transmissões de arquivos, conseqüentemente uma série de repetições de entradas de dados foram feitas, fazendo necessário acompanhar todo o processo de transferência de arquivos. Para isto, foi utilizado dois scripts em *python*. O primeiro realiza as requisições dos arquivos ao servidor *FTP* e armazena os mesmos no próprio *host*. Em paralelo outro script *python* que captura e armazenando as métricas (*Jitter*, Atraso e Perda), tornando possível avaliar o desempenho da rede durante as transmissões.

As sequências de execuções dos scripts são feitas a partir da entrada de dados de quem a executa. O script *requisição.py* necessita digitar o *IP* do servidor *FTP* e o nome do arquivo a ser requisitado, já o script *qos.py* necessita digitar o *IP* do servidor *FTP*. Com os resultados das métricas salvas, usa-se um utilitário gráfico, cujo nome é *Gnuplot*, para tornar os dados salvos

Tabela 2 – Scripts e utilitários utilizados na execução do *testbed*

| Script | Descrição |
|---------------|--|
| Requisicao.py | Script usado para requisição dos arquivos do servidor FTP. |
| Qos.py | Script usado para coletar as métricas de <i>Q.o.S</i> (<i>Jitter</i> , <i>Atraso</i> e <i>Perda</i>) |
| Gnuplot | Utilitário gráfico |

Fonte: Próprio Autor.

5 RESULTADOS E DISCUSSÕES

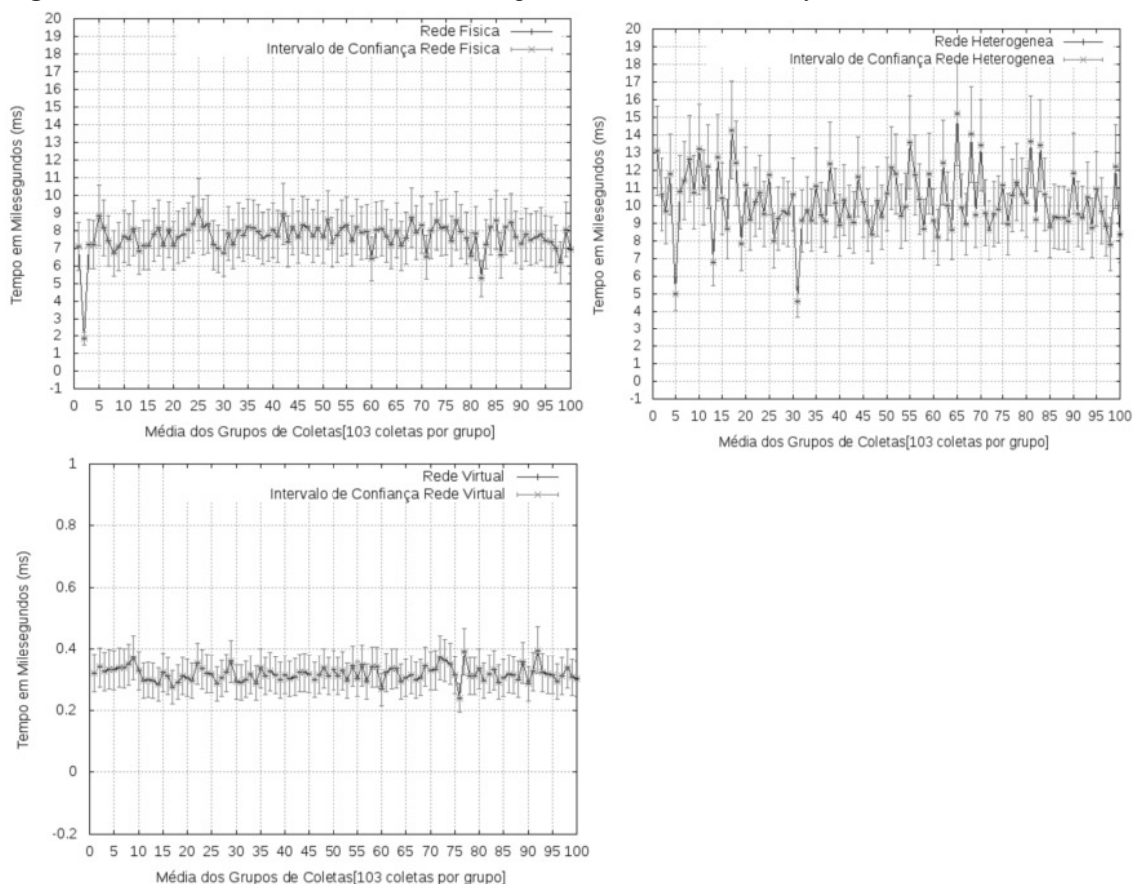
Os resultados obtidos nos testes após serem analisados, tornaram possíveis a obtenção dos valores necessários para avaliação das transmissões. Em nosso modelo, foi usada uma sequência de requisições repetitivas.

Os resultados são mostrados graficamente a seguir, comparando as médias com seus intervalos de confiança de cada transferência do arquivo de vídeo nos ambientes propostos.

- **Transferência de Arquivo Akyo, Métrica Atraso**

Pode-se analisar que a variação sofrida na métrica de atraso nos três ambientes propostos. Chegando à conclusão que na topologia heterogênea, houve maior variação da métrica. Diferentemente da topologia virtual, que houve menos entre 0 e 1.

Figura 2 – Métrica Atraso. Rede Física, Heterogênea e Virtual – Vídeo Akyo



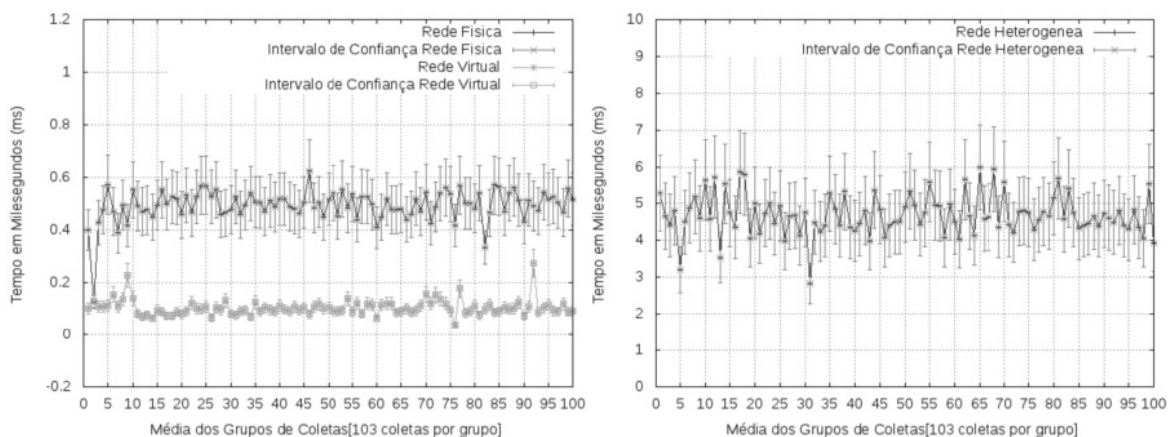
Fonte: Próprio Autor.

- **Transferência de Arquivo Akyo, Métrica Jitter**

Ao analisar os resultados do *jitter*, podemos notar que a transferência de arquivo nas

topologias usadas na rede física e rede virtual obteve pouca variação, não ultrapassando 1 Milissegundos (ms). Já a topologia heterogênea a variação foi bem maior que as duas anteriores.

Figura 3 – Métrica *Jitter*. Rede Física, Virtual e Heterogênea – Vídeo Akyo

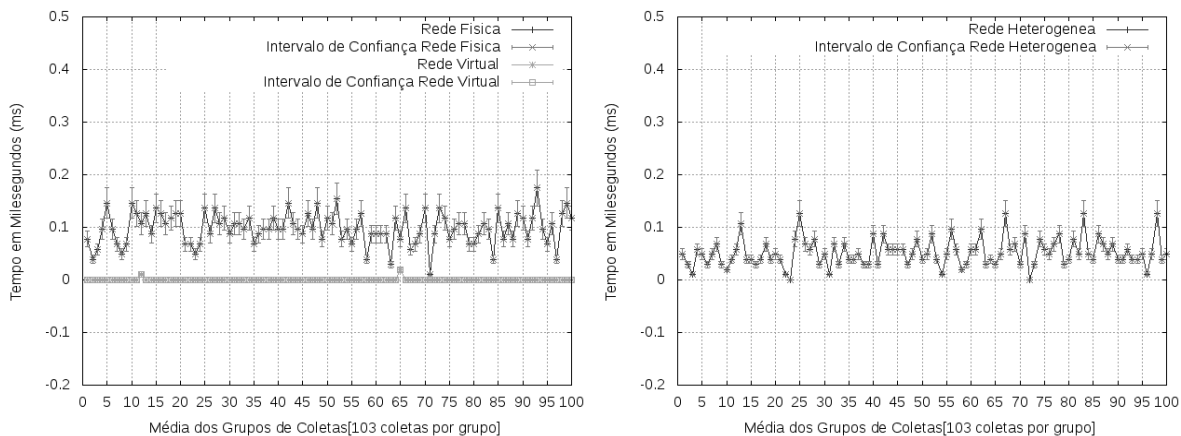


Fonte: Próprio Autor.

- **Transferência de Arquivo Akyo, Métrica Perda**

Conforme os resultados graficamente mostrados, as três topologias obtiveram uma baixa variação na perda de seus pacotes. Variando entre 0 a 0.2 Milissegundo (ms). A rede virtual destaca-se por não sofrer variação.

Figura 4 – Métrica Perda. Rede Física, Virtual e Heterogênea – Vídeo Akyo

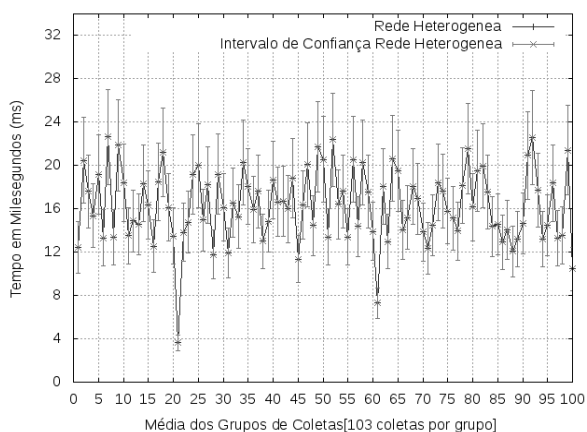
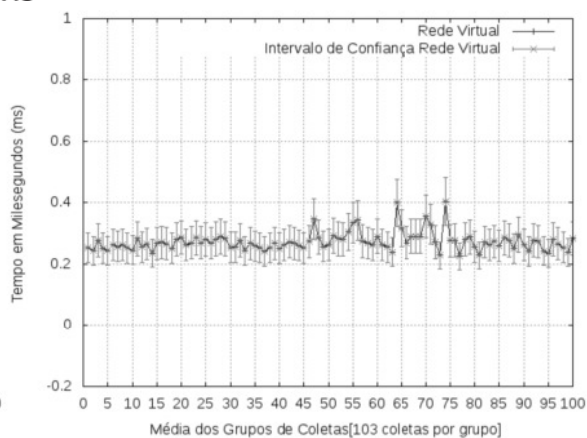
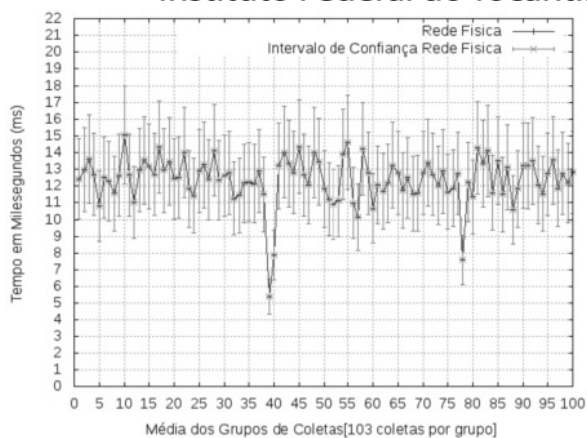


Fonte: Próprio Autor.

- **Transferência de Arquivo Grandman, Métrica Atraso**

Semelhantemente aos resultados do arquivo Akyo, a topologia heterogênea obteve maior variação na métrica de atraso. Por outro lado, a variação na topologia virtual continuou nos intervalos de 0 a 1 Milissegundo (ms) apesar do arquivo Grandman por ser maior que o arquivo Akyo.

Figura 5 – Métrica Atraso. Rede Física, Virtual e Heterogênea – Vídeo Grandman

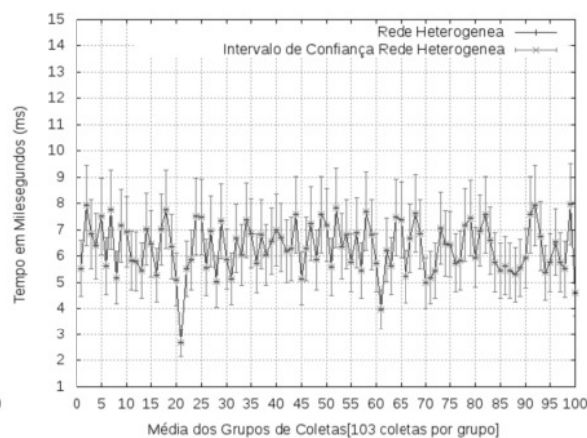
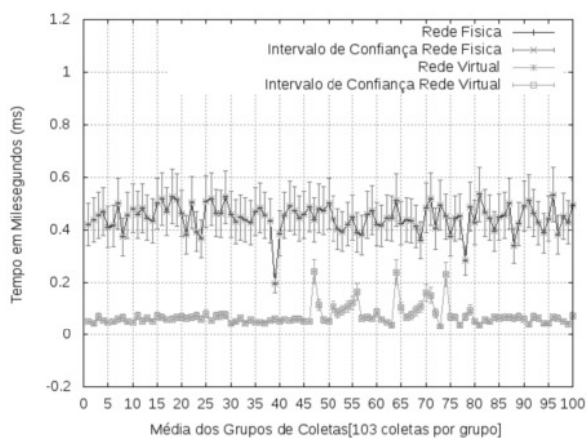


Fonte: Próprio Autor.

- **Transferência de Arquivo Grandman, Métrica *Jitter***

Observando os gráficos, as topologias de rede física e rede virtual, houve pouca variação na métrica *jitter*. Semelhante aos resultados na transferência de arquivo do vídeo Akyo, não ultrapassando 1 Milissegundo (ms).

Figura 5 – Métrica *Jitter*. Rede Física, Virtual e Heterogênea – Vídeo Grandman



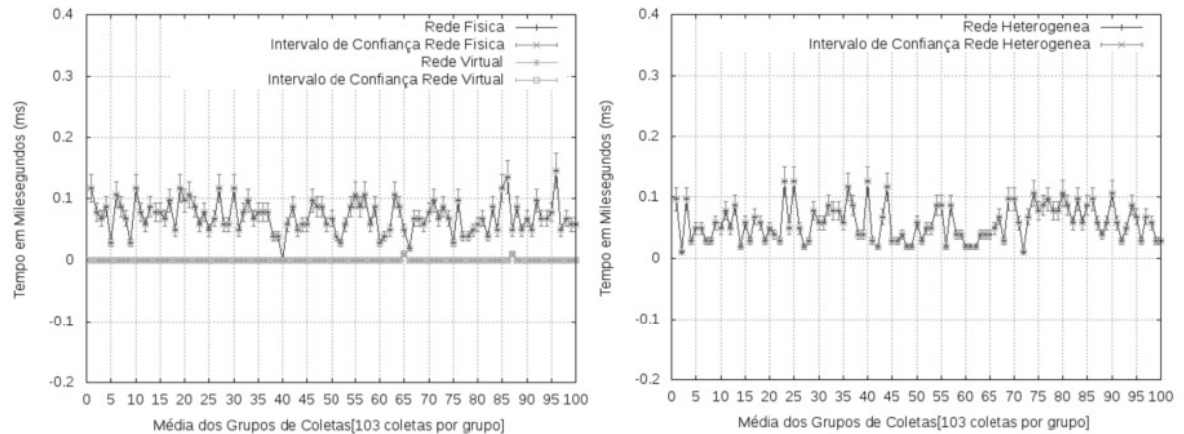
Fonte: Próprio Autor.

- **Transferência de Arquivo Grandman, Métrica Perda**

Podemos observar que as topologias rede física e rede heterogênea obtiveram semelhanças na variação da métrica perda. Por outro lado a rede virtual praticamente não teve

oscilações.

Figura 6 – Métrica Perda. Rede Física, Virtual e Heterogênea – Vídeo Grandman

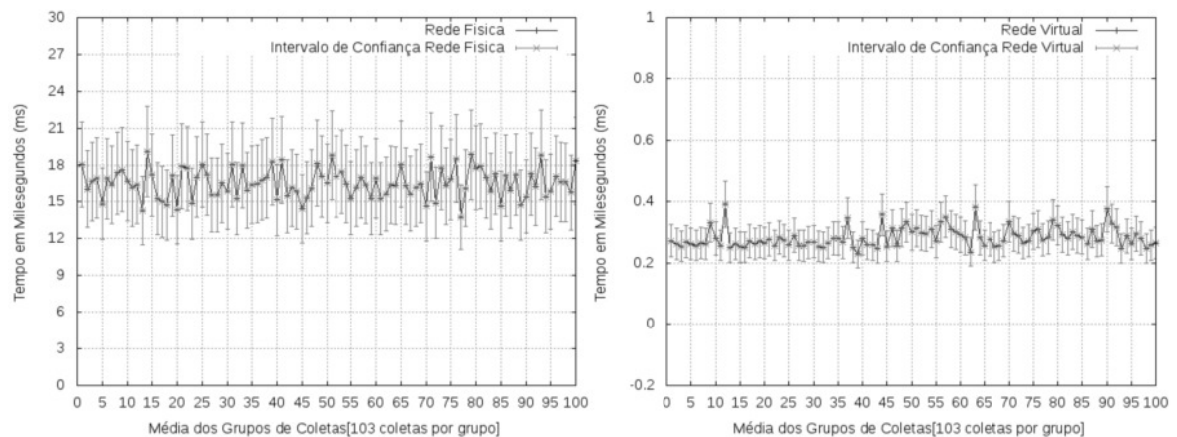


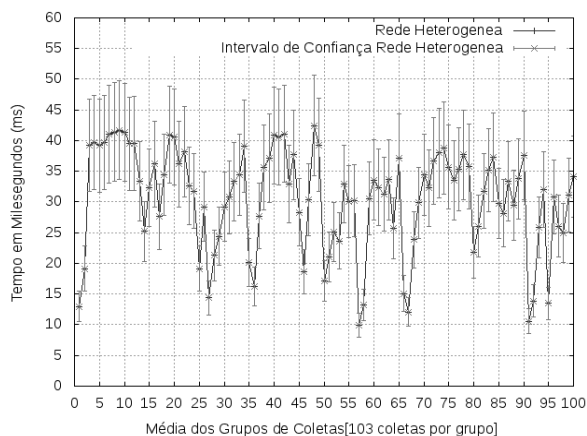
Fonte: Próprio Autor.

- **Transferência de Arquivo Highway, Métrica Atraso**

Como podemos observar, o arquivo highway por ser de maior tamanho, exigiu mais da rede. A topologia de rede heterogênea mostrou um alto índice de variação das médias podendo ultrapassar os 40 Milissegundos, diferentemente da topologia rede física que não ultrapassou os 21 Milissegundos. Por outro lado a rede virtual não ultrapassou 1 Milissegundos, semelhantemente aos arquivo akyo e grandman.

Figura 7 – Métrica Atraso. Rede Física, Virtual e Heterogênea – Vídeo Highway



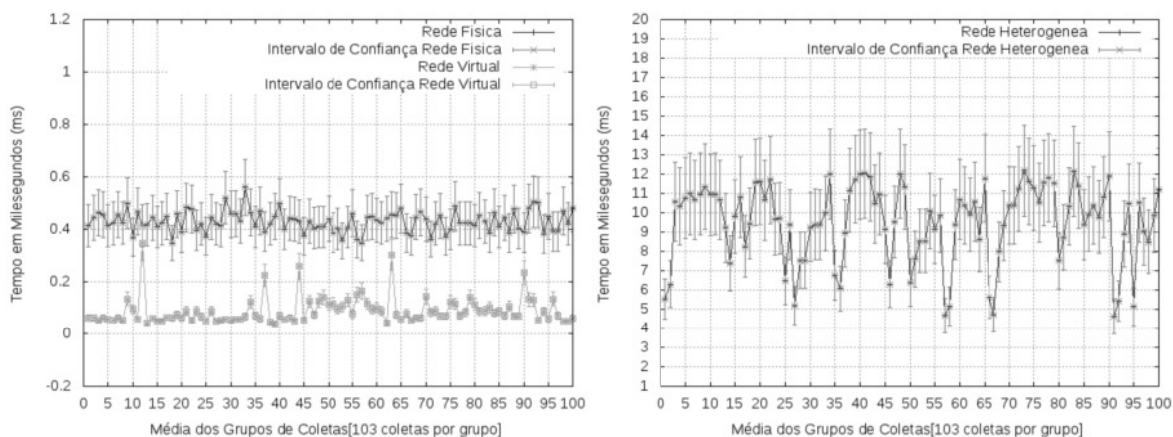


Fonte: Próprio Autor.

- **Transferência de Arquivo Highway, Métrica Jitter**

Observa-se que a topologia heterogênea obteve uma grande variação em relação as outras duas topologias anteriores. Onde as topologias rede física e rede virtual não ultrapassaram os 1 Milissegundo.

Figura 8 – Métrica Jitter. Rede Física, Virtual e Heterogênea – Vídeo Highway

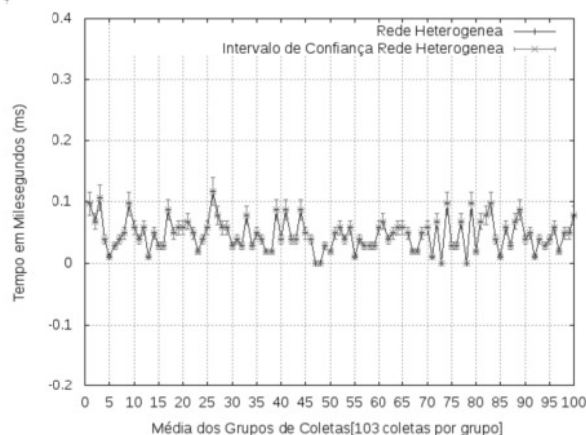
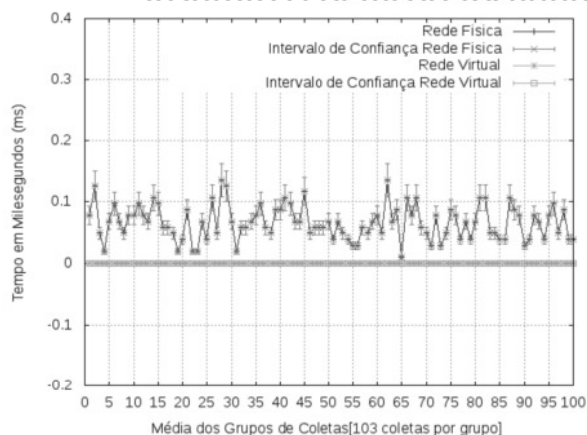


Fonte: Próprio Autor.

- **Transferência de Arquivo Highway, Métrica Perda**

Todas as topologias, tiveram pouca variação, ao se destacar a topologia rede física por esta constantemente ao 0 Milissegundos.

Figura 9 – Métrica Perda. Rede Física, Virtual e Heterogênea – Vídeo Highway



Fonte: Próprio Autor.

6 CONSIDERAÇÕES FINAIS

Este trabalho abordou o impacto das métricas de *QoS* nas transmissões de arquivos em três ambientes de rede diferente. Pode ser observado que a qualidade de serviço está proporcionalmente ligada ao meio pelo qual os arquivos são transmitidos, isso porque ao analisar os resultados, observou que a média de transferência aumenta de acordo com a diversificação de conexões de rede.

Pode-se constatar que as transmissões, sofrem mais alterações com as redes heterogêneas, onde o trajeto percorrido de uma rede física para uma rede virtual e retornando para a rede física, mostra a exigência da rede para que a transferência não tenha perda de qualidade.

Outro fator chave a se destacar é nas limitações encontradas no NetKit. O mesmo não é possível configurar dois túneis, assim a solução encontrada foi usar o *OpenVpn* para o 3º (terceiro) *gateway* se comunicar com o servidor *FTP*.

Conclui-se então, que a qualidade de serviço impacta diretamente a qualidade de transmissão de arquivos pela rede, devido ao meio pelo o qual percorre. Quanto maior a interconectividade de redes diferentes, maior é a variação das métricas de qualidade de serviço.

Para trabalhos futuros propõe-se automatizar as coletas de dados, realizar testes em redes sem fio, usando software para perturbação na rede e limitação de banda, assim tornando os resultados mais realistas.

REFERÊNCIAS

dos Passos Silva, Diego, et al. "Uma Arquitetura para o Aprovisionamento de *QoS* Interdomínios em Redes Virtuais baseadas no *OpenFlow*." *Proc. of the 31st Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)* (2013): 893-906.

Egilmez, Hilmi E., et al. "OpenQoS: An *OpenFlow* controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks." *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*. IEEE, 2012.

Ferreira¹, Antonio Marcos Almeida, and Claudio de Castro Monteiro. "TRANSMISSÃO DE STREAMS DE VIDEO EM REDES 802.11."



Kamienski, Carlos Alberto, and Djamel Sadok. "**Qualidade de Serviço na Internet.**" *mini-curso, 18o SBRC, Belo Horizonte/MG* (2000).

Ribeiro Araujo, Jean Nunes, Carlos Soares Noletto Junior, and Claudio de Castro Monteiro. "**Testbed para análise de QoS/QoV sobre redes ip.**" *VII CONNEPI-Congresso Norte Nordeste de Pesquisa e Inovação*. 2012.