



Avaliação de sistema de *streaming* de vídeo com *cache* multi-nível

L. Y. O. Omi R Immich E. R. M. Madiera

Relatório Técnico - IC-PFG-18-38

Projeto Final de Graduação

2018 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

<i>Avaliação de vídeo</i>	1
---------------------------	---

Sumário

1	Introdução	2
2	Metodologia	3
2.1	Ferramentas utilizadas	3
2.2	Adaptação do Evalvid para o OMNeT++	3
2.3	Aplicação de <i>streaming</i> de vídeo com <i>cache</i>	3
2.4	Cenário de simulação	4
3	Resultados	6
4	Conclusões	9
	Referências Bibliográficas	9

Avaliação de sistema de *streaming* de vídeo com *cache* multi-nível

Leo Yuuuki Omi Omi, Roger Immich, Edmundo Roberto Mauro Madeira

Resumo

Este trabalho visa analisar os benefícios da utilização de *caches* em diferentes camadas da rede para um sistema de *streaming* de vídeo. Aproximar o conteúdo ao dispositivo final traz vantagens, no entanto os custos e complexidade da implantação de *caches* podem ser altos, portanto uma análise dos efeitos do cache pode ser útil para a decisão de introduzir um sistema deste tipo. Além disso, a quantidade de dispositivos móveis continua crescendo e o consumo de vídeo também. Assim, uma simulação de uma rede com dispositivos móveis utilizando LTE foi executada, consumindo um serviço de vídeo com um servidor principal na *Cloud* e *caches* na *Fog* e na *Edge* e os resultados foram analisados com um foco na *Quality of Experience*.

1 Introdução

Com a crescente utilização de serviços de vídeo como YouTube e Netflix, e de dispositivos móveis como *smartphones*, a maneira que o vídeo é transmitido pela Internet precisa ser otimizada [8, 7]. É previsto que até 2021 o tráfego de *smartphones* vai exceder o de computadores pessoais e que 81% do tráfego de toda a internet será de vídeo, superando os 73% de 2016 [6]. Uma maneira de otimizar a entrega destes dados é utilizar, em conjunto, a *Cloud*, *Fog* e *Edge* com *caches* nos diferentes níveis. Apesar da *Cloud* possuir uma enorme capacidade computacional e ser conveniente por centralizar o conteúdo em um lugar, a *Cloud* traz o problema de potencialmente estar distante dos dispositivos finais e, consequentemente, apresenta um maior atraso para a entrega de pacotes e uma maior perda de pacotes [5]. Assim, a *Fog* e a *Edge* podem melhorar a transmissão do vídeo por estarem mais próximas.

O objetivo deste trabalho é avaliar o efeito na *Quality of Experience* da utilização de *caches* na *Fog* ou na *Edge* para um sistema de *streaming* de vídeo. A proximidade ao dispositivo final deve trazer vantagens, mas em situações reais, é necessário considerar o custo e a complexidade da implementação de um sistema deste tipo, já que servidores de *cache* distribuídos em inúmeros lugares precisam ser instalados, além do fato do conteúdo precisar ser sincronizado nas várias máquinas, o que aumenta a complexidade do sistema. Portanto, os resultados aqui apresentados poderão auxiliar na decisão de introduzir esses tipos de *caches*.

2 Metodologia

2.1 Ferramentas utilizadas

Para simular a transmissão de vídeo em uma rede, foram utilizadas duas ferramentas principais. O OMNeT++ [10], na versão 5.4.1, um simulador de eventos discretos geralmente utilizado para simulações de redes comunicação. E o Evalvid [9], um conjunto de ferramentas para avaliação da qualidade de vídeos transmitidos em redes reais ou simuladas.

O OMNeT++ por si só possui apenas implementações básicas para simulações em redes e não possui implementações de protocolos frequentemente utilizadas (TCP, UDP, Ethernet, etc.). Portanto, foi utilizado o INET Framework [1] na versão 3.4, que implementa muitos destes protocolos. Como o foco deste trabalho é a transmissão de vídeo para dispositivos móveis, foi necessário usar o SimuLTE [2] na versão 0.9.1, um *framework* que estende o INET Framework oferecendo uma implementação do padrão de comunicação móvel LTE.

O Evalvid foi utilizado para 3 propósitos:

- A geração de *traces* que descrevem como um vídeo é transmitido pela rede.
- A reconstrução do vídeo a partir de *traces* gerados pela simulação.
- Avaliação da qualidade do vídeo utilizando a métrica *structural similarity* (SSIM).

O software MSU VQMT [11] foi utilizado para a métrica *Peak signal-to-noise ratio* (PSNR) para comparação com os resultados SSIM do Evalvid.

Para executar as simulações, calcular as métricas de qualidade, e plotar os gráficos, foi escrita uma série de *scripts* em Python para automatizar o processo.

2.2 Adaptação do Evalvid para o OMNeT++

Tanto o INET Framework como o SimuLTE possuem implementações de aplicações de *streaming* de vídeo, no entanto a aplicação do INET é muito simples e a do SimuLTE não está documentada. Além disso, nenhuma delas seria compatível com o Evalvid. Portanto, foi desenvolvida uma aplicação servidor e uma aplicação cliente para serem executadas em cima do INET Framework compatível com o Evalvid.

A aplicação servidor utiliza os *traces* do Evalvid, que descrevem o tamanho e a quantidade de pacotes necessários para a transmissão de cada quadro do vídeo. Após receber a requisição de um cliente, o servidor envia pacotes de acordo com a descrição dos *traces* do Evalvid, e gera um *trace* de envio, que descreve quais pacotes foram enviados para o cliente. A aplicação cliente envia uma requisição e aguarda os pacotes enviados pelo servidor e gera *traces* de recebimento de acordo com os pacotes recebidos. Os *traces* de envio e de recebimento são utilizados pelo Evalvid para reconstruir o vídeo da maneira que o cliente o teria reproduzido.

2.3 Aplicação de *streaming* de vídeo com *cache*

Para o propósito deste trabalho, foi necessário adaptar a aplicação de vídeo para a simulação de *caches*. Para tal, bastou adaptar o funcionamento das requisições para as aplicações

servidor e cliente.

A aplicação servidor pode representar o servidor principal do sistema que sempre contém o conteúdo do vídeo ou pode ser um servidor de *cache* que pode conter o conteúdo do vídeo, e caso não contenha, conhece o endereço do próximo nível de *cache*. A aplicação cliente simplesmente envia pacotes de requisição para a aplicação servidor e recebe os pacotes de vídeo.

Como ilustrado na Figura 1, a aplicação cliente faz uma requisição ao servidor mais próximo, caso o servidor tiver o conteúdo a ser transmitido, a transmissão é iniciada. Caso contrário, o servidor envia o endereço do próximo nível do *cache* e o cliente repete o processo. No caso ilustrado, ambos os servidores de *caches* não possuem o conteúdo e redirecionam o dispositivo para o próximo nível.

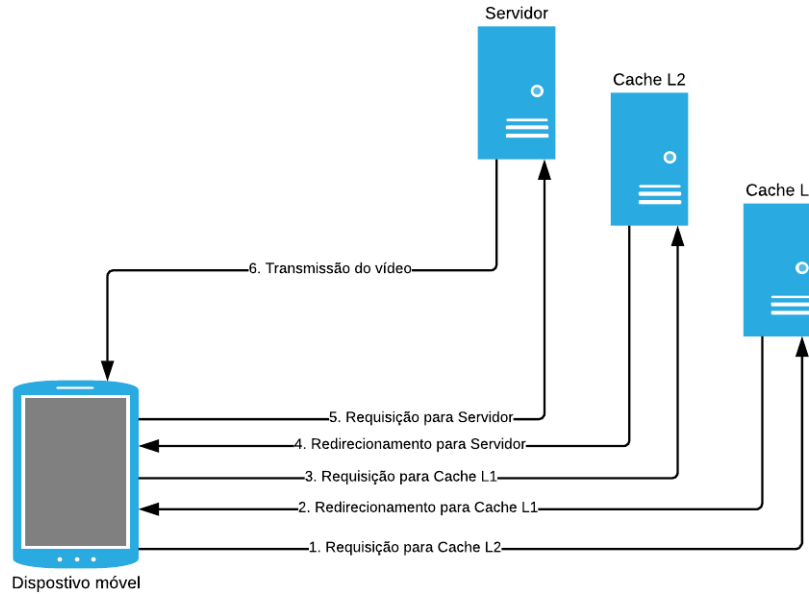


Figura 1: Diagrama do fluxo das requisições do sistema.

2.4 Cenário de simulação

Idealmente, simulações utilizando uma configuração semelhante à ilustrada na Figura 2 teriam que ser criadas. No entanto, devido ao tempo desta disciplina e à complexidade dessa configuração, foi decidido utilizar apenas a parte destacada em vermelho. Apesar de mais simples, esta configuração ainda pode trazer resultados compatíveis com os alcançados em cenários realistas.

Na Figura 3, está ilustrada a configuração do cenário base da simulação. Temos o nó servidor (*server*), que representa um computador na *Cloud*, o nó de cache nível 1 (*l1cache*), que representa uma aplicação de cache do conteúdo em algum local geograficamente mais próximo dos usuário e o nó de cache nível 2 (*l2cache*) que representa um servidor de cache

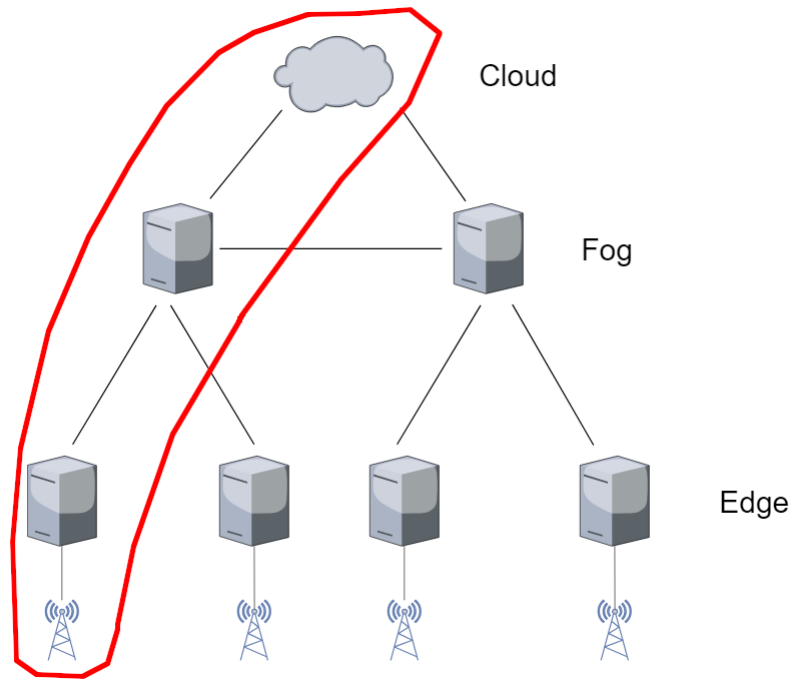


Figura 2: Diagrama da arquitetura simplificada da rede

instalado muito próxima à estação base que serve os dispositivos móveis.

Para configurar os atrasos da conexão entre os nós *router0* e *router1* e da conexão entre os nós *router1* e *router2* foram utilizados dados da página *Network Test* [3]. Esta página executa uma ferramenta no dispositivo que a acesse, fazendo medições de atraso para servidores espalhados pelo mundo. Portanto, foram utilizados dados reais tornando a simulação mais realística. Foram coletados os atrasos de três *Content Delivery Networks* (CDNs) e para cada CDN foi calculada a média de 3 amostras.

Os atrasos utilizados foram:

- Akamai CDN: 8ms
- Quantil: 28ms
- IBM Cloud: 121ms
- Swift Serve: 406ms

Os cenários utilizados com os atrasos entre *router0* e *router1* e entre *router1* e *router2* foram, respectivamente:

- Cenário 1: 8ms e 28ms
- Cenário 2: 8ms e 121ms



Figura 3: Imagem do cenário base da simulação no ambiente gráfico do OMNeT++

- Cenário 3: 28ms e 406ms

Para cada cenário foram criados três subcenários: um apenas com o servidor principal com o conteúdo, um com o *cache* nível 1 com o conteúdo e o último com o *cache* nível 2 com o conteúdo.

Para cada subcenário, a simulação foi executada com o *streaming* de 4 trechos de 15 segundos retirados dos filmes *Big Buck Bunny* e *Elephant Dreams* [4], nas resoluções 480p, 720p e 1080p.

3 Resultados

A primeira métrica avaliada, demonstrada na Figura 4, é a porcentagem de pacotes recebidos pelos dispositivos finais. Nas simulações executadas, a falta dos dados dos pacotes é o principal fator para a queda de qualidade no vídeo resultante. A perda de pacotes pode ser causada por um congestionamento na rede, ou por uma sobrecarga da estação-base, já que há uma queda da porcentagem com o aumento de dispositivos na rede. Outro fator é a qualidade da conexão com o servidor utilizado, havendo menor perda de pacotes com a utilização de caches ou servidores mais próximos.

O PSNR, ilustrado na Figura 5, é uma métrica que utiliza o erro quadrático médio (EQM) pixel a pixel da imagem, o que pode não corresponder muito bem à percepção do ser humano. No entanto, ainda é uma métrica que pode ser utilizada para a avaliação de

Porcentagem de pacotes recebidos pelo número de dispositivos móveis

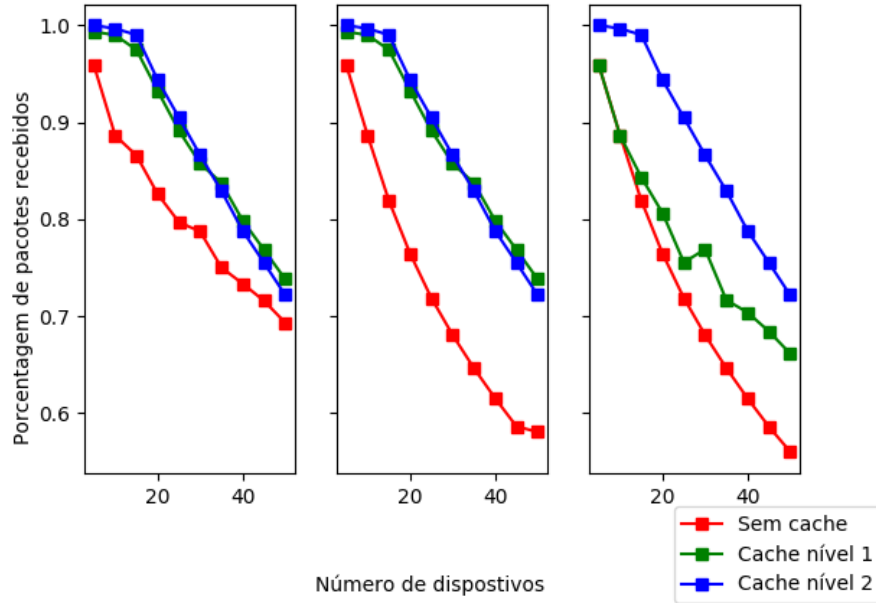


Figura 4: Gráfico da porcentagem de pacotes recebidos em cada cenário pelo número de dispositivos

qualidade de vídeo e podemos utilizá-la como comparação com o SSIM, da Figura 6, que é uma métrica considerada mais próxima da percepção humana por utilizar informações estruturais da imagem [12].

Podemos ver que os resultados estão consistentes entre si. Nos cenários com maior perda de pacotes, também ocorreram PSNR e SSIM menores na média. E em geral, as curvas tem formas semelhantes.

As curvas da Figura 5 são bem similares às curvas da Figura 4. Este resultado pode ter ocorrido pelo fato de que o PSNR utiliza o EQM entre as imagens do vídeo, e a quantidade de pacotes recebidos tem um efeito direto nessa métrica. Mas no caso das Figuras 6 e 4 o mesmo não ocorre, já que SSIM utiliza informações estruturais da imagem e a perda de dados por ter efeitos maiores na estrutura da imagem, o que explica a diminuição mais brusca desta métrica.

No Cenário 1, com poucos dispositivos, o cache consegue oferecer um serviço melhor. Mas o SSIM nos 3 subcenários alcança valores muito baixos e próximos quando o número de dispositivos na rede chega a 50. Nestes casos, o que pode ter ocorrido é que a rede chega a ser tão sobrecarregada que nem mesmo os *caches* conseguem fazer uma diferença.

No Cenário 2, o atraso para o nível 2 é muito maior (de 28ms para 121ms), e podemos ver que houve uma diferença no SSIM médio para o subcenário sem cache comparado ao Cenário 1. Apesar da diferença não ser muito grande, podemos ver que em um caso como

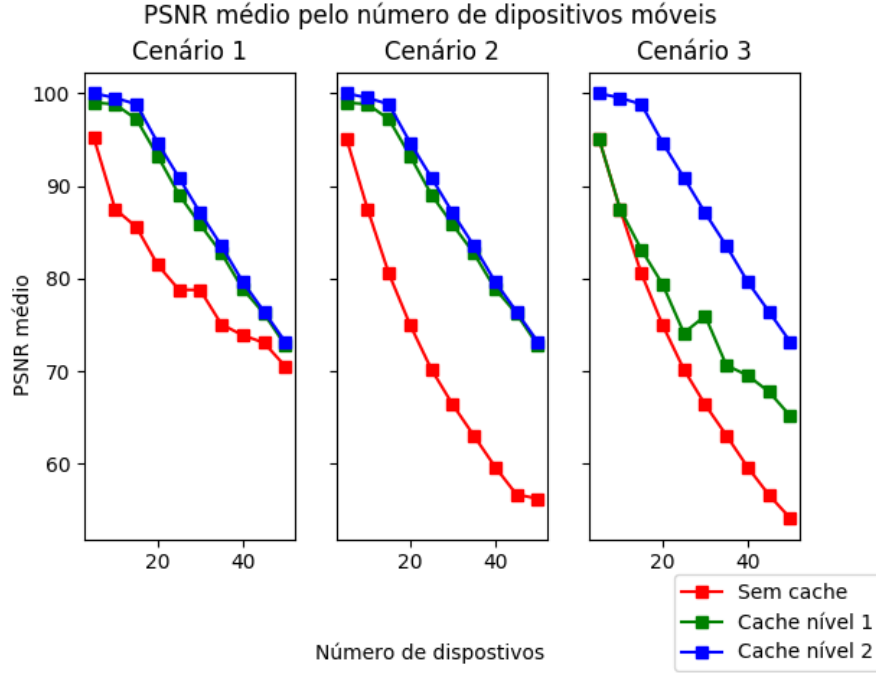


Figura 5: Gráfico do PSNR médio em cada cenário pelo número de dispositivos

esse, a utilização de *caches* na *Fog* ou *Edge* pode ser mais vantajosa do que no Cenário 1 mesmo com muitos usuários.

O Cenário 3 se diferencia dos outros cenários por causa do maior atraso no segundo nível (de 8ms para 28ms), o mesmo utilizado para o subcenário sem cache do Cenário 1 e podemos ver que resultaram em curvas similares. O interessante destes resultados é que a diferença de 20ms para o nível 2 do *cache* causa uma grande diferença, especialmente nos casos em que há menos dispositivos na rede. Ou seja, a implantação de *cache* na *Fog* pode não ser interessante se o atraso até esse nível não for muito baixo. Ainda assim, o *cache* tanto no nível 1 quanto 2 ainda apresentam resultados melhores do que o subcenário sem cache.

De maneira geral, a quantidade de dispositivos consumindo o conteúdo é o maior fator para a diminuição dos pacotes recebidos e a qualidade do vídeo. No subcenário sem *cache*, o Cenário 3 tem um atraso muito maior, 406ms, comparado aos 121ms do Cenário 2, mas o resultado foi quase idêntico para os 2 gráficos.

Algo importante a se lembrar é que estes resultados são de um caso simplificado. Como mostrado na Figura 2, foi simulado uma pequena parte de uma rede. Portanto, um cenário real com *caches* na *Fog*, mas não na *Edge*, poderia ter resultados piores por terem muitos mais clientes a servir do que nestes cenários simulados. O mesmo pode ser dito para o caso de não haver nenhum *cache* e apenas a *Cloud* servir o vídeo na rede.

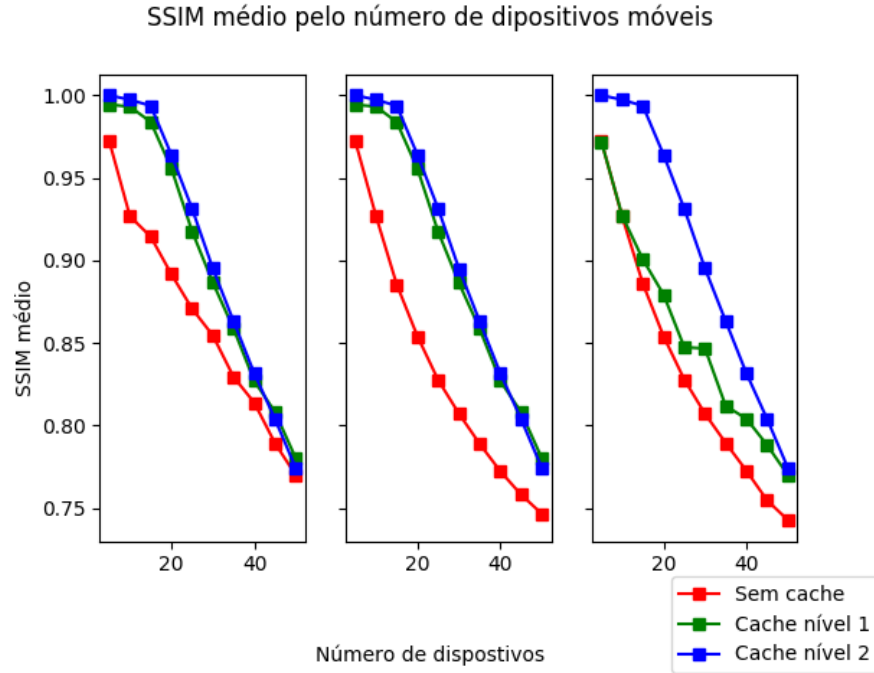


Figura 6: Gráfico do SSIM médio em cada cenário pelo número de dispositivos

4 Conclusões

A implantação de um sistema com *cache* multi-nível pode oferecer um serviço com melhor *Quality of Experience*. No entanto, benefícios significativos dependem muito das condições e estrutura da rede e, portanto, pode não ser uma boa solução para qualquer situação. A utilização de *cache* na *Edge* ou na *Fog* pode não ser interessante se o servidor principal estiver próximo. Mas caso não esteja e o *cache* estiver suficientemente próximo para oferecer um atraso muito pequeno, traz benefícios para qualidade do vídeo transmitido. Os resultados também indicam que a utilização de *caches* na *Edge* comparada à *Fog* é situacional para as características e capacidade da *Fog*.

Referências Bibliográficas

- [1] Inet framework. URL <https://inet.omnetpp.org/>. Acessado em 2018.11.18.
- [2] Lte user plane simulation model for inet omnet++. URL <http://simulte.com/>. Acessado em 2018.11.18.
- [3] *CDN Network Test*, 2018. URL <http://cloudharmony.com/speedtest-latency-for-cdn>. Acessado em 2018.11.17.

- [4] *YUV Video Sequences*, 2018. URL <http://trace.eas.asu.edu/yuv/index.html>. Acessado em 2018.11.16.
- [5] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, and O. Rana. The internet of things, fog and cloud continuum: Integration and challenges. *Internet of Things*, 3-4:134 – 155, 2018. ISSN 2542-6605. doi: <https://doi.org/10.1016/j.iot.2018.09.005>. URL <http://www.sciencedirect.com/science/article/pii/S2542660518300635>.
- [6] Cisco. White paper: Cisco VNI forecast and methodology, 2016-2021. Technical report, Cisco, September 2017. URL <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>.
- [7] M. Curado, H. Madeira, P. R. da Cunha, B. Cabral, D. P. Abreu, J. Barata, L. Roque, and R. Immich. *Internet of Things*, pages 381–401. Springer International Publishing, 2019. ISBN 978-3-319-77492-3. doi: 10.1007/978-3-319-77492-3_16. URL https://doi.org/10.1007/978-3-319-77492-3_16.
- [8] R. Immich, E. Cerqueira, and M. Curado. Efficient high-resolution video delivery over vanets. *Wireless Networks*, Feb 2018. ISSN 1572-8196. doi: 10.1007/s11276-018-1687-2. URL <https://doi.org/10.1007/s11276-018-1687-2>.
- [9] J. Klaue, B. Rathke, and A. Wolisz. Evalvid—a framework for video transmission and quality evaluation. *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 255–272, 2003.
- [10] A. Varga. Omnet++ discrete event simulator. URL <https://www.omnetpp.org/>. Acessado em 2018.11.18.
- [11] D. Vatolin, A. Moskin, O. Pretov, and N. Trunichkin. Msu video quality measurement tool, 2013. URL http://compression.ru/video/quality_measure/video_measurement_tool_en.html.
- [12] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 13(4):600–612, 2004.