

A Rede tem a Resposta: Um Modelo Escalável para Predição Integrada de Qualidade de Vídeo e QoE em Redes IP

Roberto Irajá Tavares da Costa Filho¹, William Lautenschläger¹,
Hugo Schroter Lazzari¹, Valter Roesler¹, Luciano Paschoal Gaspary¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{roberto.costa,wrlautenschlager,hslazzari,roesler,paschoal}@inf.ufrgs.br

Abstract. *Due to the fact that video streaming is the current "killer" application and to competitiveness, telecommunication service providers need to be able to answer a fundamental question: to which extent is the available network infrastructure able to successfully provide users with a satisfactory experience when running video streaming applications? Answering this question is far from trivial, because existing techniques are neither scalable nor accurate enough. To address this issue, we propose a model to predict video streaming quality based on the observation of performance indicators of the underlying IP network. To accomplish this objective, the proposed model — created using LTE networks as case study — leverages low network consumption active measurements and machine learning techniques. Obtained results show that the proposed solution produces accurate estimates (average error of less than 10%) while keeps intrusiveness around twenty times lower than traditional techniques.*

Resumo. *Pelo fato de streaming de vídeo representar aplicação "matadora" e por competitividade, provedores de serviços de telecomunicações precisam ser capazes de responder uma pergunta fundamental: em que medida a infraestrutura de rede disponível tem dado conta de oferecer aos usuários uma percepção de qualidade satisfatória quando empregam essas aplicações? Obter resposta a essa pergunta está longe de ser trivial, pois as técnicas existentes ou não são escaláveis ou são pouco acuradas. Para suprir essa lacuna, propõe-se um modelo de predição de qualidade de streaming de vídeo com base na observação de desempenho da rede IP subjacente. Para tal, o modelo proposto — criado usando redes LTE como estudo de caso — lança mão de técnica de medições ativas de baixo consumo de banda e de aprendizagem de máquina. Os resultados alcançados evidenciam que a solução proposta produz estimativas acuradas (erro médio abaixo de 10%) e apresenta intrusividade cerca de vinte vezes menor que técnicas tradicionais.*

1. Introdução

A taxa de penetração da banda larga móvel estimada pelo ITU para 2014 foi de 83,7% em países desenvolvidos, enquanto que os acessos de banda larga fixa, no mesmo período, chegaram a 27,5% da população [ITU 2015]. Em paralelo com o aumento do número de acessos, a proliferação das aplicações de vídeo aumenta a complexidade da tarefa de gerenciar a qualidade dos serviços providos, pois a aplicação de vídeo, quando comparada a outras aplicações, exige uma grande quantidade de recursos da rede. Estima-se que até 2016 o tráfego de vídeo na Internet represente cerca de 86% do volume total cursado na rede [Katsarakis et al. 2014].

Nesse cenário, tanto a comunidade científica quanto a indústria concordam que maximizar a qualidade de experiência do usuário (QoE), quando do uso de aplicações de *streaming* de vídeo, representa um desafio de pesquisa relevante [Katsarakis et al. 2014] [Quadros et al. 2015]. Um aspecto chave nessa direção consiste em aferir, de forma sistemática, a qualidade dos serviços de vídeo providos. O ambiente de operadora exige uma solução de baixa intrusividade, que permita medições em larga escala, e que mensure a qualidade do serviço de vídeo entregue ao usuário com um grau aceitável de acurácia. Essa tarefa torna-se mais complexa ao se considerar redes celulares, foco deste artigo, em que medições com maior intrusividade têm potencial para interferir diretamente na qualidade dos serviços providos.

Um caminho possível para resolver o problema foi estudado por Msakni e Youssef [Msakni and Youssef 2013]. Eles realizaram uma análise sobre técnicas propostas recentemente para predição de QoE que dispensam a necessidade de transferir vídeos para conduzir medições. O estudo concluiu que, até o momento, não existe uma técnica confiável para realizar tal estimativa. Em essência, a não linearidade da opinião humana acaba por comprometer a acurácia do uso de indicadores de camada de rede como preditores *diretos* de QoE, pois um mesmo comportamento de rede poderia ser mapeado para notas distintas de MOS (*Mean Opinion Score*). Um outro caminho possível consiste em avaliar a qualidade de vídeos de forma objetiva junto ao usuário. No entanto, as técnicas que permitem tal medição implicam na necessidade de transferir e analisar arquivos reais de vídeo, incorrendo em acréscimo substancial de tráfego na rede.

Neste artigo propõe-se um modelo para predição de indicadores de desempenho para o serviço de *streaming* de vídeo por meio da observação de indicadores de desempenho da rede IP subjacente. O modelo proposto explora a técnica de árvores de decisão para estabelecer e quantificar a relação entre indicadores de QoS¹ e indicadores objetivos que infiram a qualidade da reprodução de vídeos no contexto do usuário final, doravante denominados AppQoS². Adicionalmente, por meio de um processamento de AppQoS, o modelo proposto permite inferir a qualidade da experiência do usuário. Nesta primeira iteração para resolver o referido problema, focou-se — como estudo de caso — em redes LTE. Resultados obtidos a partir de avaliação em um ambiente controlado permitem observar que o modelo proposto é adequado para viabilizar predição em larga escala da qualidade dos serviços de vídeo entregue aos usuários, com baixa intrusividade.

O restante do artigo está organizado da seguinte forma: a seção 2 aborda os trabalhos relacionados. A seção 3 apresenta o modelo de predição proposto. Na seção 4 aborda-se a configuração do ambiente experimental e os aspectos de construção do modelo. Na seção 5 é relatada a avaliação de desempenho, bem como a aplicação do modelo, enquanto a seção 6 conclui o artigo e apresenta perspectivas de trabalhos futuros.

2. Trabalhos Relacionados

Mecanismos clássicos como MOS, PEVQ (*Perceptual Evaluation of Video Quality*) e PSNR (*Peak Signal-to-Noise Ratio*) são reconhecidamente não escaláveis quando utilizados para determinar — de forma sistemática — se uma infraestrutura de rede está (ou não) prestando o suporte necessário a aplicações de vídeo. Tal se deve por requererem entrevistas com usuários e/ou demandarem transferências de vídeos. Na tentativa de contornar tais limitações, mais recentemente foram conduzidos trabalhos de pesquisa com o objetivo de permitir inferência de qualidade, mas calcados em procedimentos de aquisição de

¹O termo indicadores de QoS é empregado para referenciar indicadores de desempenho de redes IP.

²O termo AppQoS é empregado por representar a última barreira objetiva possível de ser medida no contexto do usuário final.

dados que não tornem o seu uso proibitivo para ambientes de larga escala. Nesse contexto, dois grupos de trabalhos relacionados — doravante denominados grupo A e grupo B — se destacam. De forma geral, trabalhos que se encaixam no grupo A caracterizam-se por, partindo de indicadores de QoS, estimarem diretamente QoE. Por não levarem em conta indicadores de desempenho objetivos da aplicação, não permitem capturar fidedignamente se a rede está oferecendo o suporte adequado à aplicação e acabam sujeitos a estimativas influenciadas pela não linearidade da opinião humana. Por sua vez, trabalhos pertencentes ao grupo B preocupam-se em estabelecer relação entre indicadores de camada de aplicação e QoE, mas sem vinculá-los com indicadores de QoS. Tal não ajuda o provedor de serviço a entender a influência da rede nos indicadores de QoE.

Como um exemplo representativo de trabalho de pesquisa do grupo A, De Pessemer *et al.* [De Pessemer et al. 2013] avaliaram a influência da variação de parâmetros de QoS da rede na qualidade de experiência de usuários para aplicações de vídeo que utilizam DASH (*Dynamic Adaptive Streaming over HTTP*). Os autores realizaram experimentos em que parâmetros como atraso e vazão da rede foram degradados de forma controlada. Ao mesmo tempo, mediram por meio de MOS a satisfação dos usuários quanto à qualidade da reprodução. Embora considere e manipule parâmetros de QoS, o trabalho está focado no entendimento da tolerância do usuário em relação à qualidade de vídeos, ou seja, QoE, e não na caracterização da influência de parâmetros de QoS na qualidade objetiva da reprodução de vídeos.

Ainda como trabalho do grupo A, Hsu e Lo [Hsu and Lo 2014] propuseram um modelo de mapeamento entre QoS e QoE para o contexto de uma rede *multicast* em nuvem. Nesse caso, os autores realizaram uma série de experimentos para estabelecer a relação entre os parâmetros de QoS configurados na rede *multicast* e o seu respectivo impacto na nota MOS atribuída pelos usuários ao assistirem vídeos. Embora resolva o problema de escalabilidade via uso de um preditor, a proposta converge em um indicador impreciso para provedores de serviço, pois uma mesma condição de rede, em razão da análise subjetiva, pode ser mapeada em diferentes notas de MOS.

O grupo B reúne pesquisas orientadas ao estudo dos fenômenos que o uso do protocolo HTTP introduz na reprodução de vídeos, bem como o seu reflexo na experiência do usuário. Para mensurar os principais fenômenos foram estabelecidos indicadores, dos quais pode-se destacar: atraso de inicialização na reprodução de um vídeo, número e duração de interrupções de reprodução causadas por esvaziamento de *buffer* e somatório de tempos de parada [Balachandran et al. 2012].

Na mesma direção, o trabalho desenvolvido por Balachandran *et al.* [Balachandran et al. 2013] introduz uma técnica para predição de QoE por meio da análise de indicadores de desempenho da camada de aplicação. Para tanto, os autores consideraram uma base de indicadores de camada de aplicação já existente sobre o comportamento do *player* de usuários reais. Para obtenção desses indicadores, esse tipo de técnica necessita que o usuário tenha um *plugin* instalado no seu dispositivo de acesso e que os vídeos sejam efetivamente reproduzidos. Embora os indicadores produzidos sejam valiosos para que o provedor possa estimar QoE, essa informação é incompleta na medida em que não permite ao provedor compreender a influência do desempenho de sua rede na qualidade de experiência do usuário.

Nesse cenário, o presente trabalho avança no estado da arte na medida em que, diferente dos trabalhos dos grupos A e B, propõe o uso de indicadores de QoS para estimar indicadores intermediários entre QoS e QoE, os quais rotulou-se AppQoS. Além disso, de posse de indicadores de AppQoS é possível inferir a qualidade da experiência do usuário

de forma indireta. Dessa forma, o indicador de QoE permite destacar de forma preliminar os pontos na rede que estão sofrendo degradação de qualidade percebida, enquanto que os indicadores de AppQoS fornecem insumos que permitem ao provedor compreender as causas da queda de qualidade, bem como fornecem subsídios para avaliar a efetividade de eventuais intervenções na rede. Por fim, advoga-se que o modelo proposto neste trabalho é escalável, pois permite uma estimativa confiável do desempenho da aplicação com uma fração da intrusão de rede necessária para realizar medição utilizando tráfego real de aplicação.

3. LEAP: Um Modelo para Predição Integrada de AppQoS e QoE

Para permitir as predições recém referidas na seção anterior, propõe-se o modelo LEAP (*Lightweight AppQoS and QoE Predictor*). O modelo objetiva fornecer ao provedor uma visão analítica sobre como o desempenho de sua rede afeta o desempenho da aplicação de *streaming* de vídeo e a respectiva experiência do usuário. A Figura 1 ilustra o esquema geral do modelo proposto. Sua construção depende de indicadores de rede e de aplicação. Como indicadores de rede, neste trabalho considera-se: (i) atraso, (ii) variação do atraso, (iii) vazão e (iv) perda. Já para aferir a qualidade da reprodução de vídeos, o presente trabalho considera: (i) tempo de inicialização, (ii) número de paradas e (iii) somatório do tempo das paradas. Para cada indicador de AppQoS são utilizados os quatro indicadores de QoS, construindo-se cada um dos três preditores de forma individual. Em um segundo momento, os três indicadores de AppQoS são utilizados para estimar QoE (MOS). Além do uso para predição de AppQoS e QoE para o serviço de *streaming*, defende-se que esse arcabouço de predição é flexível e suficientemente geral para ser utilizado em outras aplicações.

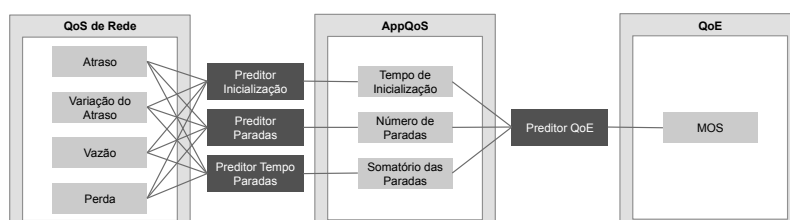


Figura 1. Esquema geral do modelo proposto

3.1. Predição de AppQoS

Neste trabalho emprega-se o modelo de predição baseado em Árvore de Decisão de Regressão em razão da capacidade desse modelo em lidar com atributos que possuem uma relação complexa e não linear entre si. Durante a fase de estudo de candidatos também foram considerados os seguintes modelos de aprendizado de máquina: Regressão Linear Múltipla, Árvores de Decisão, Redes Neurais Artificiais, Gaussian Naive Bayes e Suport Vector Regression. Esses, contudo, incorreram em desempenho inferior, em termos de acurácia, quando da execução de processo de aprendizagem via protocolo NCV (*Nested Cross Validation*) [Tsamardinos et al. 2014].

Durante a fase de treinamento, o modelo aprende a influência de cada atributo X na variável de resposta Y . Nesse caso, os indicadores de QoS são os atributos (X) e os indicadores de desempenho da aplicação de vídeo são as variáveis de resposta (Y). Após construídas as árvores, para qualquer nova instância de valores de QoS, é possível realizar a predição de cada indicador de AppQoS apenas comparando, em cada árvore, os valores X com os limiares estabelecidos, nó após nó, até a chegada em um nó final, onde estará a

predição do modelo. Por exemplo, na Figura 2 é possível observar segmentos das árvores de decisão que realizam a predição para os indicadores AppQoS de número de paradas e duração de paradas para a resolução 1080p. Os aspectos construtivos e a avaliação da acurácia dessas árvores serão explicadas na Subseção 4.3.

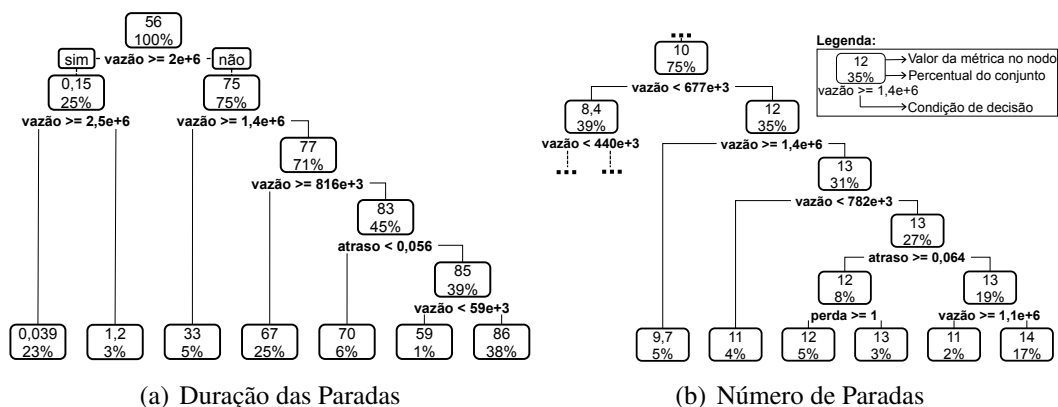


Figura 2. Árvores de decisão para duração e número de paradas para 1080p

Para construir cada árvore de decisão é empregado um procedimento recursivo que vai construindo uma árvore binária por meio de sucessivas divisões, partindo de um nó único que contém todos os atributos X . Esse procedimento ocorre balizado por dois parâmetros que controlam o crescimento da árvore: um limiar *minsplits*, que representa o número mínimo de observações associadas para que um novo ramo seja criado, e um segundo parâmetro *cp*, que indica o ganho mínimo na redução do erro que um novo ramo precisa proporcionar para ser criado. Como saída do processo de geração, cada árvore possui associada a si um erro mínimo de validação cruzada *xerror*, que consiste em uma estimativa do erro que será verificado ao se submeter dados independentes (ou seja, não utilizados no treinamento) à estrutura criada. Em um segundo momento, realiza-se o processo de poda. Nesse procedimento, para cada par de nós finais, com um pai comum, avalia-se o erro médio quadrático (MSE) quando da remoção dos nós finais, conforme a Equação 1, fazendo do nó pai um nó final. Se o erro diminui, a árvore é podada, senão é mantida igual para esse nó.

$$MSE = \sum_{c \in \text{leaves}(T)} \sum_{i \in c} (y_i - m_c)^2 \quad (1) \quad Qini = -0,963 \times \log_{10}(t_0 + 5,381) + 5 \quad (2)$$

Como um exemplo da aplicação desses parâmetros construtivos, a Figura 3 ilustra três aspectos importantes para a construção do preditor de tempo de inicialização para vídeos 1080p. Em (a) e (b) é possível verificar que tanto o erro de validação cruzada quanto o R^2 estabilizam depois de 5 *splits*. Pode-se afirmar, então, que deixar a árvore crescer para mais de 5 bifurcações não ajudaria a diminuir o erro ou melhorar o ajuste do preditor. Já em (c) é possível observar o mapa de calor que ilustra a matriz gerada pela interpolação dos parâmetros *minsplits* e *cp*, onde as zonas mais escuras representam valores menores de *xerror*. Objetivamente, para o preditor em questão chegou-se a um *minsplits*=24, *cp*= 10^{-22} e número de *splits*=5, que representam a parametrização ótima do modelo.

Antes de encerrar a apresentação do modelo, é importante destacar que seu modo de operação pode ser dividido em duas etapas. A primeira etapa, que ocorre de maneira

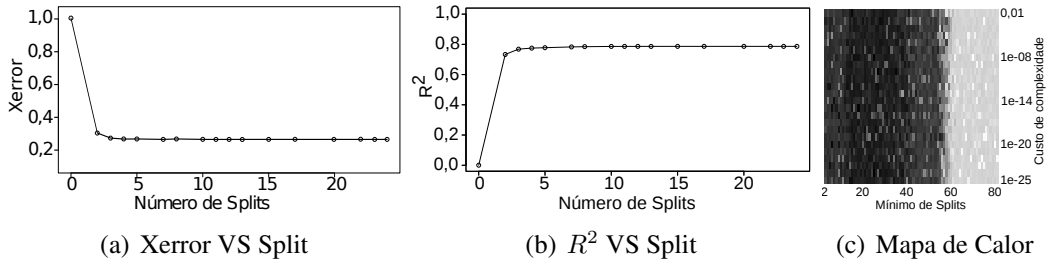


Figura 3. Aspectos construtivos do preditor de tempo de inicialização para 1080p

offline, consiste na seleção e no treinamento do modelo. Nessa etapa são executadas tarefas de alta complexidade. Mais especificamente, a geração das Árvores de Decisão binárias ótimas constitui um problema NP-Completo, exigindo o emprego de algoritmos heurísticos para obtenção de uma árvore *near-optimal*. Já a aplicação do modelo de predição é feita de forma *online*, ou seja, na medida em que os dados de QoS estão disponíveis, os indicadores de AppQoS e QoE são calculados e disponibilizados.

3.2. Predição de QoE

Complementando o trabalho do presente artigo, o LEAP permite utilizar os indicadores de desempenho de aplicação para estimar a qualidade da experiência do usuário. Para estabelecer a relação entre os indicadores de AppQoS e a nota MOS atribuída por um usuário o modelo lança mão de uma adaptação da formulação matemática oriunda de recentes trabalhos de pesquisa que, por meio de testes com usuários de serviços de *streaming* de vídeo, relacionaram a influência do tempo de inicialização de vídeos com a degradação da qualidade de experiência do usuário [Seufert et al. 2014] [Höbfeld et al. 2012]. A influência do tempo de inicialização (t_0) na nota MOS pode ser definida pela Equação 2, onde $Qini$ é a nota MOS influenciada e t_0 é o valor predito para o tempo de inicialização. Os valores 0,963 e 5,381 foram calculados através da resolução de um problema de minimização não-linear do erro médio quadrático entre os valores de MOS em t_0 e a função $f(t_0)$.

$$\lambda = \begin{cases} \frac{\sigma}{\sigma + \rho}, & \text{se } \sigma + \rho < T \\ \frac{\sigma}{T}, & \text{caso contrário} \end{cases} \quad (3) \quad Qst = \begin{cases} 1, & \text{se } n > 6 \\ a_i \times e^{-b_i \times n} + c_i, & \text{se } n \leq 6 \end{cases} \quad (4)$$

Já para modelar a influência das interrupções na percepção da qualidade de exibição de um vídeo, primeiramente é necessário definir, conforme a Equação 3, um fator λ que calcula a razão entre o tempo total σ em que o vídeo ficou parado e o intervalo transcorrido a partir do início da observação, dado pela soma de σ com o tempo efetivo ρ de exibição do vídeo. De acordo com [Casas et al. 2013], essa observação é realizada dentro de *slots* T com duração típica de um minuto, o que permite generalizar o método para vídeos de qualquer duração. Uma vez que se tenha calculado λ , observa-se seu valor para definir um nível $1 \leq i \leq 5$ de degradação, dado pelos intervalos de λ descritos nas colunas da Tabela 1. O MOS Qst é, então, calculado de acordo com a Equação 4, onde a_i, b_i, c_i são constantes definidas $\forall i = 1, 2, 3, 4, 5$ de acordo com a Tabela 1, e n é o número de paradas dentro do *slot* T . Pode-se observar na referida equação que para um número de paradas maior que seis, a qualidade estará totalmente comprometida, o que resulta em um MOS com nota 1.

Tabela 1. Tabela de fatores em razão de λ para cálculo de MOS Qst

Fator	$\lambda < 0,05$	$0,05 \leq \lambda < 0,10$	$0,10 \leq \lambda < 0,20$	$0,20 \leq \lambda < 0,50$	$\lambda \geq 0,50$
a	3,012682983	3,098391523	3,190341904	3,248113258	3,302343627
b	0,765328992	0,994413063	1,520322299	1,693893480	1,888050118
c	1,991000000	1,901000000	1,810138616	1,751982415	1,697472392

4. Construção do Modelo

Esta seção apresenta os aspectos práticos relacionados com a construção do modelo de predição. Primeiramente, aborda-se a aquisição de indicadores de QoS e AppQoS. Logo após, descreve-se detalhes do ambiente de treinamento do modelo e, por fim, apresenta-se e discute-se os modelos de predição resultantes.

4.1. Aquisição de Indicadores de QoS e AppQoS

Para obtenção de indicadores de desempenho de rede, empregou-se a plataforma NetMetric [dos Santos et al. 2007]. A plataforma NetMetric teve sua primeira implementação funcional em 2007 e, desde então, está em constante desenvolvimento para adaptar-se aos padrões de medição em redes IP, principalmente os definidos pelo IETF IPPM. Para tanto, emprega técnicas de medição ativa, em que rajadas controladas de pacotes são injetadas em um ponto de origem em direção a um ponto de destino, permitindo avaliar a deformação dos pacotes ao longo da rede e extrair indicadores de desempenho da rede IP.

O NetMetric possui uma entidade Gerente, que permite configurar agendas de testes a serem executados por entidades Agente. Por sua vez, uma entidade Agente pode ser utilizada tanto como origem de uma medição (fonte) quanto como destino (refletora). A plataforma foi configurada para trabalhar com conjuntos de duas rajadas de pacotes distintas. Uma primeira rajada utiliza o protocolo UDP e serve para medir atraso de uma via (OWD - *One Way Delay*), variação do atraso (*jitter*), bem como perda de pacotes, também em um sentido. Cada instância dessa rajada injeta um conjunto de 400 pacotes de 100 bytes separados por um *gap* de 50 ms. Já para medir vazão, é utilizada uma rajada com o protocolo TCP contendo 640 pacotes de 1.488 bytes para cada sondagem entre dois pontos. As duas rajadas somadas geram um volume unidirecional de 992 KB. Embora a plataforma NetMetric seja capaz de realizar medições de forma bi-direcional, optou-se por extrair indicadores de forma unidirecional a fim de isolar o canal foco desse estudo, ou seja, o *downlink* de redes LTE.

Para obtenção dos indicadores de AppQoS foi desenvolvido um módulo para a plataforma NetMetric, materializado na forma de um *plugin* para o navegador Chrome, que viabiliza a extração de indicadores de desempenho relacionados à reprodução de vídeos diretamente do controlador de vídeo nativo do HTML5. A justificativa para tal escolha reside na migração de grandes provedores de *streaming* de vídeo para a tecnologia HTML5, tal como o Youtube e o Netflix.

Para viabilizar a medição da camada de aplicação, o NetMetric foi configurado para que o agente fonte realize reproduções periódicas de vídeos via navegador Chrome, que recupera vídeos armazenados no agente refletor. Para realização do treinamento do algoritmo foram utilizados dois vídeos de sessenta segundos cada, todos no formato MPEG v4 e codificados em H.264, com resoluções de 720p e 1080p, com volumes de 9,2 MB e 14,3 MB respectivamente. Cabe destacar que a transferência de vídeos em diferentes resoluções torna-se necessária, pois cada resolução possui um *bitrate* diferente e, dessa forma, exige um nível diferenciado de desempenho da camada de rede. Nesse sentido, cada um dos três preditores de parâmetros de AppQoS necessita ser treinado de forma

isolada para cada resolução de vídeo, permitindo que o algoritmo de treinamento aprenda a demanda de rede para reprodução de vídeos em cada resolução.

4.2. Ambiente de Treinamento do Modelo

Uma abordagem possível para obtenção do conjunto de dados que relaciona QoS e AppQoS seria a realização das medições em uma rede LTE em produção, realizando as medições da camada de rede e de aplicação em paralelo, objetivando garantir que as condições de rede não seriam alteradas durante as medições nas duas camadas. No entanto, em razão da natureza invasiva das duas estratégias de medição, a medição concomitante nas duas camadas implicaria em uma competição por recursos compartilhados de rede, resultando em uma interferência mútua entre as técnicas. Uma outra abordagem possível poderia prever a serialização das medições no tempo. No entanto, devido à grande variabilidade de desempenho em redes LTE, observada mesmo em pequenos intervalos de tempo, concluiu-se que esse não seria um ambiente adequado para treinar o modelo. O treinamento exige, obrigatoriamente, que a condição de rede permaneça estática entre a aferição de desempenho da rede e da aplicação. Caso contrário, o algoritmo de aprendizado estabeleceria relações erradas sobre a influência de cada parâmetro de rede nas variáveis de resposta.

Diante dessas reconhecidas dificuldades, optou-se por criar um ambiente controlado que emula de forma fiel as condições de rede observadas em um ambiente de produção. Para emular as condições de vazão, atraso, variação de atraso e perda de pacotes inerentes aos ambientes LTE, foi empregada a solução WANEM [Kalitay and Nambiar 2011]. A ferramenta WANEM permite impor, de forma controlada, condições variadas de rede ao tráfego de interesse. A Figura 4 ilustra a topologia utilizada para implantar o ambiente controlado.

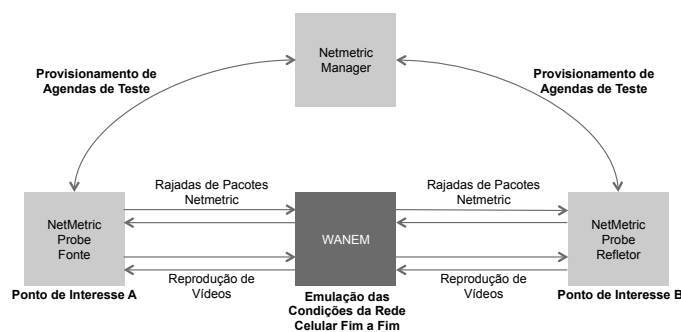


Figura 4. Ambiente controlado de treinamento

A escolha dos valores utilizados para a configuração do WANEM foi feita com base em 7.450 medições realizadas pelo NetMetric em uma rede LTE de abrangência nacional em produção³, entre maio e outubro de 2015. A Tabela 2 enumera os quatro níveis selecionados para cada indicador de QoS. Para convergir para os referidos valores, cada indicador foi testado individualmente quanto à normalidade de suas distribuições com um teste de Shapiro-Wilk, a uma significância $\alpha = 0,05$. Em razão da distribuição normal, os indicadores de atraso, variação do atraso e vazão foram segmentados por meio da análise de quartis. Já os valores para o indicador de perda foram selecionados via análise de moda para valores inteiros de perda.

³Por questões de confidencialidade, não é possível apresentar caracterização detalhada da rede.

Por meio da realização de experimentos preliminares (via *design Full Factorial* $2^{k.r}$), constatou-se que o indicador de variação de atraso não ajudava a explicar o modelo de regressão proposto na seção anterior. Uma vez determinados os parâmetros relevantes (vazão, atraso e perda), foi realizado um segundo conjunto de experimentos, dessa vez com o *design Full Factorial* $4^{k.r}$, permitindo uma maior variação de níveis para cada parâmetro sem incorrer em um número exagerado de experimentos. Com três parâmetros de entrada e quatro níveis, o *design* resultou em 64 combinações possíveis. Por meio da observação da variabilidade dos resultados foi definido o número de dez replicações para cada combinação, considerando-se um nível de significância de 95%.

Tabela 2. Valores da rede LTE real usados no treinamento do modelo

Indicador	Valor
Vazão	0,9 Mbps; 8,8 Mbps; 15,5 Mbps; 25,3 Mbps
Atraso	22 ms; 56 ms; 64 ms; 98 ms
Perda	0%; 2%; 5%; 13%

4.3. Preditores Resultantes

Nesta etapa, com os dados medidos no ambiente controlado, trabalhou-se na geração de árvores de decisão binária para cada um dos três indicadores de AppQoS preditos. Essas árvores devem ser específicas para cada uma das duas resoluções testadas (720p e 1080p), totalizando a derivação de seis dessas estruturas.

As árvores das Figuras 2 e 5, assim como as demais geradas no escopo deste trabalho, refletem as podas ótimas das estruturas. Na Figura 2(a) está representada a árvore para o tempo total de parada de um vídeo, na resolução 1080p, em função dos três indicadores primários de rede. O nodo raiz, que simboliza o primeiro critério de decisão da árvore, é dado em função da vazão de dados observada na rede (representada em *bits* por segundo). O valor indicado neste nodo, 2 Mbps, está relacionado ao *bitrate* do vídeo utilizado nos testes - em torno de 1,9 Mbps. Para valores de vazão acima dessa taxa, a duração média do total de paradas (0,15 segundos, de acordo com o valor central indicado pelo nodo filho à esquerda da raiz) é consideravelmente menor do que a observada para redes com uma taxa menor (75 segundos, conforme registrado pelo nodo imediatamente à direita da raiz). Isto indica que o fato de a rede ter capacidade para dar suporte ao *bitrate* do vídeo é o fator mais importante para determinar o comportamento macro de sua exibição.

Outro aspecto relacionado é a existência de uma faixa intermediária acima do *bitrate* em que há uma pequena degradação nos indicadores de exibição do vídeo - no caso da Figura 2(a), entre 2 Mbps e 2,5 Mbps. Nessa faixa, qualquer variação instantânea na vazão pode levar a uma necessidade de preenchimento de *buffer*, levando à observação dessa degradação. Para valores acima de 2,5 Mbps, no entanto, a reprodução do vídeo se dá praticamente sem interrupções, independente do valor de qualquer outro indicador preditor. Destaca-se que um comportamento semelhante foi observado para 720p, em que o nodo raiz foi determinado por uma vazão idêntica ao *bitrate* do vídeo (1,2 Mbps), mas a reprodução sem falhas foi mapeada apenas para vazões superiores a 1,9 Mbps.

Para o indicador de quantidade total de paradas na resolução 1080p, o nodo raiz, assim como a porção da árvore à sua esquerda, são análogos aos seus correspondentes para o indicador de duração total de paradas. Isto significa que a primeira decisão é tomada em relação a uma vazão semelhante ao *bitrate*, e que valores superiores a esse parâmetro ainda apresentam degradações até um segundo limiar de vazão (2,3 Mbps). No entanto, o ramo destacado pela Figura 2(b), que representa a porção dessa árvore conectada imediatamente à direita do nodo raiz, tem um comportamento distinto dos observados até então.

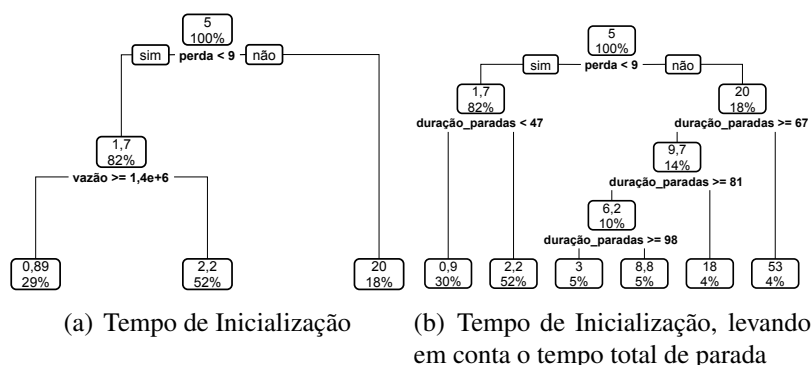


Figura 5. Árvores de decisão para tempo de inicialização na resolução 1080p

Para esse subconjunto, valores entre 677 kbps e 2 Mbps (limitação dada pelo nodo raiz) possuem em média 12 paradas, uma quantidade maior se comparada àquela observada para vazões menores (8,4 paradas em média). Uma análise conjunta com o tempo total de paradas no conjunto de treinamento mostra que, para as amostras em que a velocidade fica abaixo de 677 kbps, a média do tempo total em que o vídeo fica interrompido é de 83,76 segundos. No caso complementar, essa mesma grandeza é avaliada em 64,98 segundos. Pode-se concluir, portanto, que o vídeo efetivamente fica mais tempo parado em redes mais lentas, mas que o intervalo demandado por cada interrupção é menor.

Para o tempo de inicialização de vídeo, de acordo com a Figura 5(a), a decisão no nodo raiz é tomada por uma condição muito degradada de rede (perda de pacotes acima de 9%). De acordo com a Tabela 3, os fatores de ajuste deste preditor em relação aos valores efetivamente medidos foram baixos, com $R^2 = 0,3147$ no melhor caso, para a resolução 1080p. Isto indica a insuficiência dos três indicadores de QoS de rede para a derivação do indicador de tempo de inicialização. Para tornar a predição mais precisa, buscou-se, entre todos os indicadores envolvidos, aqueles em que a correlação de Pearson com o indicador em questão fosse mais forte. Verificou-se que, além dos indicadores já utilizados na árvore (perda e vazão), o próximo indicador com maior correlação era o de duração total de paradas, cujo vínculo com o tempo de inicialização mostrou-se cinco vezes mais forte do que o existente entre esse indicador e o atraso unidirecional, por exemplo. A Figura 5(b) ilustra a árvore de decisão gerada com a utilização do tempo total de paradas como variável de entrada. A predição resultante da submissão das amostras de teste a essa estrutura levou a um crescimento do fator de ajuste, que subiu para $R^2 = 0,8085$ para o vídeo 1080p, com redução também na raiz do erro quadrático, conforme pode ser observado na coluna Tempo de Inicialização 2 da Tabela 3.

Tabela 3. RMSE e R^2 dos preditores para 720p e 1080p

	Duração Paradas		Total Paradas		Tempo Inicialização		Tempo Inicialização 2	
	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
720p	5,7548 s	0,9627	1,4747	0,8222	4,2321 s	0,2169	2,0151 s	0,8774
1080p	9,8849 s	0,9274	1,1847	0,9449	6,5461 s	0,3147	3,9102 s	0,8085

5. Avaliação de Desempenho

Esta seção apresenta a avaliação de desempenho do modelo proposto. Adicionalmente, a seção aborda os resultados potenciais da aplicação do modelo na rede da operadora móvel mencionada ao final da Subseção 4.2. Essencialmente, busca-se responder as seguintes perguntas de pesquisa: (i) Que nível de acurácia é possível obter-se ao predizer o desempenho da aplicação de vídeo com base na observação do desempenho de rede? (ii) Qual

o ganho, em termos de redução de intrusividade, que o modelo proposto fornece quando de sua aplicação em uma rede LTE de grande porte? (iii) De que maneira um provedor poderia tirar proveito de uma predição integrada de qualidade de vídeo e QoE?

5.1. Acurácia do Modelo

A acurácia do modelo proposto pode ser avaliada com a submissão do conjunto de teste, o qual é independente dos dados de treinamento, para predição das variáveis representadas. Cada amostra i deste conjunto contém um valor medido para as três variáveis preditoras (vazão, atraso e perda) e para os três indicadores de aplicação (duração de paradas, contagem de paradas e tempo de inicialização de vídeo). Utilizando-se o primeiro grupo de variáveis como entrada das árvores representadas nas Figuras 2 e 5, obtém-se uma saída estimada para cada um dos indicadores do segundo grupo, que então passam a ter um valor observado x_i e um valor predito \hat{x}_i . Isto permite calcular, para cada amostra i , o resíduo normalizado r_i , definido pela Equação $r_i = |\hat{x}_i - x_i|/N$. O fator N é utilizado para normalização dos erros. Para os três indicadores de aplicação, o valor de N é derivado da duração dos vídeos utilizados nos experimentos realizados no ambiente controlado (60 segundos), o que permite generalizar a avaliação do método para um vídeo de extensão arbitrária.

Os gráficos da Figura 6 variam r_i no eixo horizontal, associando cada valor desse eixo a uma porção do conjunto amostral (no eixo vertical) em que o próprio valor de r_i é igual ou inferior ao limiar fixado. Assim, considerando a resolução 1080p na Figura 6(c), o valor 0,093 no eixo horizontal está associado ao valor 0,9 no eixo vertical, indicando que 90% das amostras do respectivo conjunto de teste apresentam $r_i \leq 0,093$. Já para a resolução 720p, no mesmo gráfico, 90% das amostras possuem $r_i \leq 0,0086$. Isso pode ser interpretado como um erro de 9,3% e 0,86%, respectivamente. Calculando-se a média dos valores de r_i associados a 90% das amostras para todos os indicadores, nas duas resoluções, tem-se um valor médio de $\bar{r}_i = 0,0982$ (9,8%).

Também com base nos valores observados e preditos dos indicadores de aplicação, é possível calcular o MOS de acordo com a abordagem apresentada na Subseção 3.2. Para esse indicador, que é independente da duração do vídeo, foi usado um valor $N = 1$, ou seja, os erros apresentados são dados em função da escala absoluta de MOS, que varia de 1 a 5. De acordo com a Figura 6(d), a predição de MOS apresentou um erro de até $r_i = 0,11$ para 90% dos casos do conjunto de teste 1080p. Analisando o significado desse indicador com um exemplo, tem-se que, se um $MOS = 3,5$ for calculado utilizando os indicadores de aplicação observados, um valor $3,39 \leq \widehat{MOS} \leq 3,61$ será computado com a predição desse indicadores em 90% das situações. Para a resolução 720p, o erro foi ainda menor, apresentando o valor $r_i = 0,05$. Embora o erro dos preditores de AppQoS já seja bem baixo, o impacto desse erro é ainda menor quando se usa os indicadores de AppQoS para predição de QoE, conforme ilustrado na Figura 6(d).

5.2. Intrusividade do Modelo

A Figura 7(a) ilustra um comparativo de intrusividade, montado com parâmetros de volume descritos na Subseção 4.1, entre o modelo proposto e a medição de qualidade por meio da transferência de vídeos reais. Como é possível observar, o caso mais intrusivo de LEAP (intervalo de *polling* de 10 minutos) para 500 sondas, o volume de dados gerados atinge 66,65 GB/dia. Para a mesma configuração, a estratégia da transferência de vídeos reais em série para medição da qualidade da rede geraria um volume de 1.574,71 GB/dia. Ou seja, a intrusividade das técnicas convencionais é mais de vinte vezes maior do que a observada ao se empregar LEAP.

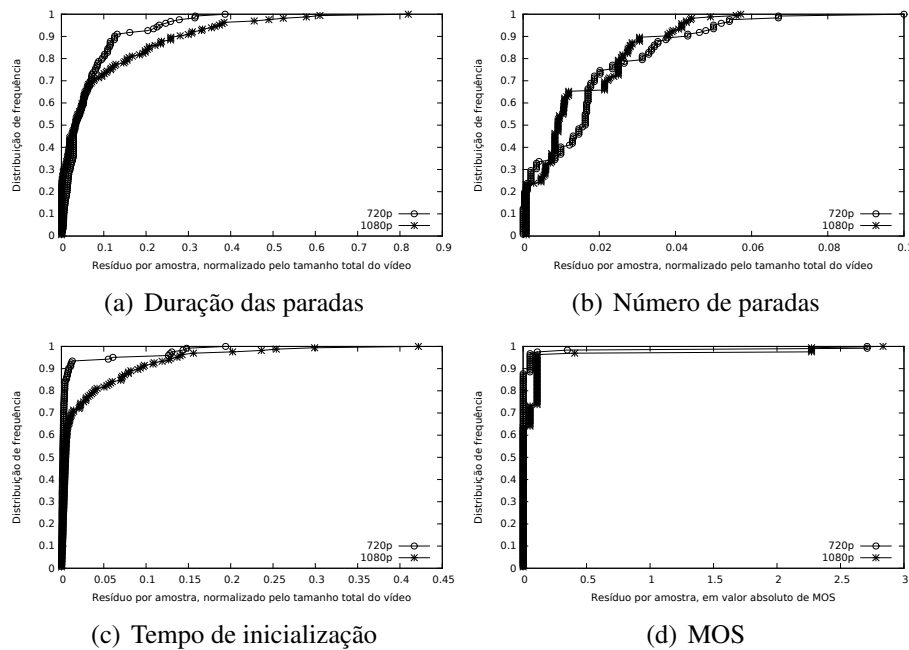


Figura 6. CDFs dos erros para predição de AppQoS e QoE em vídeos de 1080p

Complementando a análise recém apresentada, destaca-se, ainda, que a alta variabilidade de desempenho da rede LTE, justificada principalmente pela mobilidade dos usuários e pela alta competição por recursos da interface de rádio, exige uma frequência elevada de monitoramento de qualidade para que o operador de serviços tenha uma visão realista do ambiente de rede. Nesse cenário, o aspecto de baixa intrusividade é indispensável para que uma solução de monitoramento de desempenho seja escalável. Caso contrário, o tráfego gerado para medição da qualidade da rede consumiria um grande volume de recursos, gerando uma diminuição significativa dos recursos disponíveis para os usuários da rede.

5.3. Aplicação do Modelo

Após construído, o modelo foi colocado em operação em uma operadora celular de grande porte, já referida na Subseção 4.2. Para tanto, foi definido um ponto de interesse para instalação do agente refletor, que foi configurado junto ao servidor CDN da operadora. De forma complementar, foram instalados agentes fonte espalhados por todo o território nacional. Essa configuração permitiu o monitoramento sistemático da qualidade do serviço de vídeo em um contexto fim-a-fim, entre o ponto de interesse refletor e os pontos de interesse regionais.

Por restrição de espaço, apresenta-se agora um único exemplo concreto de uso do modelo. Para apresentar os resultados consolidados foi utilizado um gráfico de coordenadas paralelas que evidencia o relacionamento dos diferentes indicadores em uma mesma sessão de medição. Na Figura 7(b) é possível observar duas linhas em destaque. As duas linhas compartilham uma condição idêntica de rede: 77 ms de atraso unidirecional, 0% de perda de pacotes e 1,950 Mbps de vazão. Para a mesma condição de rede, foi utilizado o preditor para inferir o comportamento da aplicação. Para o vídeo de 720p o número de paradas foi zero, o tempo total de paradas também foi zero e, conseqüentemente, o MOS foi próximo de cinco. Já para o vídeo de 1080p, para as mesmas condições de rede, o número de paradas foi 9,68, o tempo total de paradas foi 32,9 segundos e o tempo de

inicialização de vídeo foi 0,89 segundos. Esse comportamento de aplicação levou a uma nota MOS 1,75.

A predição integrada permite que o provedor de serviços utilize o indicador de QoE para isolar rapidamente qualquer degradação significativa para o serviço de *streaming* de vídeo na rede LTE. Em um segundo momento, os indicadores de AppQoS permitem que o operador da rede identifique qual é o principal ofensor da degradação de QoE (no caso analisado, número e duração de paradas). Isto feito, basta que o operador observe o indicador de QoS que mais influencia o comportamento do indicador de AppQoS de interesse, por meio da observação da Árvore de Decisão relacionada (no caso, como indicadores mais influentes destaca-se vazão seguido por atraso unidirecional). Fechado esse ciclo, o operador pode realizar intervenções na rede (por exemplo, eliminação de gargalos para aumento de vazão e/ou ajustes de filas de prioridade para diminuição de atraso), para logo após observar seus reflexos ao longo das camadas de rede, aplicação e experiência do usuário.

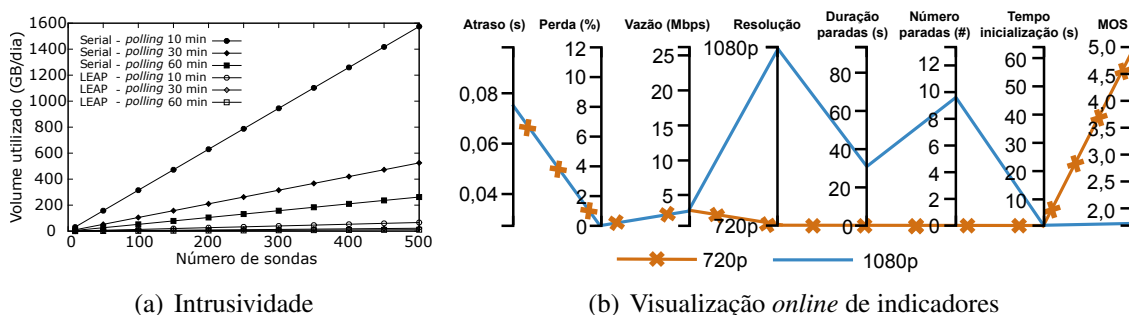


Figura 7. Análise de intrusividade e caso de uso do modelo

6. Conclusão e Trabalhos Futuros

Os experimentos realizados evidenciaram a viabilidade de realizar a estimativa de qualidade da aplicação de *streaming* de vídeo utilizando indicadores de QoS da rede como preditores. A estimativa apresentou um erro médio para AppQoS menor que 9,92%, e um erro para a nota MOS menor que 0,11 pontos para mais de 90% dos casos. Além de acurado, o modelo proposto gera um volume de dados que representa, em média, apenas 4% da intrusividade de técnicas convencionais que necessitam transferir vídeos reais para medição de desempenho. Essa baixa intrusividade permite ao provedor realizar uma medição sistemática e com um pequeno intervalo entre medições, sem que isso consuma um volume excessivo de recursos da rede. É importante mencionar que para garantir que os preditores não percam acurácia ao longo do tempo, principalmente em razão da evolução tecnológica ou da alteração de configurações da rede LTE, o modelo precisa passar por um processo de recalibração periódica. Por ser executado de forma *offline*, e durar na ordem de poucas horas, pode ser conduzido com a frequência desejada pelo operador.

O modelo proposto mostra-se efetivo ao fornecer a operadores de rede informações para suporte à decisão, principalmente durante a execução de rotinas de configuração e detecção de degradações. Para prover subsídios sobre como cada resolução de vídeo irá desempenhar sob determinada condição de rede, optou-se por trabalhar com vídeos de resolução fixa. Como trabalho futuro, pretende-se evoluir o modelo para prover uma predição complementar considerando mecanismos adaptativos, tal como DASH. Além disso, como trabalho futuro vislumbra-se a aplicação do modelo para outros serviços, tais como aplicações de VoIP e IPTV, objetivando ratificar a generalidade do modelo. Por

fim, pretende-se explorar melhor os modelos de QoE por meio da condução de testes com usuários reais para validar ou aprimorar os recentes modelos que realizam o mapeamento AppQoS para QoE.

Referências

- Balachandran, A., Sekar, V., Akella, A., Seshan, S., Stoica, I., and Zhang, H. (2012). A quest for an internet video quality-of-experience metric. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, pages 97–102.
- Balachandran, A., Sekar, V., Akella, A., Seshan, S., Stoica, I., and Zhang, H. (2013). Developing a predictive model of quality of experience for internet video. In *Proceedings of the ACM SIGCOMM 2013*, pages 339–350.
- Casas, P., Schatz, R., and Hoßfeld, T. (2013). Monitoring youtube qoe: Is your mobile network delivering the right experience to your customers? In *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, pages 1609–1614.
- De Pessemier, T., De Moor, K., Joseph, W., De Marez, L., and Martens, L. (2013). Quantifying the influence of rebuffering interruptions on the user’s quality of experience during mobile video watching. *IEEE Transactions on Broadcasting*, 59(1):47–61.
- dos Santos, G. L., Guimaraes, V. T., Silveira, J. G., Vieira, A. T., de Oliveira Neto, J. A., da Costa, R., and Balbinot, R. (2007). Uama: a unified architecture for active measurements in ip networks; end-to-end objective quality indicators. In *Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 246–253.
- Hoßfeld, T., Egger, S., Schatz, R., Fiedler, M., Masuch, K., and Lorentzen, C. (2012). Initial delay vs. interruptions: between the devil and the deep blue sea. In *Proceedings of the 4th International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 1–6.
- Hsu, W.-H. and Lo, C.-H. (2014). Qos/qoe mapping and adjustment model in the cloud-based multimedia infrastructure. *IEEE Systems Journal*, 8(1):247–255.
- ITU (2015). Measuring the information society. Technical report. <http://www.itu.int/en/ITU-D/Statistics/Pages/publications/mis2015.aspx>.
- Kalitay, H. K. and Nambiar, M. K. (2011). Designing wanem: A wide area network emulator tool. In *Proceedings of the 3rd International Conference on Communication Systems and Networks (COMSNETS)*, pages 1–4.
- Katsarakis, M., Fortetsanakis, G., Charonyktakis, P., Kostopoulos, A., and Papadopouli, M. (2014). On user-centric tools for qoe-based recommendation and real-time analysis of large-scale markets. *IEEE Communications Magazine*, 52(9):37–43.
- Msakni, H. and Youssef, H. (2013). Is qoe estimation based on qos parameters sufficient for video quality assessment? In *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 538–544.
- Quadros, C., Santos, A., Gerla, M., and Cerqueira, E. (2015). A qoe-aware mechanism to improve the dissemination of live videos over vanets. In *Proceedings of the XXXIII Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, pages 31–40.
- Seufert, M., Egger, S., Slanina, M., Zinner, T., Hobfeld, T., and Tran-Gia, P. (2014). A survey on quality of experience of http adaptive streaming. *IEEE Communications Surveys & Tutorials*, 17(1):469–492.
- Tsamardinos, I., Rakhshani, A., and Lagani, V. (2014). Performance-estimation properties of cross-validation-based protocols with simultaneous hyper-parameter optimization. In *Artificial Intelligence: Methods and Applications*, pages 1–14.