

# Reprogramação de Fluxos de Dados em Redes SDN Baseada na Análise Contextual do Tráfego

Francisco J. Badaró V. Neto<sup>1,2</sup>, Constantino J. Miguel<sup>1</sup>, Paulo N.M. Sampaio<sup>1</sup>

<sup>1</sup>Universidade Salvador (UNIFACS) - Salvador - BA - Brasil

<sup>2</sup>Centro Universitário UniRuy (UNIRUY) - Salvador - BA - Brasil

{fjbneto, constantino.jacob, pnms.funchal}@gmail.com

**Abstract.** *Software-Defined Networking (SDN) is a network paradigm that allows applying new traffic engineering techniques by separating the control plane from the routing plan. To support the dynamic characteristics of the computational environment (such as dynamic traffic conditions and variety of network protocols, user mobility, device heterogeneity, among others) and others innovative traffic engineering techniques should propose handling time-sensitive applications such as video conferencing and voice over IP. This work presents the framework Context-Aware Adaptive Routing Framework SDN (CAARF-SDN), a solution for optimizing network traffic, compatible with any SDN controller, through a dynamic and proactive context-based approach. In this work, is applied the optimization of voice traffic addressing the change of the priority of the flow and the change of topology, basing the notion of context on the metrics of quality of service, experience and device (QoE, QoS and QoD) demonstrating the optimization of network resources in adverse scenarios.*

**Resumo.** *Software-Defined Networking (SDN) é um paradigma de rede que permite a aplicação de novas técnicas de engenharia de tráfego através da separação do plano de controle do plano de encaminhamento. De forma a dar suporte às características dinâmicas do ambiente computacional (tais como, condições dinâmicas de tráfego e variedade protocolos de rede, mobilidade do usuário, heterogeneidade de dispositivos, entre outros), técnicas inovadoras de engenharia de tráfego devem ser propostas para lidar com aplicações sensíveis às variações de atraso, tais como videoconferência e voz sobre IP. Neste trabalho apresentamos o arcabouço Context-Aware Adaptive Routing Framework SDN (CAARF-SDN), uma solução para otimização do tráfego de redes, compatível com qualquer controlador SDN, através de uma abordagem dinâmica e proativa baseada em contexto. Neste trabalho, está aplicado a otimização de tráfego de voz abordando a mudança da prioridade do fluxo e a mudança de topologia, baseando a noção de contexto nas métricas de qualidade de serviço, experiência e dispositivo (QoE, QoS e QoD) demonstrando a otimização dos recursos de rede em cenários adversos.*

## 1. Introdução

As redes de computadores modernas evoluíram para um ambiente altamente complexo e heterogêneo, que compõe um desafio de gerência e escalabilidade de recursos para cumprimento dos requisitos de SLA das aplicações e dos usuários de forma otimizada.

As redes convencionais, que possuem os planos de controle e encaminhamento integrados nos elementos de rede, apresentam desafios de otimização para operadores que precisam atender serviços críticos e sensíveis à variação de qualidade como voz e vídeo. As redes definidas por software (SDNs) apresentam um novo paradigma para redes de computadores, com a proposta de desacoplamento dos planos de controle e de encaminhamento centralizando o plano de controle no controlador SDN. Com base na inédita perspectiva da topologia global da rede que o controlador SDN possui, torna este novo elemento um componente ideal para a otimização do comportamento do plano de dados, maximizando as possibilidades de gerenciamento, controle e otimização.

No entanto, como a demanda por recursos de rede aumentou exponencialmente nos últimos anos, é possível observar algumas limitações na infraestrutura de comunicação: (1) A capacidade de *backbone* não aumenta conforme a demanda de tráfego, conotando a necessidade da aplicação de engenharia de tráfego e otimização com uma abordagem distribuída, e tal demanda é possível graças ao paradigma de redes definidas por software [Goransson et al. 2017] ; (2) A atual natureza convergente das arquiteturas digitais associadas à crescente demanda por tráfego sensível em tempo real torna as soluções baseadas apenas em Qualidade de Serviço (QoS) insuficientes [Song et al. 2016]; (3) Apesar da arquitetura de SDN fornecer um gerenciamento de rede flexível e confiável, suportando a programação de fluxo inteligente para a melhoria da utilização de link, a configuração dentro de controladores SDNs ainda é realizada estaticamente, o que não permite o alinhamento com natureza dinâmica dos sistemas (redes, usuários e dispositivos de apresentação), e; (4) As soluções de reprogramação de redes baseadas em SDN, são associadas a um controlador em específico, não sendo agnóstica a controladores SDN [Neto et al. 2018].

Neste trabalho apresentamos uma solução de otimização baseada em contexto (mesclando aspectos centrados no usuário com aspectos centrados na rede) aplicada a redes SDN com objetivo da otimização do uso dos recursos de rede e em consequência a melhoria da experiência do usuário. A solução proposta, chamada de *Context-Aware Adaptive Routing Framework applied to SDN* (CAARF-SDN) [Neto et al. 2018], surge como uma alternativa para o processo de análise e otimização dinâmica do uso dos recursos da rede.

O arcabouço tem como proposta a análise de diversas métricas (KPI – Indicadores Chaves de Desempenho), que podem ser categorizadas em dados de Qualidade de Serviço (QoS), Qualidade de Dispositivo (QoD) e Qualidade de Experiência (QoE). QoS abrange as informações pertinentes às condições de tráfego de rede; QoD se refere aos dados de desempenho e condições dos dispositivos de rede (seja elemento de rede ou dispositivo do usuário), e; QoE reflete a percepção do usuário em relação a qualidade do serviço prestado. O lado inovador do CAARF-SDN se encontra no uso integrado dessas métricas, de forma a promover a engenharia e otimização de tráfego, visando a melhoria da percepção do usuário, mudando a natureza do tráfego (como sua prioridade) ou a topologia da rede (modificando caminhos).

As principais contribuições apresentadas neste artigo são:

1. A apresentação de uma solução dinâmica e escalável para fornecer otimização adaptável ao tráfego sensível ao contexto através do arcabouço CAARF aplicado às redes SDN (CAARF-SDN);

2. A proposta de um modelo de contexto integrado, baseado nos conceitos de Qualidade de Serviço (QoS), Qualidade de Experiência (QoE) e Qualidade de Dispositivo (QoD) para roteamento adaptativo, com a proposta de uma métrica de Qualidade de Contexto (QoC);
3. A representação de um modelo genérico, usando *Javascript Object Notation* (JSON) para descrever todo o modelo proposto;
4. A apresentação de um experimento, através de um modelo simulado para demonstrar a solução proposta, aplicada em SDN otimizando tráfego de voz em cenários com congestionamento com a respectiva disponibilização pública;

Este artigo está organizado da seguinte forma: As Seções 1 e 2 compõem de forma resumida a introdução ao trabalho e um breve levantamento dos trabalhos relacionados; Na Seção 3 é descrita toda a arquitetura da solução proposta; Na seção 4 é descrito e especificado todo o experimento de simulação realizado onde podemos aferir os resultados, e; por fim, na Seção 5 são apresentadas as conclusões do trabalho e relacionadas as perspectivas de trabalhos futuros.

## 2. Trabalhos Relacionados

Através da revisão de literatura, podemos constatar a ampla variedade de contribuições correlatas à temática de otimização de redes. De forma a compararmos algumas dessas abordagens existentes, foram selecionados alguns critérios que indiquem a sua representatividade dentro do domínio estudado tais como: (1) Qualidade de Serviço (QoS), através do uso/adoção dos indicadores-chave de desempenho (KPIs) para QoS como latência, perda de pacotes, jitter, entre outros; (2) Qualidade de Experiência (QoE), que denota a percepção da qualidade na perspectiva do usuário; (4) Qualidade de Dispositivo (QoD), através do uso de KPIs do dispositivo do dispositivo do usuário ou elemento da rede como a carga de cpu ou de memória; (5) Abordagem agnóstica à aplicação, capaz de otimizar qualquer aplicação em rede; (6) Abordagem agnóstica ao controlador SDN, capaz de ser executado na interface *northbound* (NBI) de qualquer controlador SDN; (7) Abordagem *multivendor* - critério para estabelecer se a solução é capaz de ser aplicada em equipamentos de diversos fornecedores, e; (8) Disponibilidade de Código - critério para estabelecer se a solução tem seu código totalmente disponibilizado na internet para o esforço evolutivo comunitário. Observando esses critérios comparativos, estudamos as seguintes contribuições na literatura:

[Xezonaki et al. 2018], propõem uma aplicação SDN que monitora e preserva o QoE para voz e vídeo sobre IP, monitorando periodicamente vários parâmetros de rede e usando modelos de estimativa de QoE, entretanto a otimização é restrita às aplicações de voz e vídeo não sendo aplicável a outros tipos de tráfego, também não considera métricas do dispositivo (QoD) para a tomada de decisão; [Mon and Mon 2018] propõem um mecanismo para o monitoramento de entradas de fluxo de rede entre os switches, analisando os KPIs de QoS e encontrando um caminho ótimo modificando a tabela de roteamento usando *Segment Routing* sob SDN. No entanto, não considera métricas de QoD, e nem considera estimativas de QoE. Da mesma forma, também não se propõe a ser agnóstico a controladores sendo implementado em ONOS; [Shimokawa et al. 2021] propõem técnicas de medição e estimativa de QoE da rede baseado em dois estágios com foco em aplicações de vídeo apenas, convertendo informações da rede e informações do dispositivo final para a estimativa do QoE utilizando SDN, entretanto não considera

métricas de QoD e também não considera uma abordagem agnóstica a controlador SDN sendo implementado no *framework* trema.

Em comparação com as abordagens encontradas na revisão da literatura, é possível observar que, o CAARF-SDN não é orientado/limitado a uma métrica específica, ou aplicação (vídeo ou voz) ou um tipo de rede (fixa ou móvel). Também não é baseado em apenas um controlador SDN. Ademais, o CAARF-SDN não pode ser confundido com uma plataforma de operação, administração e gerenciamento de redes (a exemplo de prometheus e similares), pois além de apoiar no monitoramento do tráfego de rede, também considera os dados de contexto coletados para a reprogramação da rede seja através da modificação das prioridades do tráfego ou da mudança de topologia de acordo com as necessidades do tráfego. Analisando o mercado, dois fornecedores de renome (Cisco e Juniper) tem soluções similares, Cisco DNA Center [Cisco 2021] e Juniper Contrail [Juniper 2021], entretanto insta salientar que ambas as soluções além de serem soluções proprietárias (cujo código não está publicamente disponível) também são soluções de alto custo e fechadas aos seus produtos. Vale a pena ressaltar que o CAARF-SDN tem seu código disponível publicamente na internet para a colaboração comunitária. A Tabela 1 ilustra um resumo comparativos entre as abordagens estudadas.

**Tabela 1. Trabalhos Relacionados**

ABORDAGENS	QoS	QoE	QoD	AGNÓSTICO APLICAÇÃO	AGNÓSTICO SDN	MULTI VENDOR	DISP. CÓDIGO
Xezonaki et al	X	X	-	-	-	X	-
Mon e Mon	X	-	-	X	-	X	-
Shimokawa et al	X	X	-	-	-	X	-
Juniper Contrail	X	X	X	X	-	-	-
Cisco DNA Center	X	X	X	X	-	-	-
Caarf-SDN	X	X	X	X	X	X	X

### 3. CAARF-SDN

#### 3.1. Visão Geral da Arquitetura

Sob o ponto de vista de implementação aplicada às redes definidas por software, o CAARF-SDN é uma aplicação de rede que atua na interface *northbound* (NBI) de um controlador SDN. Uma visão geral da arquitetura CAARF-SDN é descrita em [Neto et al. 2018] e sua estrutura é ilustrada na Figura 1. O fluxo de funcionamento do CAARF-SDN pode ser resumido nos seguintes passos:

(1) O Módulo de Coleta monitora o estado da rede, coletando os dados de QoS, QoD, estimando as informações de QoE e conforme as alterações, gera uma notificação de mudança de contexto. (2) O Módulo de Otimização, ao receber esta notificação, aplica algoritmos de otimização para determinar que ações devem ser tomadas com o objetivo de otimizar o tráfego e determinar melhor rota para sua entrega. O resultado desta dá origem as diretivas de configuração, que são enviadas ao (4) Módulo de Atuação para realizar o mapeamento para uma diretiva genérica de configuração, originando uma notificação para o (5) Módulo de Integração que, através do *gateway* relacionado ao controlador SDN utilizado, efetiva a reprogramação da rede, processando as diretivas de configuração. Estas instruções irão alterar as informações de controle de fluxo de tráfego, presente nas tabelas de fluxo dos elementos de rede.

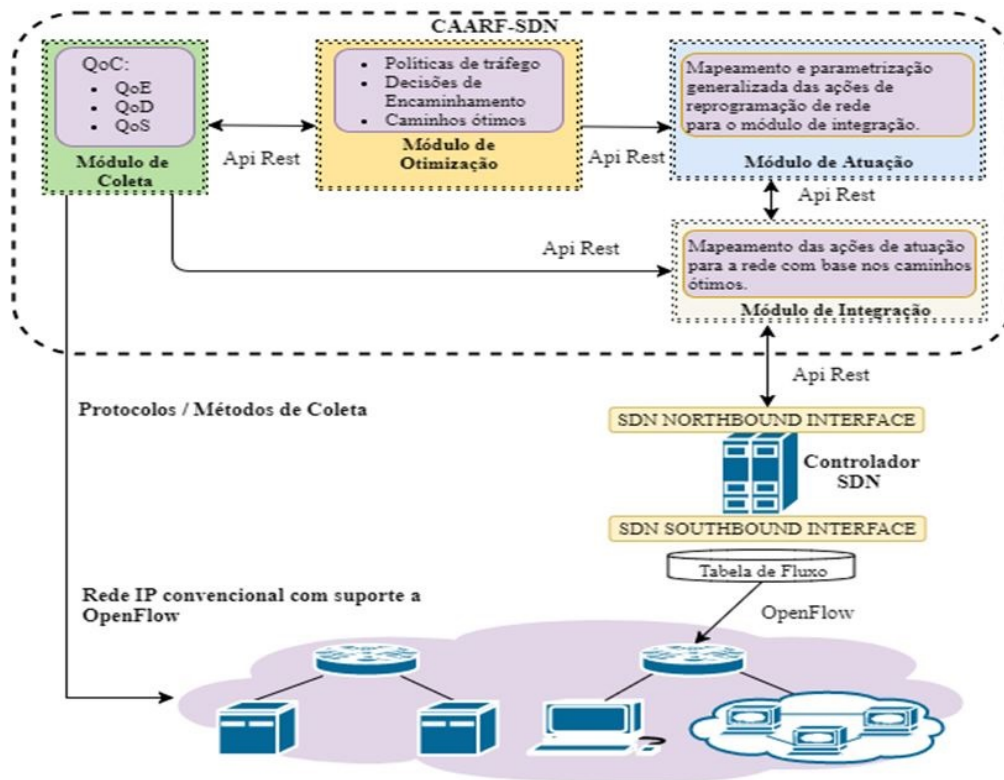


Figura 1. Arquitetura CAARF-SDN

### 3.2. Módulo de Coleta

O Módulo de Coleta de Dados usa os dados obtidos dos elementos de rede e da comunicação fim-a-fim do tráfego dos serviços para definir as métricas de QoS, QoD e QoE, necessárias para o cálculo do QoC (Qualidade de Contexto), conforme descrito em [Miguel et al. 2019]. Estas métricas, quando analisadas em conjunto, definem um contexto de análise de dados mais amplo do que a análise de cada variável individualmente.

Conforme descrito em [Miguel et al. 2019], o cálculo de QoC é realizado para cada porta do elemento de rede, tendo como referência a saída do fluxo de dados. O resultado do cálculo sofre uma interpolação de forma que seu resultado varie entre os números 1 a 5, coerentes com a *Mean Opinion Score* (MOS) [Neto et al. 2018]. Alguns eventos que após observações e análise de resultados, concluímos que afetam o valor do QoC impactando no MoS final, no contexto das aplicações de voz sobre IP, são: (a) Congestionamento da interface da rede acima de 95% ; (b) Latência de rede alta acima de 90ms; (c) *Jitter* elevado acima 60ms, e; (d) Alto consumo de CPU acima de 90%. Em situação de decréscimo de qualidade, uma notificação (notificação de mudança de contexto) é enviada para o Módulo de Otimização. Esta notificação contém o estado de QoC de toda a rede monitorada, para aferição no Módulo de Otimização. Na Figura 2, apresentamos o cálculo do QoE estimado, conforme descrito em [Neto et al. 2018]. As métricas de QoC compõem a avaliação conjunta dos dados de QoS, QoD e QoE. O cálculo do QoE atua como um gatilho para disparar a execução do processo de otimização. Um valor de QoE atingindo um limite igual ou inferior a 2, indica um comprometimento na qualidade da comunicação de voz, conforme é demonstrado pelo ITU-T para o cálculo

do MOS estimado [ITU-T 2016], chamado neste artigo de mosQoE. A reprogramação dos caminhos dos fluxos tem como objetivo interromper o processo de queda do QoE, de forma a manter a qualidade de transmissão dentro de parâmetros aceitáveis pelo usuário.

Latência Média	Delta	Perda de Pacotes	Latência Efetiva	R	MOS-CQE
3,2	---	0	19,6	92,71	4,3996
6,8	3,6	0	23,2	92,62	4,3977
9,6	2,8	0	26,0	92,55	4,3963
Jitter	3,2	---	---	---	---

Figura 2. Fórmula e Resultado de QoE Estimado

### 3.3. Módulo de Otimização

O módulo de otimização tem como função a seleção dos melhores caminhos ou a aplicação de políticas de otimização do fluxo com base em informação contextual gerada pelo Módulo de Coleta. Portanto, quando uma variação de contexto é observada, uma notificação de mudança de contexto é gerada pelo módulo de coleta e enviada ao módulo de otimização.

O módulo de otimização analisa a topologia de rede dinamicamente com as informações enviadas pelo módulo de coleta, determinando possíveis vias de encaminhamento. Depois disso, o módulo de otimização envia ao módulo de atuação as diretivas de configuração e mapeamento que devem ser aplicadas sobre a rede.

Conforme apresentado em [Neto et al. 2018], no *core* do módulo de otimização está o ranqueamento de caminhos onde os submódulos de informações de encaminhamento e processamento de caminho baseado em contexto que depende do módulo de seleção de caminhos vão efetuar o *post* no submódulo de políticas de encaminhamento. Após esta ação, serão geradas as diretivas de modificação/ajuste para a operação de otimização da rede (Como uma modificação em um fluxo ou mudança na topologia) a ser executada pelos módulos de atuação e integração.

Este submódulo do *core* do módulo de otimização (O ranqueamento de caminhos) é que vai gerar as diretivas de configuração a serem enviadas via API ao módulo de atuação.

### 3.4. Módulo de Atuação

O módulo de atuação tem como função o mapeamento das variáveis/índices definidos pelo módulo de otimização associando a diretivas de configuração armazenados no banco de dados. Através de um diálogo em JSON, generalista em relação as funções desempenhadas, o módulo de atuação funciona como uma camada que garante o diálogo generalizado do arcabouço deixando-o compatível com diversos tipos de redes. Em seu fluxo de funcionamento, o módulo de atuação passa para o módulo de integração, com um diálogo JSON

generalizado as diretivas de configuração que serão processadas no módulo de integração. A figura 3 ilustra a arquitetura conceitual do módulo de atuação e seu relacionamento com os módulos de otimização e integração.

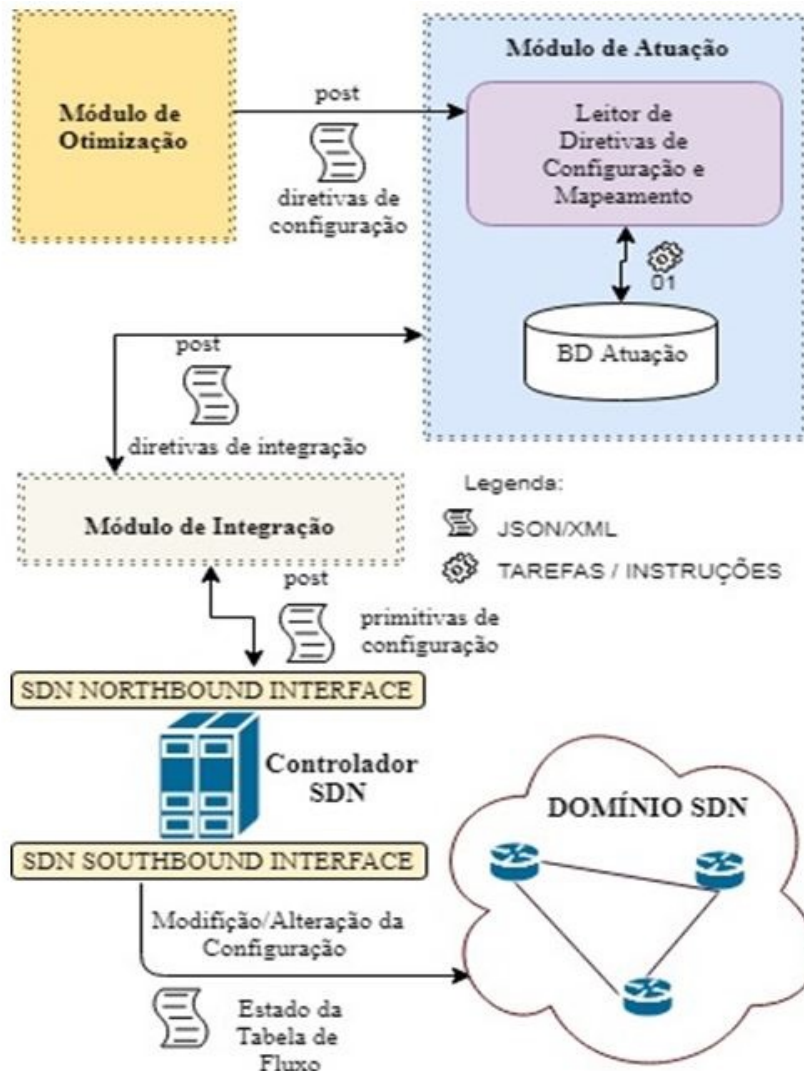


Figura 3. Arquitetura Conceitual Módulo de Atuação

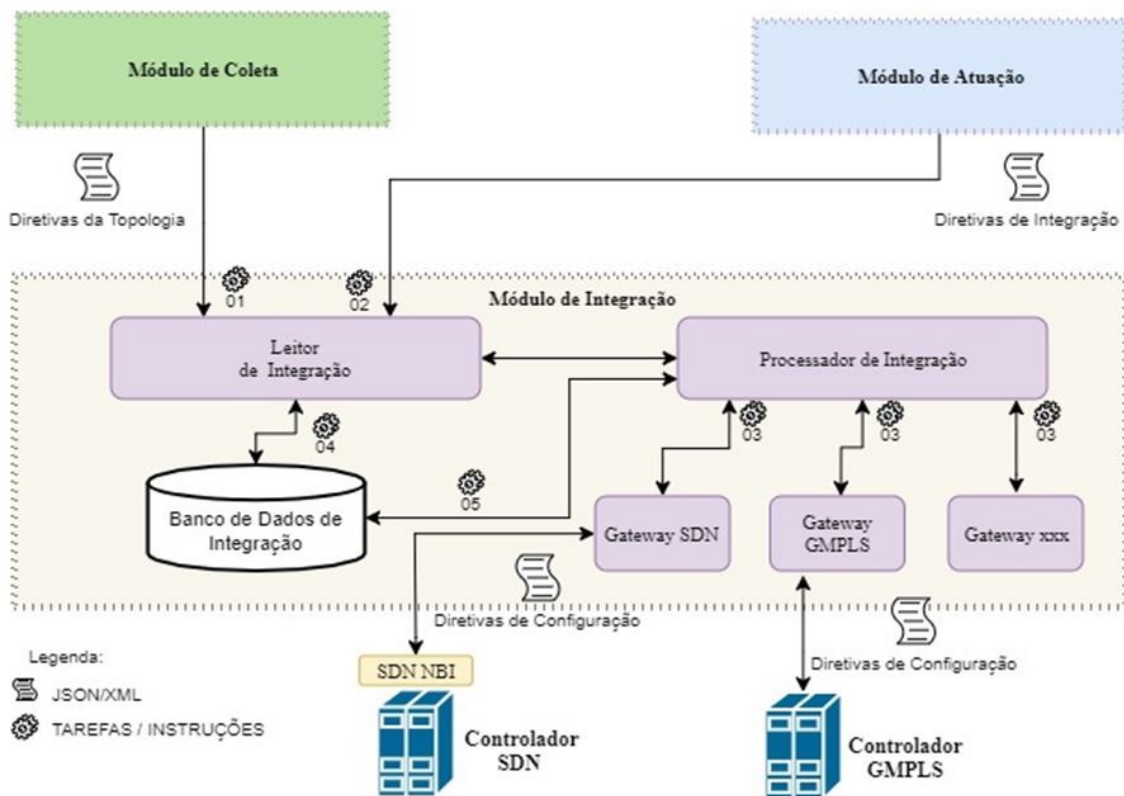
Conforme ilustrado na figura 3, o módulo de atuação realiza a leitura das diretivas de configuração enviadas pelo módulo de otimização e realiza a persistência das diretivas em banco de dados para consulta futura. Em seguida, realiza o *post* via API no módulo de integração enviando as diretivas de integração utilizando o diálogo JSON generalizado do arcabouço.

### 3.5. Módulo de Integração

O objetivo principal do módulo de integração é reprogramar a tabela de fluxos do controlador SDN baseado na configuração de diretivas emitidas pelo módulo de atuação. Além do objetivo principal, também é função do módulo de integração a conexão com o controlador SDN para a coleta de dados da rede (a exemplo de informações sobre a topologia da rede), disponibilizando tais informações para o módulo de coleta através do leitor de

integração. Esta configuração é realizada através das funções da API *northbound* do controlador de SDN, interface que permite a interação externa com o controlador SDN. As funções da API *northbound* do controlador SDN (que tem pequenas diferenças entre os diversos controladores) são mapeadas, através dos *gateways* do módulo de integração.

Cada *gateway* tem o mapeamento das funções e parâmetros de configurações de cada controlador, e por design pode ser ampliado. Uma vez que a tabela de fluxo do controlador é reconfigurada, atualiza o controlador SDN, comuta a tabela de fluxo através da sua interface *southbound* e despacha a reconfiguração de rede. O módulo de integração também armazena toda a configuração realizada no controlador SDN mantendo uma base de dados como registros de ação para outras estatísticas consultas. A interação com a interface NBI do controlador SDN permite diversos controladores devido a existência dos *gateways* de SDN que são elementos de compatibilidade com os diversos controladores existentes e suas eventuais individualidades de configuração e aspectos. A Figura 4 ilustra a arquitetura conceitual do módulo de integração, onde é possível observar sua interação via API com (i) o módulo de atuação, de quem recebe as diretivas de integração; (ii) o Controlador SDN através de sua interface *northbound* API, por onde é possível reconfigurar dinamicamente a tabela de fluxos do controlador; (iii) a base de dados de configuração, para a estatística e registro da configuração aplicada.



**Figura 4. Arquitetura Conceitual do Módulo Integração**

Os componentes do módulo de integração conforme ilustrado na figura 4, recebem e processam as diretivas primitivas de integração e topologia enviados pelos módulos de coleta e atuação, realizando a persistência dos dados para uso futuro quando necessário e sendo processado pelo componente processador de integração onde ocorre a padronização



do diálogo para os *gateways* e cada um destes é projetado para a produção das diretivas específicas de configuração que é realizado na rede.

Os *gateways* do módulo de integração garantem a característica agnóstica da solução proposta.

#### 4. Experimento e Análises

O objetivo do experimento é demonstrar como as variáveis de QoC, QoD e QoE interagem entre si na solução CAARF-SDN e como seus processos podem resultar em um melhor uso dos recursos alocados em uma rede de computadores.

O experimento realizado foi desenvolvido em um ambiente virtualizado conforme descrito em [Miguel et al. 2020] sendo utilizado para mostrar a viabilidade do cálculo do QoC, com base no controlador HP SDN Controller demonstrando a viabilidade da solução proposta. Neste artigo, iremos explorar dois cenários de otimização, um por priorização do tráfego e outro por mudança de porta. O ambiente de rede SDN é emulado através do programa Mininet [ONF 2021].

Esta solução permitirá que outros pesquisadores possam reproduzir os experimentos realizados pois todo o código está disponível publicamente no Github. Para este experimento, a aplicação CAARF-SDN, aborda uma aplicação típica de tráfego com taxa de bit constante (*Constant Bit Rate* – CBR), que descreve o tráfego associado ao áudio que impõe maiores restrições à comunicação, qualquer limitação afetará negativamente a qualidade de percepção do usuário sobre o conteúdo entregue. A solução tradicional se baseia no uso apropriado de *codecs*, como os G.711, G.723.1 e G.729 por exemplo associado a, quando possível e viável, uma política de QoS.

O uso do *codec* apropriado vai depender dos parâmetros de QoS analisados como *jitter* médio, latência, perda de pacotes e banda disponível [Gupta et al. 2018]. A solução proposta com o CAARF-SDN visa atuar na rede, como uma forma de otimizar a comunicação percebendo a mudança de contexto e atuando para manter os níveis mínimos de SLA. Conforme a topologia proposta em [Miguel et al. 2020] para este trabalho foi realizada uma simulação, composta por quatro elementos de rede, fornecendo dois caminhos possíveis para o tráfego entre os *hosts*. No cenário proposto existe *host* que hospeda a solução CAARF-SDN e os softwares de monitoria Prometheus [Prometheus 2021] e o gerenciador gráfico Grafana [Grafana 2021].

O CAARF-SDN foi escrito em Javascript, com base no serviço Node.js [OpenJS 2021]. Os dados são armazenados em uma base de dados NoSQL MongoDB [mongoDB 2021]. O tráfego entre os elementos na simulação é gerado através do *software* Linux SIPp [Sipp 2021], configurado para gerar um fluxo de dados baseado no protocolo RTP, com uma amostra exemplo baseada no *codec* G.711. Na topologia proposta, um *host* tem a função de gerador de tráfego de ruído simulando um congestionamento na rede para os cenários de experimentação propostos, descritos nas subseções seguintes.

##### 4.1. Geração de tráfego

O método usado para a geração do tráfego é o mesmo para os dois cenários. Usando o *software* SIPp, onde um *host* fará o papel de *User Agent Server* (UAS), e um outro *host* fará o papel de *User Agent Client* (UAC). A troca de mensagens é executada de

acordo com o especificado pelo protocolo SIP. A largura de banda configurada foi de 10 Mbps, com atraso de 0 ms. Este valor foi ajustado para suportar o tráfego RTP sem erros de transmissão. Devido às limitações do ambiente, não foi possível usar no Mininet o programa Startrinity [Startrinity 2021], que poderia dar informações adicionais do tráfego mais precisas. Foi utilizado o programa Wireshark [Wireshark 2021] para analisar os pacotes e fazer a análise de resultados.

#### 4.2. Cálculo do QoC e do QoE

Os dados do QoC são calculados com base nos parâmetros de QoS de cada porta do elemento, fornecidos pelo controlador SDN. Os dados de QoD (como consumo de CPU e memória) de cada elemento de rede também são coletados. Em um ambiente físico real, os dados do QoD iriam depender de um agente de telemetria ou SNMP instalado em cada elemento de rede para a coleta. Os dados do QoE, são calculados com base na geração de dados feito pelo programa MTR [TravisCross 2021] prospectando via ICMP o *jitter* e o RTT que serão base nesta versão da solução, para os dados necessários ao cálculo do QoE. Estas informações serão processadas pelo Módulo de Coleta do CAARF-SDN e o valor obtido no QoE (piorando) é a base para o disparo do processo de otimização.

#### 4.3. Cenário I – Alteração da prioridade do tráfego

O Cenário I demonstra o impacto da reprogramação dinâmica da rede, ilustrado pela Figura 5, trabalhando com a mudança de prioridade do tráfego. A duração do experimento foi limitada pelo tamanho do arquivo com os dados capturados, devido a limitações do hardware utilizado.

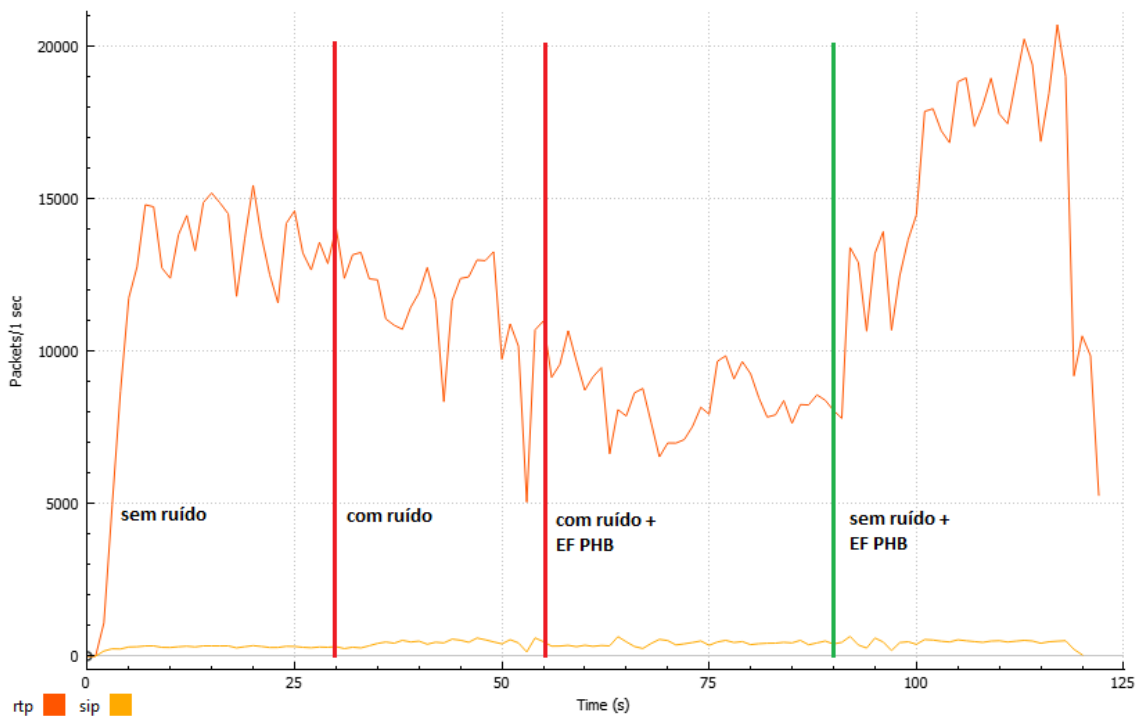


Figura 5. Comportamento do tráfego da rede no Cenário I

Com a rede criada através do Mininet [ONF 2021], foi gerado um cenário onde se testou situações com ruído e sem ruído e com modificação do DSCP em situações ad-

versas para se avaliar a efetividade do procedimento. Conforme pode ser visto na Figura 5, o experimento tem 4 fases distintas. O tempo do experimento consumiu aproximadamente 120 segundos de capturas de tráfego de rede feitas pelo programa Wireshark [Wireshark 2021]. O procedimento foi iniciado com um ambiente sem ruídos e configurado para trabalhar com o serviço de rede em modo de melhor esforço. Neste cenário, foi obtido um pico de transmissão de pacotes por volta de 15.000 pacotes por segundo.

Após 30 segundos de teste, foi injetado ruído na comunicação e foi observado uma rápida degradação da taxa de transmissão dos pacotes chegando a um mínimo de 5000 pacotes por segundo. Durante o experimento, o Módulo de Coleta do CAARF-SDN analisou o valor do mosQoE, enviado pelo agente de coleta instalado no host analisado. Quando o mosQoE atingiu um valor abaixo de 2, o processo de alteração das características do tráfego é aplicado. Este evento ocorre ao final dos 60 segundos de experimento, a aplicação CAARF-SDN envia uma instrução ao Controlador SDN alterando o valor do campo DSCP, através da reprogramação do fluxo dos elementos de rede mudando o campo de 0 (*Best Effort*) para 46 (*Expedited Forwarding Per-Hop Behavior / EF PHB*), que na definição da arquitetura dos Serviços Diferenciados (DSCP), é a configuração ideal para atender demandas que são mais sensíveis a atrasos na rede, conferindo a prioridade máxima de transmissão na rede, conforme está especificado na RFC 3246. Neste cenário, foi identificado que processo de degradação do envio de pacotes foi interrompido e se manteve praticamente estável durante os próximos 30 segundos do experimento.

Tal ação demonstrou que a mudança do DSCP foi capaz de mitigar o processo de degradação da comunicação para um nível mínimo. Quando foi alcançado os 90 segundos do experimento, a ingestão de ruído na rede foi suspensa e, pode-se notar que a taxa de transmissão de pacotes subiu consideravelmente a um patamar ainda mais elevado de quando o tráfego estava configurado para melhor esforço e sem ruído.

Com isso, podemos concluir que a alteração dinâmica do DSCP, feita pelo CAARF-SDN, teve o efeito desejado neste ambiente emulado. O Cenário II vai demonstrar o impacto da reprogramação dinâmica da rede, trabalhando com a mudança de portas no elemento de rede, para criar uma rota alternativa.

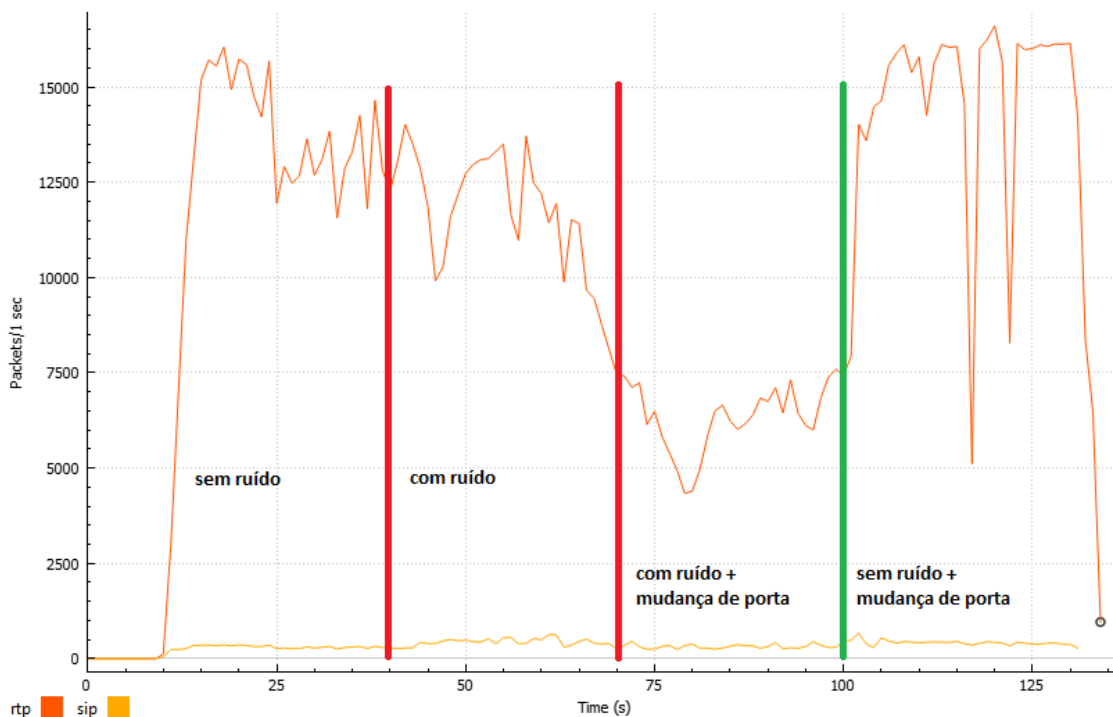
Para efeitos comparativos, o mesmo intervalo de tempo e as mesmas configurações de transmissão foram utilizadas no emulador.

#### **4.4. Cenário II – Alteração no fluxo com mudança de porta/Mudança de Topologia**

A topologia inicial conforme o cenário I ilustrada em [Miguel et al. 2020] será a partida para o cenário II. A topologia tem 3 *switches* interconectados entre si e os 4 *hosts*, estabelecendo dois caminhos entre os hosts analisados. A ideia deste cenário II é que, com o congestionamento, um segundo caminho existente entre o host de origem e o de destino, fosse escolhido em alternativa ao principal inicialmente utilizando e sob congestionamento.

Com isso, como resultado dois comportamentos foram observados: (1) aumento da carga de processamento em um dos hosts, obtida de forma artificial no ambiente simulado; (2) atraso e perdas de pacotes como resultado da geração de ruído, sendo que estes dados são coletados do Controlador SDN e do agente de coleta responsável pelas capturas das variáveis necessárias para o cálculo do mosQoE.

Como critério de tempo, o experimento usou aproximadamente os mesmos 120 segundos do Cenário I. Os primeiros 30 segundos sem ruído, com um pico de 15000 pacotes por segundo. Nos 30 segundos seguintes, foi adicionado o ruído. Nota-se uma queda na taxa de transmissão de pacotes, conforme ilustrado na Figura 6. Essa queda na transmissão, influencia diretamente na leitura dos dados do mosQoE, com o módulo de coleta gerando uma notificação de mudança de contexto, sendo o ponto de partida para a ação de otimização, neste cenário, a mudança da topologia da rede com a escolha do caminho secundário, sem o congestionamento degradante do caminho principal. O novo



**Figura 6. Comportamento do tráfego da rede no Cenário II**

fluxo não elimina o antigo, que ainda dá vazão aos pacotes que ainda se encontram sobre sua responsabilidade, mas os novos pacotes que chegam já utilizam a nova rota. Nota-se neste momento, que o processo de perda de pacotes para de cair, apresentando uma leve recuperação. Ao final deste período, o ruído é suprimido, e praticamente temos tráfego nas mesmas condições quando do início do experimento. Desta forma, concluímos que a reprogramação dinâmica executada pelo CAARF-SDN, em um ambiente emulado é capaz de otimizar o fluxo de rede nas duas situações propostas.

#### 4.5. Lições Aprendidas

Durante a execução do experimento observamos os aspectos práticos da proposta realizada, devido ao cenário de emulação com o Mininet tivemos algumas limitações em comparação a um cenário utilizando equipamentos reais. Entretanto a simulação permitiu a comprovação prática do presente trabalho de pesquisa. Portanto, conforme os resultados apresentados, foi possível observar que:

1. O ambiente de emulação do Mininet tem a limitação na questão da não acurácia para a prospecção da carga de cpu devido as características do ambiente de emulação e o respectivo compartilhamento dos recursos computacionais;

2. Não é possível prospectar o diferencial da implantação com elementos de rede de múltiplos fornecedores (para podermos prospectar as questões de carga de cpu e *jitter*, que ocorrem em cenários reais diante da diversidade de fabricantes e das respectivas características individuais dos equipamentos (Como tipo de cpu, capacidade de processamento, de memória e *backplane* do elemento), tendo em vista o ambiente do *OVS* generalizado, do ambiente simulado, e;
3. Para uma melhor acurácia na prospecção do QoC, faz-se necessário ambiente real de teste.

## 5. Conclusões

Neste artigo, foi apresentado o CAARF-SDN, uma solução para a otimização dinâmica de redes, aplicada em SDN. Tendo como diferencial as características-chave de ser compatível com qualquer controlador SDN e em ser solução de software livre com código fonte publicamente disponível. Os resultados experimentais mostram que a solução CAARF-SDN proporciona uma otimização cuja qualidade mensurada pelo QoE é perceptível principalmente em cenários de congestionamentos seja pela modificação dinâmica das prioridades de tráfego até a mudança da topologia com a escolha de novos caminhos para o tráfego o que indica sua utilidade para ISP's/Operadores de telecomunicações com *backbones* multisserviços que possuem serviços com alto SLA e necessitam de reprogramação dinâmica e distribuída da sua rede, em função do cumprimento de SLA estabelecido.

O código produzido neste trabalho está disponível no GitHub em <https://github.com/caarf-sdn> para a colaboração comunitária. Como trabalho futuro, para o módulo de coleta relacionamos a implantação da coleta com o protocolo ipfix para uma melhor prospecção de dados sobre o tráfego. Propomos também como trabalho futuro a adoção de agente de protocolo para extensão das funções de telemetria (melhoria para a coleta de dados para o QoD e QoS), a exemplo de um agente gRPC. Também relacionamos como trabalho futuro a implantação de agentes para análise na camada da aplicação, para prospecção de dados para melhor estimativa de qualidade como o RTT de aplicação, taxas de perdas e o *jitter*, a exemplo da análise via RTP/RTCP. Para o módulo de integração, relacionamos como trabalho futuro a necessidade do desenvolvimento de um número maior de *gateways* permitindo uma maior abrangência e cobertura de controladores SDN. Reforçando a característica da solução proposta de ser aplicada em qualquer controlador SDN. Assim como *gateways* para GMPLS e ASON, reforçando a característica de aplicação em qualquer tipo de rede.

## Referências

- Cisco (2021). Cisco dna center. <https://www.cisco.com/c/en/us/products/cloud-systems-management/dna-center/index.html>. Acesso em: 28/07/2021.
- Goransson, P., Black, C., and Culver, T. (2017). *Software Defined Networks, A Comprehensive Approach. Segunda Edição*. Editora Elsevier, 978-0-12-804555-8.
- Grafana (2021). Grafana. <https://grafana.com/>. Acesso em: 28/07/2021.
- Gupta, N., Kumar, N., and Kumar, H. (2018). Comparative analysis of voice codecs over different environment scenarios in voip. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 540–544.

- ITU-T (2016). P.800.1 : Mean opinion score (mos) terminology. <https://www.itu.int/rec/T-REC-P.800.1>. Acesso em: 28/07/2021.
- Juniper (2021). Juniper contrail. <https://www.juniper.net/br/pt/products-services/sdn/contrail/>. Acesso em: 28/07/2021.
- Miguel, C. J., Neto, F. B., and Sampaio, P. N. M. (2019). Data collection in sdn networks with contextual analysis. In *14th Iberian Conference on Information Systems and Technologies. Coimbra, Portugal”(CISTI)*, pages 1–6.
- Miguel, C. J., Neto, F. B., and Sampaio, P. N. M. (2020). The caarf approach towards monitoring and analysis of contextual data within sdn networks. In *8th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2020)*, hal-02495244, version 1, pages 1–8, Cancun, MX. Advance.
- Mon, O. M. and Mon, M. T. (2018). Quality of service sensitive routing for software defined network using segment routing. In *2018 18th International Symposium on Communications and Information Technologies”(ISCIT)*, pages 180–185.
- mongodb (2021). Mongoddb. <https://www.mongodb.com/>. Acesso em: 28/07/2021.
- Neto, F. B., Miguel, C. J., Santos, J. A., and Sampaio, P. N. M. (2018). Context-based dynamic optimization of software defined networks. In *2018 6th International Workshop on ADVANCEs in ICT Infrastructures and Services, Santiago De Chile, Chile.”(ADVANCE 2018)*, pages 1–6.
- ONF (2021). Mininet. <http://mininet.org/>. Acesso em: 28/07/2021.
- OpenJS (2021). Node.js. <https://nodejs.org/en/>. Acesso em: 28/07/2021.
- Prometheus (2021). Prometheus. <https://prometheus.io/>. Acesso em: 28/07/2021.
- Shimokawa, S., Taenaka, Y., Tsukamoto, K., and Lee, M. (2021). Sdn based in-network two-staged video qoe estimation with measurement error correction for edge network. *IEEE Access*, 9(10.1109/ACCESS.2021.3061488):39733–39745.
- Sipp (2021). Sipp. <http://sipp.sourceforge.net/>. Acesso em: 28/07/2021.
- Song, J., Yang, F., Zhou, Y., Wan, S., and Wu, H. R. (2016). Qoe evaluation of multimedia services based on audiovisual quality and user interest. *IEEE Transactions on Multimedia*, 18(3):444–457.
- Startrinity (2021). Startrinity. <http://startrinity.com/>. Acesso em: 30/07/2021.
- TravisCross (2021). Mtr. <https://github.com/traviscross/mtr>. Acesso em: 28/07/2021.
- Wireshark (2021). Wireshark. <https://www.wireshark.org/>. Acesso em: 30/07/2021.
- Xezonaki, M.-E., Liotou, E., Passas, N., and Merakos, L. (2018). An sdn qoe monitoring framework for voip and video applications. In *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks”(WoWMoM)*, pages 1–6.