Pós-Graduação em Ciência da Computação

ERLAN FONSECA DE SOUZA

ESTRATÉGIA DEFINIDA POR SOFTWARE PARA O COMPARTILHAMENTO E CONTROLE DINÂMICO DE TAXA DE TRANSMISSÃO EM REDES DE CAMPUS

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife
2017

ERLAN FONSECA DE SOUZA

ESTRATÉGIA DEFINIDA POR SOFTWARE PARA O COMPARTILHAMENTO E CONTROLE DINÂMICO DE TAXA DE TRANSMISSÃO EM REDES DE CAMPUS

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciências da Computação, como parte dos requisitos necessários à obtenção do título de Mestre em Ciências da Computação.

Orientador: Kelvin Lopes Dias

Recife
2017

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S729e Souza, Erlan Fonseca de
Estratégia definida por software para o compartilhamento e controle dinâmico de taxa de transmissão em redes de campus / Erlan Fonseca de Souza. – 2017.
81 f.: il., fig., tab.

Orientador: Kelvin Lopes Dias.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2017.
Inclui referências, apêndices e anexos.

1. Redes de computadores. 2. Redes definidas por software. I. Dias, Kelvin Lopes (orientador). II. Título.

004.6 CDD (23. ed.) UFPE- MEI 2018-055

Erlan Fonseca de Souza

Estratégia definida por software para o compartilhamento e controle dinâmico de taxa de transmissão em redes de campus

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre Profissional em 21 de dezembro de 2017.

Aprovado em: 21/12/2017.

BANCA EXAMINADORA

_____

Prof. Carlos André Guimarães Ferraz
Centro de Informática / UFPE

_____

Prof. Obionor de Oliveira Nóbrega
Universidade Federal Rural de Pernambuco

_____

Prof. Kelvin Lopes Dias
Centro de Informática / UFPE
(Orientador)

Dedico este trabalho à minha família, bem maior existente em minha vida.

AGRADECIMENTOS

Primeiramente, agradeço ao grande arquiteto do universo por me conduzir neste caminho de crescimento pessoal e profissional.

Agradeço à minha família, minha avó Ana, meus pais Maralúcia e Airon, minha irmã Aline e meu irmão Thiago, que sempre torceram e incluíram meus sonhos em suas orações. Agradeço à Sâmia, pela torcida, motivação, compreensão e afago nos momentos difíceis.

Agradeço ao meu orientador, professor Kelvin Lopes Dias, por compartilhar de sua sabedoria e ter apoiado o desenvolvimento de uma solução para a demanda da infraestrutura de rede do IFRO.

Aos amigos que já possuía e aos que ganhei nessa trajetória: Bruce, Alan e Marcos, pois compartilhamos dos mesmos anseios nos últimos anos. Aos amigos que ganhei nesta reta final: Éric e David, a esse último em especial, pelo compartilhamento de experiências e motivação para a conclusão.

Agradeço ao Instituto Federal de Rondônia pela oportunidade e apoio, honrando os compromissos firmados. Em especial, agradeço ao meu time, Osvino, Roseane, Danilo, Carlos e Daniel pela compreensão e suporte durante minhas ausências e limitações de desempenho das atividades cotidianas.

Emfim, agradeço a todos que, direta ou indiretamente, contribuíram e torceram para a conclusão deste mestrado, sonho que eu tanto almejava e agora, torna-se realidade.

'A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original' (Albert Einstein).

RESUMO

É notável a crescente demanda por conectividade. Tecnologias emergentes como internet das coisas (IoT), tudo como serviço (XaaS), traga seu próprio dispositivo (BYOD), entre outros existentes, demadam melhoramentos e inovação na infraestrutura vigente. Em ambientes acadêmicos, o uso destas tecnologias é constante e prover acesso de qualidade satisfatória, aos diversos perfis de usuário, é um desafio enfrentado pelos departamentos de tecnologia da informação. Atender a esta demanda através da contratação de serviço de conexão à internet com maiores capacidades de tráfego, tem um custo elevado e quando possível, possui qualidade questionável. Neste contexto, tecnologias para obter um melhor controle/aproveitamento do tráfego de rede foram desenvolvidas oferecendo maior qualidade de serviço (QoS) a estes ambientes. Dentre elas, técnicas de enfileiramento e limitação de tráfego através de filas (Queues) frequentemente são usadas por administradores de redes. A utilização destas técnicas impedem que um perfil com menor privilégio de tráfego prejudique outro com maior privilégio. Em contrapartida, por serem configuradas de forma estática, acabam por gerar desperdício de recursos de rede pois o que fora reservado para o perfil privilegiado, nem sempre é usado em sua totalidade.Vários fabricantes de equipamentos de redes oferecem soluções de QoS, contudo, são fechadas e geralmente associadas a equipamentos e licenças de alto custo. Recentemente, surgiram as redes definidas por software (SDN), um novo paradigma das redes de computadores de arquitetura dinâmica, gerenciável, econômica e adaptável.Esta dissertação tem o objetivo de propor uma solução SDN para o compartilhamento e controle dinâmico de taxa de transmissão, considerando as particularidades e demandas de uma rede de campus. Foi realizada uma investigação no ambiente de rede Instituto Federal de Rondônia – Campus Ariquemes, para que fossem identificados os perfis de tráfego e definidas políticas de priorização de taxa de vazão aos ambientes acadêmico e administrativo. Foi elaborada uma estratégia definida por software, através de um controlador SDN, em um ambiente virtual. O conjunto proposto envolve uma aplicação que através da ação de medidores monitora o tráfego de fluxos agregados de um ou mais domínios e, baseado nas políticas definidas, aplica dinamicamente a re-marcação de pacotes - caso o tráfego satisfaça a política - alterando sua priorização. O benefício ocorre quando o perfil privilegiado não utiliza a taxa de transmissão reservada e o perfil com privilégio inferior pode usufruir desse recurso. Testes de latência, perda e vazão foram realizados, em cenários variados e os resultados obtidos demonstraram que pode-se ampliar em até 33% o aproveitamento do enlace contratado.

Palavras-chave: QoS. Controle. Priorização. Redes definidas por Software - SDN. Ambientes Acadêmicos.

ABSTRACT

The growing demand for connectivity is noteworthy. Emerging technologies such as Internet of Things (IoT), everything as a service (XaaS), bring your own device (BYOD), among others exist, improve improvements and innovation in the current infrastructure. In academic environments, the use of these technologies is constant and providing satisfactory quality access to the various user profiles is a challenge faced by the information technology departments of these institutions. In this way, meeting this demand through the hiring of Internet connection service with greater traffic capacity, has a high cost and when possible, has questionable quality. In this context, technologies to obtain a better control / utilization of the network traffic were developed offering higher quality of service (QoS) to these environments. Among them queuing techniques and limiting queuing traffic are often used by network administrators. The use of these techniques prevents a profile with less traffic privilege from harming another one with greater privilege. In contrast, because they are configured in a static way, they end up generating wastage of network resources because what was reserved for the privileged profile is not always used in its entirety. Several network equipment manufacturers offer QoS solutions, however, they are closed and generally associated with expensive equipment and licenses. Recently, software-defined networks (SDN), a new paradigm of dynamic, manageable, economical and adaptable architecture of computer networks have emerged. This dissertation aims to propose an SDN solution for sharing and dynamic control of transmission rate, considering the particularities and demands of a campus network. An investigation was carried out in the Rondônia - Campus Ariquemes network environment, in order to identify the traffic profiles and prioritization of flow rate prioritization to the academic and administrative environments. A software-defined strategy was developed through an SDN controller in a virtual environment. The proposed set involves an application that, through the action of meters, monitors the traffic of aggregate flows of one or more domains and, based on the defined policies, dynamically applies the re-tagging of packages - if the traffic satisfies the policy - changing its prioritization. The benefit occurs when the privileged profile does not use the reserved transmission rate and the profile with lower privilege can enjoy it. Latency, loss and flow tests were performed in different scenarios from different perspectives and the results obtained were satisfactory with respect to the best use of the available transmission rate.

Keywords: QoS. Control. Prioritization. Software Defined Networks - SDN. Academic Environments.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BE	Best Effort
BYOD	Bring Your Own Device (Traga seu próprio dispositivo)
CSS	Cascading Style Sheets
DC	Datacenter
DS	Differentiated Services
DSCP	Differentiated services code point
EF	Expedited Forwarding
GB	Gigabyte
GUI	Graphical User Interface ou Interface Gráfica de Usuário
ID	Identificação Digital
IETF	Internet Engineering Task Force
IFRO	Instituto Federal de Educação, Ciência e Tecnologia de Rondônia
IP	Internet Protocol
ISP	Internet Service Provider
MPLS	Multiprotocol Label Switching
NAT	Network address Translation
NOS	Network Operational System
ONF	Open Network Foudation
OS	Operating System
RAM	Random Access Memory - Memoria de Acesso Aleatorio
REST	Representational State Transfer
SDN	Software Defined Networking
SNMP	Simple Network Management Protocol
SSL	Secure Socket Layer

TI	Tecnologia da Informação
VLAN	Virtual Local Area Network
VPN	Virtual private network
WAN	Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

Em nosso cotidiano, é notável a crescente demanda por conectividade. Conceitos tecnológicos emergentes como XaaS (*Anything as a Service*), IoT (*Internet of Things*), BYOD (*Bring your own device*), entre outras existentes e que ainda estão por vir, demandam melhoramentos e inovação na infraestrutura vigente. Com o avanço tecnológico, as redes de computadores representam um papel fundamental nas tecnologias emergentes. A internet do futuro passa a exigir cada vez mais dos recursos de rede, fato que impulsiona o avanço das tecnologias das Redes de Computadores.

Em ambientes acadêmicos, o uso destas tecnologias é frequente, onde mestres, alunos, colaboradores e visitantes buscam o conhecimento na rede, a qualquer hora e qualquer lugar. Desta forma, os departamentos de tecnologia da informação destas instituições possuem funções similares aos de provedores de internet: Garantir que diversos perfis de usuários, conectados por dispositivos heterogêneos que operam com dependência de conexão à internet, tenham uma boa experiência de uso da rede. Trata-se um desafio, pois são, em sua maioria, atendidas por enlaces com largura de banda insuficientes para a demanda.

Em princípio, considera-se que, para este desafio, a solução seria a sobreprovisão, que consiste em fornecer para a rede, capacidade suficiente de suportar qualquer tráfego que será jogado nela. Neste contexto, onde existe limitação de conectividade, remete-se aos administradores destes ambientes de redes, buscar atender a demanda: Fazer o melhor possível com o que se tem disponível.

Como se trata de um problema comum a inviabilidade/impossibilidade de sobreprovisão das redes, técnicas de QoS (*Quality of Service*) foram desenvolvidas concomitantemente com o avanço tecnológico dos últimos anos, permitindo que um melhor controle/aproveitamento fosse realizado sobre o tráfego de dados. Administrar uma rede de médio porte, sem o auxílio destas técnicas pode impactar diretamente na produtividade destas organizações (COMER, 2006).

A indústria, com foco no atendimento desta demanda, passou a incluir em seus equipamentos, principalmente roteadores e *switches*, recursos de QoS. A maioria dos equipamentos gerenciáveis do mercado saem de fábrica com este recurso incluso. Em contraponto, são soluções fechadas, e engessadas. As soluções comerciais que possuem como foco o QoS, como por exemplo equipamentos da *Palo Alto Networks*[1], que oferecem um controle e até um grau de flexibilidade maior de personalização mas, geralmente são de custo elevado.

[1] *A Palo Alto Networks* é uma empresa situada na califórnia e fornece um amplo conjunto de firewalls de próxima geração de nível empresarial, com uma gama diversificada de recursos de segurança para sua rede.

Comunidades de software livre com foco em gerenciamento de redes também avançaram no quesito do tratamento do QoS. Algumas soluções existentes tratam o QoS com bons níveis de eficiência. Porém, nos últimos anos, grandes avanços científicos/tecnológicos ocorreram nas redes de computadores. Dentre eles, destaca-se o SDN (*Software Defined Network*) que trazem um novo conceito para redes, que encontravam-se limitadas e como muitos autores citam, engessadas. Em resumo, agrega-se o conceito de liberdade ilimitada que impulsionou o avanço tecnológico dos sistemas de informação/sistemas operacionais livres, para as redes de computadores. Nela, separa-se o plano de dados do plano de controle, que é definido por software(JARSCHEL et al., 2014).

O impacto deste novo conceito de redes se deu de tal forma que a indústria demonstrou muito interesse a este novo paradigma. Empresas renomadas no mercado de redes e tecnologia como, CISCO[2] , HP[3] , IBM[4] , já estão investindo no desenvolvimento de controladores e comercializando soluções de redes baseadas em software.

Soluções livres de ferramentas existentes para a gestão de redes como, por exemplo, o pfSense[5] (muito utilizado pelos institutos federais) aderiram muito pouco ainda aos recursos de SDN. Aplicações livres ainda estão mais imergidas no âmbito acadêmico/experimental e pacotes completos como os comercializados pelas empresas anteriormente citadas, ainda estão em fase de desenvolvimento pelas as diversas comunidades que abordam SDN. Um dos benefícios da programabilidade da rede em SDN, é que com base em um ou mais recursos de informação, uma rede pode ser adaptada dinamicamente, para otimizar a utilização dos recursos (JARSCHEL et al., 2014).

A motivação para a pesquisa sobre este assunto, surgiu no âmbito do Instituto Federal de Rondônia - *Campus* Ariquemes, onde identificou-se a necessidade de melhoria nos controles da qualidade de serviço (devido à insuficiência de taxa de vazão contratada). Salienta-se que é um problema comum em outros *campi* do IFRO e outros da região norte que vivenciam a dificuldade de contratação de enlaces com taxas de vazão adequadas e custo dentro dos padrões de mercado.

Diante deste cenário, foi realizado um estudo sobre o comportamento do tráfego e identificados os fatores chave do processo. Este trabalho propõe e avalia uma nova estratégia para o gerenciamento de tráfego da rede e a proposta foi desenvolvida em um ambiente de emulação. Os resultados obtidos demonstram que através do

[2] Cisco Systems é uma companhia multinacional estadunidense sediada em San José, Califórnia, Estados Unidos da América

[3] A Hewlett-Packard Company é uma companhia de tecnologia da informação multinacional americana, até sua divisão, ocorrida em 2015

[4] International Business Machines é uma empresa dos Estados Unidos voltada para a área de informática.

[5] O pfSense é open source, licenciado sob BSD license, baseado no sistema operacional FreeBSD e adaptado para assumir o papel de um firewall e/ou roteador de redes.

monitoramento constante do tráfego, é possível ampliar a utilização do serviço de conexão à internet contratado, garantindo a de qualidade de serviço.

1.1 Objetivos

Propor um controle dinâmico por meio do monitoramento constante da taxa de transmissão, definido por *software*, com o intuito de propiciar a utilização eficaz dos recursos de rede, garantindo a priorização de serviços essenciais entre perfis de tráfego, limitando a vazão quando a capacidade de tráfego atinge seu limite e a liberação quando houver ociosidade.

1.1.1 Objetivos Específicos

- Realizar investigação da rede para identificação dos perfis de vazão de tráfego;
- Analisar as principais técnicas de QoS para definição da que mais se aproxima à necessidade personalizada exigida pelas políticas de tráfego definidas;
- Criar política de QoS, baseada na identificação dos parâmetros definidos pela investigação da rede;
- Implementar ambiente experimental da proposta, definido por software e recursos de virtualização;
- Desenvolver proposta de QoS que seja o mais adaptável possível ao ambiente de rede investigado, com finalidade de priorizar tráfego para perfis prioritários ao ponto que se evite ao máximo o desperdício de link de internet;
- Verificar se a proposta de priorização de tráfego baseada em serviços diferenciados produz melhora no compartilhamento da taxa de transmissão entre perfis de tráfego pré-definidos pela investigação;
- Criar interface gráfica para Administradores.

1.2 Organização da Dissertação

O presente documento está organizado de forma que no Capítulo 2, é apresentada fundamentação teórica envolvendo estudos e pesquisas mais recentes sobre redes definidas por *software*, qualidade de serviço, gerenciamento de tráfego, priorização e controle de banda. No Capítulo 3, é apresentado o estudo realizado no ambiente da pesquisa, onde se apresentam dados sobre o comportamento do tráfego existentes e a identificação das características a serem implementadas na nova estratégia de compartilhamento/controle do serviço de conexão à internet contratado. No

Capítulo 4, apresenta-se estratégia proposta por meio de uma nova topologia a ser adotada por meio da inserção de recursos virtuais e definidos por *software*. O Capítulo 5 apresenta a avaliação dos resultados obtidos. A conclusão e os trabalhos futuros são apresentados no Capítulo 6.

2 REFERENCIAL TEÓRICO

Neste capítulo realiza-se uma abordagem conceitual sobre recentes pesquisas e trabalhos relacionados. Posteriormente, realiza-se uma análise crítica sobre o assunto e discussão sobre trabalhos relacionados.

Este capítulo apresenta pesquisas e trabalhos relacionados a Qualidade de Serviço e Redes definidas por Software. A seção 2.1 apresenta conceitos sobre redes definidas por *software* e características da especificação 1.3 do protocolo *Openflow*. A seção 2.2 apresenta conceitos sobre os controladores SDN e características dos principais existentes. Conceitos sobre qualidade de serviço, e trabalhos mais recentes relacionados ao assunto são tratados na seção 2.3.

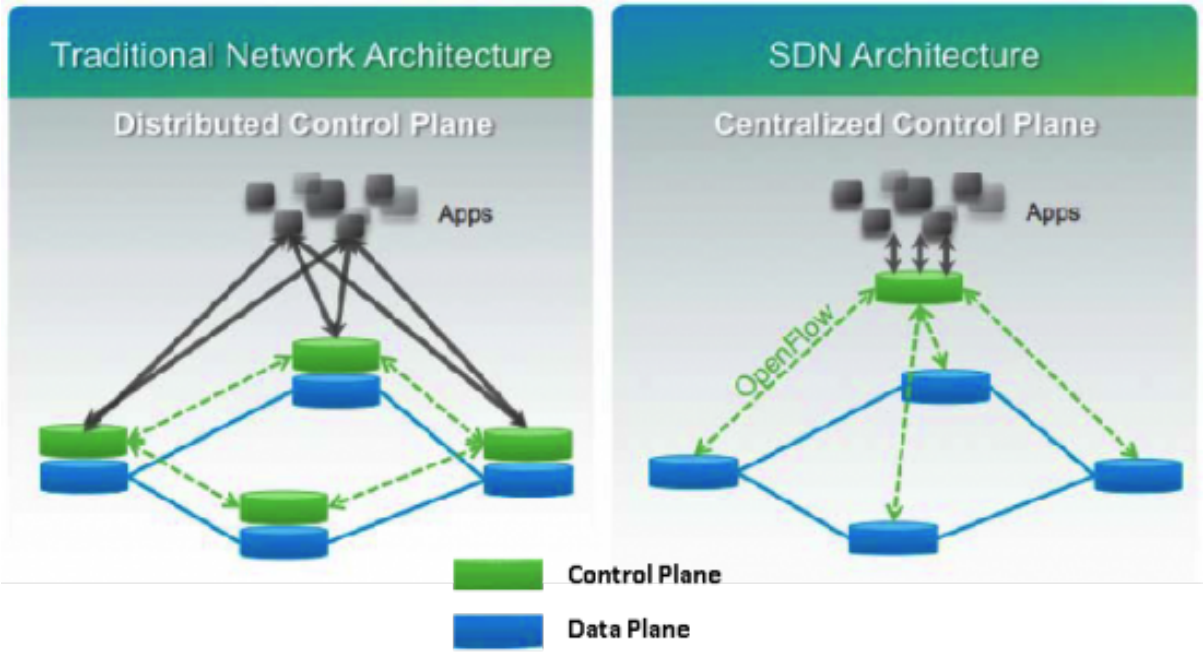
2.1 Redes definidas por *software* (SDN)

As redes definidas por software (SDN) representam um novo paradigma para as redes de computadores. É uma arquitetura emergente dinâmica, gerenciável, econômica e adaptável, tornando-a ideal para a alta largura de banda e a natureza dinâmica das aplicações atuais (OPEN NETWORKING FOUNDATION, 2017e). A Open Networking Foundation (ONF) é um consórcio liderado por operadores de redes sem fins lucrativos que impulsiona a transformação da infra-estrutura de rede e dos modelos comerciais de operadoras.

Um elemento chave no SDN é a introdução de uma abstração entre os fluxos (tradicionais) de emcaminhamento e controle para separar eles e fornecer aplicativos com os meios necessários para controle da rede. O objetivo é aproveitar esta separação e a programação associada, a fim de reduzir complexidade e permitir uma inovação mais rápida nas redes de computadores (IETL - RFC 7426, 2015).

A figura 1 ilustra a diferenciação existente entre a arquitetura das redes tradicionais e SDN:

Figura 1 – Diferença entre redes tradicionais e SDN



http://www.decom.ufop.br/imobilis/redes-definidas-por-software-software-defined-networks-sdn/

DAS et al. (2015) refere-se a SDN, como um novo paradigma de arquitetura, que separa o Plano de Dados do Plano de Controle, facilitando a programabilidade das funções de controle de uma rede. Desta forma, espera-se de SDN a redução do custo de operação pois possibilita simplificar operações, otimizar o uso de recursos por meio de dados ou algoritmos centralizados, além de simplificar a atualização do software controlador de rede.

Segundo informações da comunidade ONF (OPEN NETWORKING FOUNDATION, 2017d) possui as seguintes características:

- **Diretamente programável:** Controle de rede é desacoplado das funções de encaminhamento.
- **Ágil:** Permite ajustes de fluxo dinamicamente.
- **Controle centralizado:** A inteligência da rede é logicamente centralizada em controladores baseados em Software, que mantêm uma visão global do todo.
- **Automatizada:** Administradores de redes podem configurar, gerenciar, proteger e otimizar os recursos de rede muito rapidamente, via programas SDN automatizados sem depender de software proprietário.
- **Baseada em Padrões abertos:** Instruções são fornecidas pelos controladores de SDN em vez de vários dispositivos, específicos do fornecedor.

Segundo Gelberger, Yemini e Giladi (2013) , SND já existe ha aproximadamente 20 anos, porém apenas recentemente pode ser validada, mediante o surgimento do protocolo OpenFlow. Este validou SDN, transformando a abordagem convencional das redes de computadores em design de rede. Gelberger levanta também uma das desvantagens de SDN, ao afirmar que a adição de flexibilidade e funcionalidade, exige uma maior capacidade de processamento do equipamento, o que pode ocasionar perda de desempenho e rendimento.

Um dos benefícios da programabilidade da rede em SDN, é que com base em um ou mais recursos de informação, uma rede pode ser adaptada dinamicamente, para otimizar a utilização dos recursos. O Google, por exemplo, utiliza este mecanismo para otimizar o uso de largura de banda na ligação entre os Datacenters da empresa. Isto é possível pois ele aproveita as informações das fontes de tráfego e agrupa tráfego de aplicativos em grupos de fluxos com prioridades diferentes (JARSCHEL et al., 2014) .

Segundo Kumar (2013), aborda-se a possibilidade de provedores de acesso a internet, com o intuito de fornecer uma melhor qualidade de experiência do usuário, através do uso de SDN, oferecer aos seus clientes mecanismos de gerenciamento de sua própria conexão, possibilitando através de um aplicativo, baseado em três conjuntos de aplicações: Vídeo Streaming, Grandes *Downloads* e Navegação *Web*.

As redes definidas por *software* tiveram sua ampliação motivada pela criação do protocolo OpenFlow, que será explicado a seguir.

2.1.1 OpenFlow

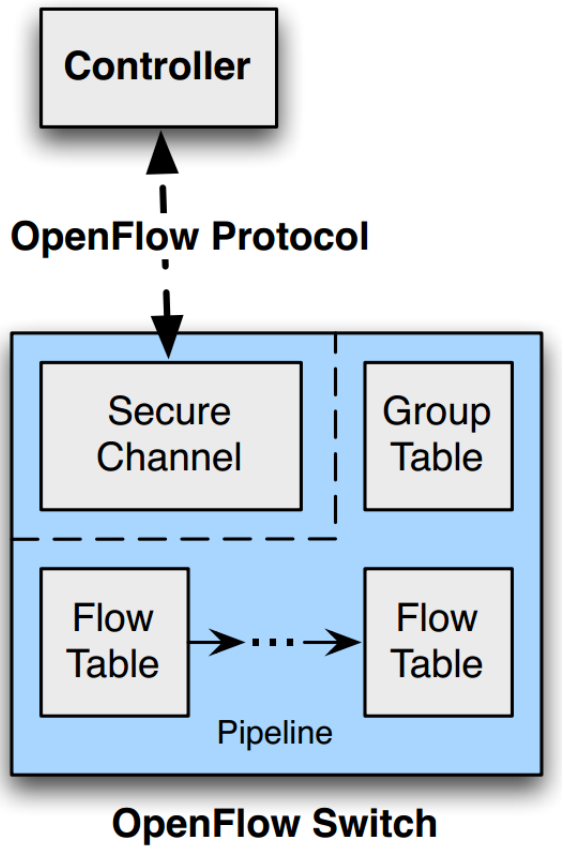
OpenFlow foi originalmente criado na Universidade de Stanford e atualmente é mantido pela ONF. Passou por muitas revisões e atualmente encontra-se na versão 1.5. A versão 1.6 da especificação já está sendo desenvolvida todavia o acesso ainda é permitido somente pelos membros da organização (OPEN NETWORKING FOUNDATION, 2017d).

Segundo LARA, KOLASANI e RAMAMURTHY (2013), o protocolo OpenFlow surgiu com o objetivo de padronizar a comunicação entre os componentes (*switches*, roteadores e controladores) da arquitetura SDN. Um dos fatores que impulsionaram as SDN e OpenFlow, foi a dificuldade para a comunidade de pesquisa em redes de computadores, testar novas ideias no *hardware* convencional (*switches*, roteadores convencionais). Isso ocorre porque o código-fonte do *software* destes *hardwares* são fechados e, por isso, costuma-se afirmar que as redes de computadores com equipamentos convencionais encontram-se “engessadas”. Afirmam também que o OpenFlow tinha como objetivo, fornecer uma plataforma que permitiria aos pesquisadores, executar experiências sobre redes em ambientes de produção.

Em suma, OpenFlow define o protocolo utilizado para estalecer a comunicação entre o controlador e o *switch*, a fim de distinguir entre solicitações SDN destinadas a ele próprio e outros pacotes que o controlador está enviando como por exemplo, os pacotes enviados para outros controladores (COMER, 2015) .

A figura 2 ilustra os componentes de um *switch OpenFlow*:

Figura 2 – Componentes principais de um *switch OpenFlow*



Fonte: https://www.opennetworking.org/software-defined-standards/specifications/

Nesta arquitetura, o controlador centraliza logicamente o controle da rede. A comunicação entre o controlador e os *switches* utiliza o protocolo *OpenFlow*. Esta comunicação pode ser realizada através de um canal seguro (SSL) ou não. Desta forma, estes *switches,* sendo virtuais ou não, para realizarem os encaminhamentos de acordo com o *software* executado por um controlador, necessita possuir compatibilidade com *OpenFlow*, de acordo com a versão do protocolo.

As principais empresas fabricantes de equipamentos de redes já possuem linhas de produtos compatíveis com *OpenFlow*. Nestes equipamentos, a partir do momento em que é definido em sua configuração que o gerenciamento será remoto, ou seja, por um controlador, seu *software* nativo não realiza mais nenhuma ação de encaminhamento. Ao observarmos esta situação, toda as funcionalidades oferecidas

pelo seu *firmware*[1] , ficam desativadas, funcionando apenas o sistema que gerencia a camada de comunicação com o controlador e o *hardware*. Além disso, alguns destes equipamentos podem funcionar de modo híbrido onde configura-se que determinadas portas são controladas pelo controlador SDN e outras, pelo *firmware.*

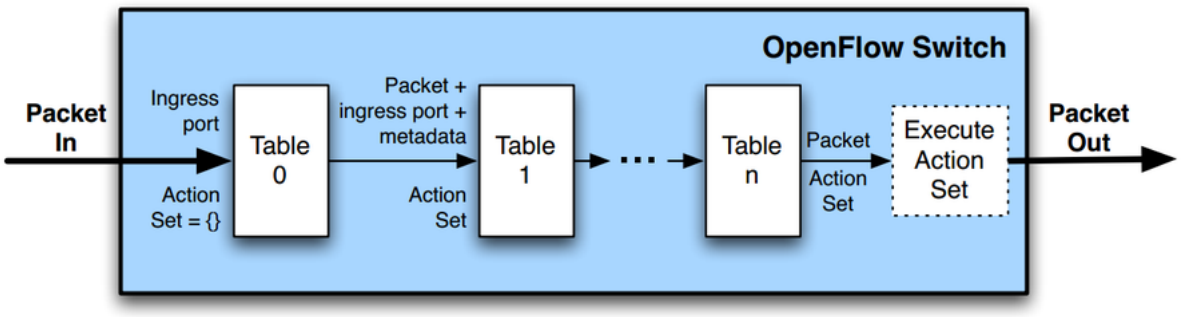
Neste caso, estima-se que estas empresas poderão lançar novos conceitos de fabricação e comercialização de equipamentos, separando a linha de produtos de redes em *hardware* e *software*. Poderão oferecer opções como equipamentos tradicionais (*firmware* acoplado), soluções híbridas, (*firmware* acoplado e compatível com *OpenFlow*), *hardware* puro (equipamento sem *software*, com dependência total de um controlador) e ainda comercializar também apenas a licença de uso de sua versão de *software* controlador. Um exemplo deste caso, é a HP que já comercializa licenças de aplicações SDN em parceria com a Arubanetworks (AIRHEADS COMMUNITY, 2017) .

Além disso, estima-se que com estas novas possibilidades proporcionadas pela SDN/Openflow, o custo de implantação de redes seja reduzido, mantendo o mesmo ou maior desempenho das redes tradicionais.

2.1.1.1 Especificação *OpenFLow*

Cada *switch* compatível com *OpenFlow* mantém uma ou mais tabelas de fluxo, que são usadas para pesquisas e encaminhamentos de pacotes. Uma tabela de grupo e um canal *OpenFlow* para controladores externos também fazem parte da especificação do *switch* (RFC 7426, 2015) . A figura 3 ilustra a forma como o pacote é inserido em uma tabela de fluxo, na entrada de um *switch* OpenFlow:

Figura 3 – Verificação de combinação (match) em tabela de fluxos



Fonte: https://www.opennetworking.org/software-defined-standards/specifications/

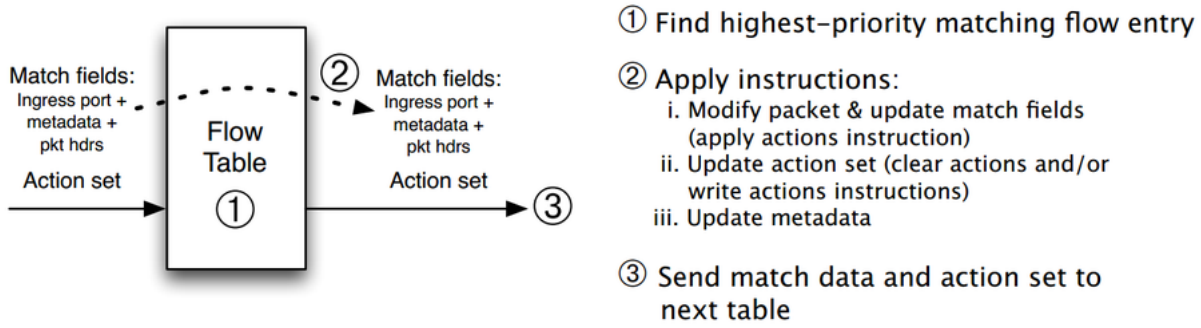
Ao chegar na porta de entrada de um *switch OpenFlow,* verifica se o pacote combina (*match)* com uma ou mais tabelas de fluxo registradas. Estas tabelas são

[1] Sistema operacional programado diretamente no hardware de um equipamento eletrônico.

constituídas por fluxos de entrada. Sequencialmente numeradas, começam em 0 e, para cada novo fluxo, a verificação de combinação inicia por ele. Os principais componentes de um fluxo envolvem campos de: match, prioridade, contadores, instruções, tempo limite e *cookie* (OPEN NETWORKING FOUNDATION, 2017a).

Após encontrado o caminho a seguir na tabela, modifica-se o pacote, atualizando o campo de *match*, a ação sobre ele e atualização dos metadados. Carregando estas atualizações, os pacotes são encaminhados para a próxima tabela e assim, sucessivamente. A figura 4 , exemplifica este processo.

Figura 4 – Processamento de fluxos de entrada

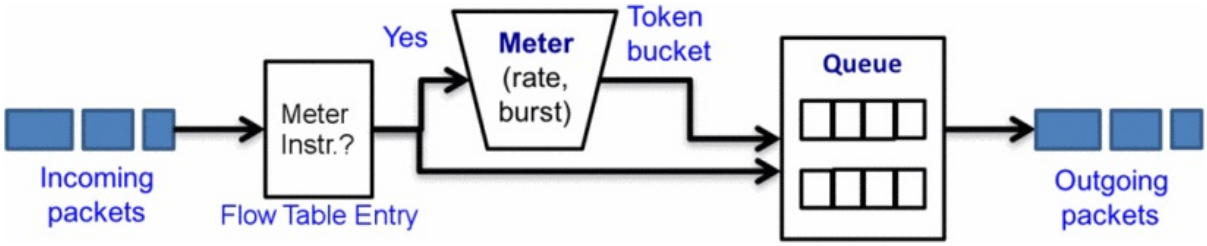


https://www.opennetworking.org/software-defined-standards/specifications/

2.1.2 Medidores

Um dos componentes observados na especificação do *OpenFlow* 1.3 são os medidores. Estes, possibilitam o *OpenFlow* a implementar técnicas de QoS, desde as mais simples, até as mais complexas.

Figura 5 – Linha de processamento de pacotes



(BOLEY; JUNG; KETTIMUTHU, 2017)

A figura 5 ilustra o fluxo do processo de medição na especificação 1.3 do *OpenFlow*. Os pacotes recebidos são encaminhados para um medidor conforme instruído por uma entrada da tabela de fluxo correspondente. O *switch* procura o medidor correspondente na tabela do medidor do *switch*, que limpa e passa ou descarta pacotes associados pela taxa de *token bucket's* e pelo tamanho do *burst* (BOLEY; JUNG; KETTIMUTHU, 2017).

Conforme ilustra a figura 5 , ao contrário das filas (que realizam o controle baseado por portas do *switch*) os medidores são conectados diretamente às entradas de fluxo. Qualquer entrada de fluxo pode especificar um medidor em seu conjunto de instruções. O medidor mede e controla a taxa agregada de todas as entradas de fluxo às quais ele está anexado. Pode-se dizer que os medidores atuam em uma etapa anterior ao processo de enfileiramento, podendo este ultilizar estas informações de medição para manipular os fluxos.

De acordo com a especificação do *OpenFlow* 1.3, os principais componentes de um medidor são:

- **identificador do medidor:** um inteiro exclusido de 32 bits
- **banda do medidor:** uma lista de bandas de medição onde cada faixa de medição possui uma taxa específica e maneira de processar o pacote
- **contadores:** atualiza a contagem quando pacotes são processados por um medidor

Cada banda de um medidor é identificada por uma taxa e formada pelos componentes: Tipo da banda, taxa, contador e tipo de argumentos específicos.

O tipo da banda define como o pacote será processado, a taxa é usada para selecionar a banda do medidor e os contadores são atualizados quando os pacotes são processados. Cada tipo de banda pode possuir um tipo de argumento específico opcional, contendo a opção de descartar o pacote baseado na taxa de limitação de banda como também, redefinir o rótulo de código de serviços diferenciados (DSCP) do cabeçalho IP, podendo ser usado para policiamento *DiffServ* (OPEN NETWORKING FOUNDATION, 2017a).

2.2 Controladores

O termo controlador está associado diretamente ao papel que este componente representa em SDN. Nesta seção, serão apresentadas as características de diferentes tipos de controladores existentes hoje em dia. Ressalta-se que as comunidades desenvolvedoras destes controladores também os nomeiam como Sistemas Operacionais de Redes (NOS).

A arquitetura SDN depende muito da eficiência do plano de controle, onde o controlador SDN reside. Para tornar o plano de controle mais eficiente, é muito importante ter um controlador que não só leve menos tempo para fazer a tomada de decisões para o plano de encaminhamento, mas também lida com os modernos desafios de tráfego de rede. O avanço da arquitetura SDN deu origem a vários controladores SDN e, com

tantos controladores existentes atualmente, muitas vezes torna-se uma tarefa difícil escolher com qual controlador trabalhar (RASTOGI; BAIS, 2016).

A seguir, serão apresentados algumas características de uns dos principais controladores existentes, mantidos por comunidades de *software* livre.

2.2.1 ONOS

O ONOS é um controlador mantido pela ONF. Segundo informações da comunidade OPEN NETWORKING FOUNDATION (2017b) , este controlador oferece escalabilidade, alto desempenho, alta disponibilidade e está preparado para soluções de SDN/NFV de próxima geração.

O ONOS oferece abstrações norteáveis inovadoras que permitem a criação, implantação e operação de aplicativos de configuração, gerenciamento e controle. A visão global da rede e a estrutura da intenção do aplicativo são dois exemplos. As aplicações podem ser adicionadas para executar “on-box” usando interfaces nativas, ou “off-box” usando as interfaces de transferência de estado representacional - REST e / ou gRPC[2] . O ONOS abstrai as características do dispositivo de modo que o sistema operacional principal não precisa estar ciente do protocolo particular que está sendo usado para controlar ou configurar um dispositivo (OPEN NETWORKING FOUNDATION, 2017c).

2.2.2 OpenDayLight

O OpenDaylight (ODL) é uma plataforma aberta modular para customizar e automatizar redes de qualquer tamanho e escala. Surgiu do movimento SDN, com um foco claro na programabilidade das redes de computadores. Foi projetado desde o início como uma base para soluções comerciais que abordam uma variedade de casos de uso em ambientes de rede de produção. Possui arquitetura de modelo dirigido, modular e multi protocolo (OPENDAYLIGHT.ORG, 2017).

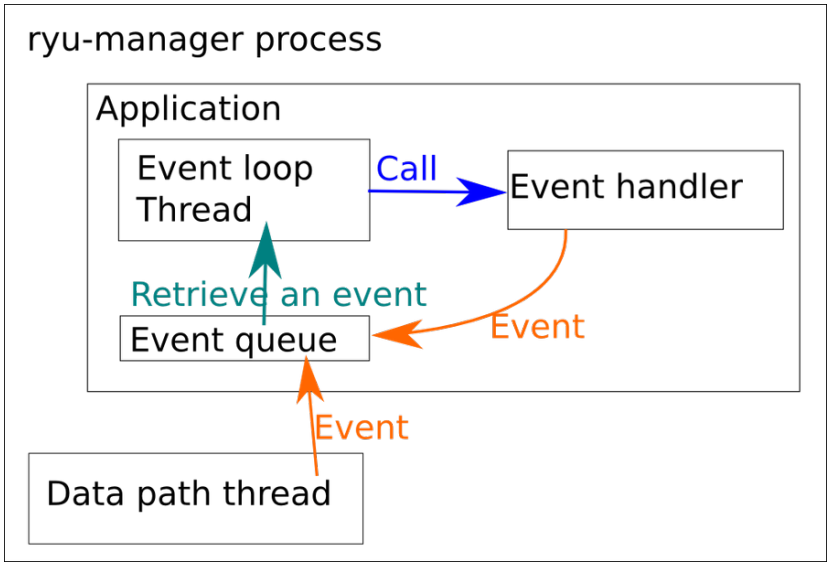
2.2.3 RYU - Sistema operacional de rede (NOS)

O Ryu é um controlador de rede definido por *software*, baseado em componentes. Escrito em *python*, possui uma interface de programação de aplicativos (API) bem definida facilitando o desenvolvimento de ferramentas de controle e gestão de redes decomputadores. Todo seu código está disponível livremente, sob a licença Apache 2.0 (RYU PROJECT TEAM, 2017).

[2] Tecnologia que permite que um aplicativo cliente possa chamar diretamente métodos em um aplicativo de servidor em uma máquina diferente como se fosse um objeto local, tornando mais fácil para você criar aplicativos e serviços distribuídos

Optou-se por este controlador por possuir sua documentação bem construída além de suportar vários protocolos de gerenciamento de dispositivos de rede, dentre eles, o OpenFlow. Suporta totalmente as versões 1.0, 1.2, 1.3, 1.4 e 1.5 (NIPPON TELEGRAPH CORPORATION, 2017) .

Figura 6 – Modelo de aplicações Ryu



https://osrg.github.io/ryu-book/en/html/arch.html#

A figura 6 ilustra o modelo de programação usado para aplicações Ryu. Neste modelo, a lógica do usuário é descrita como uma aplicação. As aplicações são classes que herdam da base ryu.app_manager. Os eventos são tratados como os objetos das classes e cada aplicação possui uma única fila para receber eventos.

Ryu oferece um visualizador de topologia básico que fornece uma ilustração gráfica da topologia da rede, do status do link e das entradas de fluxo. Nota-se que ainda não é muito usual esta interface (SCOTT-HAYWARD, 2015).

RASTOGI e BAIS (2016), realizaram um estudo comparativo entre dois controladores desenvolvidos em Python, o POX e Ryu ultilizando como métricas, taxa de bits, atraso médio, taxa de pacotes e média de *jitter*. Obteve como resultados de sua pesquisa a verificação de pontos positivos e negativos de ambos como por exemplo, quado a mudança da camada 1 está em questão, o POX possui uma melhor capacidade de gerenciamento de tráfego (alta taxa de bits, menos atraso e jitter). Quando trata de ações de manipulação de pacotes na camada 2, o Ryu é melhor no gerenciamento de cenários, roteando novos pacotes para seu destino, encaminhamento de fluxos e instruindo *switches* sobre como lidar com pacotes repetitivos na rede.

2.3 Mininet

Mininet é um emulador de redes OpenSource, sob licença BSD, que cria uma rede de hosts virtuais, *switches*, controladores e links. Os hosts Mininet executam um *software* de rede Linux padrão e seus *switches* suportam o OpenFlow para roteamento personalizado altamente flexível e SDN . O Mininet apoia a pesquisa, o desenvolvimento, a aprendizagem, a prototipagem, o teste, a depuração e quaisquer outras tarefas que possam se beneficiar de ter uma rede experimental completa em um laptop ou outro PC (MININET.ORG, 2017) .

O Mininet usa a virtualizaçõa baseada em processos para executar vários hosts e alternar em um único kernel do sistema operacional. Desde a versão 2.2.26, o Linux suporta *espaços de nomes de rede*, um recurso de virtualização leve que fornece processos individuais com interfaces de rede separadas, tabelas de roteamento e tabelas ARP.

Os designs criados virtualmente podem ser migrados de forma simples para o ambiente físico. Possibilita também a prototipagem rápida de redes definidas por *software*, teste de topologia complexa sem a necessidade de conectar a uma rede física, desenvolvimento colaborativo independente em uma mesma topologia.

É quase inteiramento escrito em Python. Fornece uma extensiva API Python para criação e experimentação de rede. É lançado sob uma licença permissiva BSD Open Source e é ativamente desenvolvido e suportado por comunidade de redes e entusiastas.

2.4 *Ofsoftswitch*

Esta é uma implementação do *software switch* do espaço de usuário compatível com OpenFlow 1.3. Lançado sob a licença BSD (BSD-like para o código do *switch* original de Stanford) o código baseia-se na implementação do *softswitch* Ericsson TrafficLab 1.1 (ERICSSON TRAFFIC LAB, 2012) , com alterações no plano de encaminhamento para suportar o OpenFlow 1.3 (CPQD, 2017b) .

No pacote de instalação da ferramenta estão incluídos os seguintes componentes:

- ofdatapath: a implementação do *switch*
- ofprotocol: canal seguro para conectar a chave ao controlador
- oflib: uma biblioteca para converter a partir formato do OpenFlow 1.1
- dpctl: uma ferramenta para configurar o *switch* a partir do console

2.5 QoS

O termo Qualidade de Serviço (*Quality of Service* - QoS) se refere às garantias de desempenho estatístico que um sistema de rede pode dar com relação as métricas de perda, retardo, vazão e *jitter* (COMER, 2006). Esta técnica surgiu a partir da necessidade de garantir uma qualidade mínima para conexões, como por exemplo, a transferência de voz(VoIP) e vídeo em tempo real, sobre IP (*Internet Protocol*). Métricas como retardo e perda podem impactar significativamente na qualidade de experiência do usuário, a ponto de inviabilizar sua utilização.

Comer (2006), cita que uma rede isócrona é projetada para atender a limites de desempenho estritos portanto, fornece a garantia de QoS. Já a rede de comutação de pacotes é baseada pelo tráfego de melhor esforço (BE) que não possuido controle de QoS. Cita ainda sobre uma controvérsia existente sob QoS onde engenheiros que projetaram os sistemas de telefonia insistem que a reprodução de voz de qualidade exige que a rede por onde trafega, forneça garantias de QoS quanto a retardo e perda para cada chamada telefônica. Por outro lado, os engenheiros que projetaram o IP, insistem que a internet funciona razoavelmente bem sem garantias de QoS e que acrescentar QoS por fluxo é inviável por que os roteadores tornarão o sistema caro e lento.

Os dois modelos convencionais de qualidade de serviço (QoS) são o modelo de serviço integrado (IntServ) e o modelo de serviço diferenciado (DiffServ), que correspondem ao controle QoS baseado em fluxo e ao controle QoS baseado em classe, respectivamente. Na prática, os modelos QoS são implementados por filas e medidores em *switches*. Na SDN, o primeiro rascunho oficial da especificação OpenFlow (1.0) forneceu filas básicas de classe de serviço que se anexam a uma porta específica. Essas filas, como as tradicionais, devem ser configuradas diretamente no *switch* e não podem ser administradas pelo controlador. A largura de banda não utilizada por qualquer classe de tráfego será dividida proporcionalmente no resto das filas com base na atribuição estática. O padrão OpenFlow foi posteriormente expandido (1.3) para incluir a noção de fluxos de medição (BOLEY; JUNG; KETTIMUTHU, 2017) .

2.5.1 Serviços Diferenciados (*DiffServ - DS*)

Serviços diferenciados foram desenvolvidos com o intuito de habilitar a classificação de serviço escalável na internet sem a necessidade de ser baseado por fluxo e a marcação ocorre em cada nó da rede. As regras deste serviço são implementadas de acordo com políticas administrativas de tráfego. Um nó, compatível com DS possui um classificador que seleciona pacotes com base no valor do campo DS, juntamente com o gerenciamento de buffer e mecanismos de agendamento de pacotes, indicado

pelo valor do campo DS (IETF - RFC 2474, 1998)

Segundo a IETF-RFC 4594 (2006) , as especificações de DiffServ fornecem mecanismos equivalente aos de serras de fita, plainas, perfuradoras e outras ferramentas. Nas mãos de um especialista, não há limite para o que pode ser construído, ou seja, dependo da criatividade e conhecimento deste recurso por um administrador de redes, soluções personalizadas podem ser desenvolvidas. É definida por classes de serviço como DSCPs de serviços diferenciados, condicionadores de trânsito, comportamentos Per-Hop (PHBs) e o active Queue Management (AQM). Estas definições de classe de serviço, baseiam-se nas diferentes características do tráfego (atraso, perda e jitter) e no desempenho exigido das aplicações/serviços.

Nesta especificação, os principais conceitos de serviços diferenciados envolvem:

- **enfileiramento** - utilizado quando a demanda de tráfego é maior que a capacidade de transmissão do meio. Subdividem-se em **enfileiramento prioritário** (um agendador inspeciona a fila de prioridade mais alta e se houver dados presentes, retira-os desta fila) e **taxa de enfileiramento** (combinação de um conjunto de filas e um agendador que as esvazia baseada em uma taxa especificada).
- **gerenciamento ativo de filas** - procedimentos que usam o lançamento ou marcação de pacotes para gerenciar a profundidade de uma fila.
- **acondicionamento de tráfego** - procedimentos de medição realizados no momento de chegada de um pacote ao roteador, podendo ser medido e descartado ou marcado de acordo com uma política ou configurado na entrada da rede.
- **ponto de código de serviços diferenciados** - um número no intervalo 0..63 que é colocado em um pacote IP para marcá-lo de acordo com a classe de tráfego que pertence sendo metade destes valores destinados a serviços padronizados e a outra metada está disponível para definição local.
- **comportamento Per-Hop** - combinação dos mecanismos acima para formar um conjunto específico de características para lidar com diferentes tipos de tráfego.

Dentro desta arquitetura de serviços diferenciados, ainda citando a especificação IETF-RFC 4594 (2006), três comportamentos de encaminhamento fundamentais foram definidos e caracterizados para uso geral descritos nas seções a seguir.

2.5.2 Encaminhamento Padrão (DF)

Define-se por encaminhamento padrão o comportamento básico de encaminhamento aplicado a qualquer classe de tráfego, associado ao serviço de melhor esforço(BE). Os pacotes em trânsido podem ser perdidos, reordenados, duplicados ou atrasados, de acordo com as cargas de tráfego.

No encaminhamento padrão, é fornecido um único valor DSCP para o serviço de melhor esforço para identificar o tráfego, uma fila para armazená-lo e aplicado um gerenciamento de filas ativo para proteger a rede e limitar atrasos.

2.5.3 Encaminhamento garantido (AF)

É um serviço de melhor esforço melhorado, considerando a elasticidade na origem do tráfego. O receptor detecta perda ou variação no atraso na rede e fornece informações ao remetente para ajustar sua taxa de transmissão para apriximar a capacidade disponível.

No AF, são fornecidos múltiplos valores DSCP para identificar o tráfego, uma fila para armazenar o agregado e gerenciamento ativo de filas para proteger a rede e limitar atrasos. O tráfego é medido no momento em que entra na rede e marcado variadamente de acordo com as taxas estabelecidas.

Os valores DSCP são divididos em três pools. O primeiro grupo de 32 pontos de código é atribuído ao IETF, dos quais 22 registraram significados bem conhecidos. A Tabela I especifica os pontos de código comummente usados e as classes herdadas de compatibilidade para trás com o campo ToS pintado pelos pontos de código do Seletor de Classe (CS) (VENNE, 2017) .

Figura 7 – Pontos de código mais comuns

Binary	Decimal	PHB class	Priority
000000	0	BE	Default, CS0
001000	8	CS1	Priority, Class Selector 1
001010	10	AF11	Low drop probability
001100	12	AF12	Medium drop probability
001110	14	AF13	High drop probability
010000	16	CS2	Immediate, Class Selector 2
010010	18	AF21	Low drop probability
010100	20	AF22	Medium drop probability
010110	22	AF23	High drop probability
011000	24	CS3	Flash, Class Selector 3
011010	26	AF31	Low drop probability
011100	28	AF32	Medium drop probability
011110	30	AF33	High drop probability
100000	32	CS4	Flash Override, Class Selector 4
100010	34	AF41	Low drop probability
100100	36	AF42	Medium drop probability
100110	38	AF43	High drop probability
101000	40	CS5	Critical/ECP, Class Selector 5
101100	44	VA	Voice Admit
101110	46	EF	Expedited Forwarding
110000	48	CS6	Internetwork Control, Class Selector 6
111000	56	CS7	Network control, Class Selector 7

Fonte: VENNE,2017

2.5.4 Encaminhamento acelerado (EF)

Sua finalidade é fornecer QoS para serviços de necessitem de baixa perda, baixo atraso e baixo jitter. Pode ser usado para aprimorar o tráfego de melhor esfoço onde, usando técnicas de enfileiramento, pode minimizar a probabilidade de atraso ou variação do atraso.

Por possuir estas características, geralmente é utilizado apra transporta voz e vídeo sob IP, que são altamente sensíveis as características de atraso e variação de atraso.

2.6 Trabalhos Relacionados

Nesta seção são apresentados os trabalhos e pesquisas mais recentes relacionadas ao assunto abordado que envolve QoS e SDN.

HU, WANG e DAI (2015) apresentam em sua pesquisa uma abordagem de arquitetura SDN sobre IP (SoIP), para fornecer uma melhor QoS para usuários finais e aplicativos. A ideia básica de SoIP é atualizar ou reconstruir a borda da rede e construir

Redes de sobreposição baseadas em SDN, enquanto o núcleo da rede mantém os serviços diferenciados existentes com base no campo de protocolo Tipo de serviço (ToS) do cabeçalho IP. A característica do SDN é a sua capacidade intrínseca de implementar a Gerenciamento por fluxo e controle centralizado na rede recursos e, portanto, a borda da rede pode realizar alocação de recursos e agendamento de acordo com os requisitos de QoS de cada fluxo. Nesse caso, a infraestrutura e os dispositivos físicos que usados na Internet podem garantir QoS e SoIP, podendo manter-se eficaz em diferentes escalas de redes.

Ainda pôde-se observar que os autores acima direcionam à implantação de QoS via manipulação do cabeçalho IP da borda da rede para a WAN. Visto que para este tipo de QoS funcionar em uma WAN, todos os hosts por onde passa o tráfego, da origem até o destino, precisam estar preparados para aplicar a priorização de acordo com o campo DSCP, o que torna-se um desafio, propondo-se o estabelecimento de conexões MPLS entre as bordas de LAN's permitindo a QoS. Assemelhando-se a presente proposta, no sentido de que criar-se um domínio *DiffServ* local para manipulação do ToS. A conexão de diferentes domínios DiffServ pela proposta do SoIP, possibilita a QoS neste meio, independente da heterogeneidade do núcleo da WAN por onde passa o tráfego.

Para HONG et al. (2013) a WAN que conecta os *datacenters* (DC) é uma infraestrutura crítica para provedores de serviços *on-line*, como Amazon® , Google® e Microsoft®. Muitos serviços dependem da comunicação intercontinental de baixa latência para uma boa experiência do usuário e em transferências de alto custo (por exemplo, ao replicar atualizações). Dada a necessidade de tráfego de alta capacidade inter-*datacenters* e, por possuírem características de tráfego únicas, a WAN entre *datacenters* muitas vezes uma rede dedicada, distinta da WAN que se conecta com ISPs para atingir usuários finais. É um recurso caro, com custo anual amortizado de 100s de milhões de dólares, pois fornece 100s de Gbps para Tbps de capacidade em longas distâncias. No entanto, os provedores não conseguem maximizar este investimento hoje.

Ainda sob análise de HONG et al. (2013) citam que as WAN inter-DC possuem uma eficiência extremamente fraca. A utilização média de até mesmo os links mais movimentados é de 40-60%. Um culpado é a falta de coordenação entre os serviços que utilizam a rede e garantem as prioridades de tráfego via estabelecimento de limites grosseiros e estáticos. Observa-se a mesma percepção na presente análise, que se trata de baixa produtividade de serviços de conexão contratados. Eles propõem o SWAN (WAN baseada por *software*) e, seus dois principais aspectos são: 1) a coordenação global de taxas de de envio, e 2) a coordenação de serviços e centralização do gerenciamento dos caminhos de rede. Por meio de provedores e corretores de alocação de recursos, que agem mediante rotinas de verificação, impõem-se a manipulação

DSCP para definição/re-definição da prioridade de tráfego.

A pesquisa de Boley, Jung e Kettimuthu (2017) apresentam um algoritmo de gerenciamento de fluxo de qualidade de serviço adaptável (AQoS) que aproveita a capacidade do controlador SDN para coletar estatísticas usando o protocolo OpenFlow. O algoritmo ajusta-se para otimizar os *throughputs* de QoS em tempo real, observando garantias de QoS por fluxo. Essa ideia assemelha-se a presente proposta no sentido de que leva-se em consideração a dinamização da manilupação dos fluxos usando os componentes de medidores do *OpenFlow* 1.3. Entretando, utiliza como referencia o tráfego de melhor esforço concorrente no meio de conexão enquanto propomos a criação de um domínio com QoS aplicado desde a entrada dos fluxos nele. Desta forma, os pacotes entrantes possuem seu ToS alterado para a classe 10 do DSCP, e só são alterados mediante a condição da medição do fluxo ser alcançada ficando a cargo da borda do domínio aplicar a priorização mediante os pacotes que chegam até ele.

No trabalho de WANG, ZHANG e CHEN (2017) propõem-se um mecanismo de gerenciamento de filas adaptável baseado em *feedback* e um algoritmo de mapeamento para alocar fluxos em diferentes subesquemas. Subesquemas com maior prioridade garante QoS em atraso e taxa de perda de pacotes. O fluxo por fluxo é mapeado para diferentes subesquemas em cada porta de saída ao longo do seu caminho de roteamento à luz de suas restrições de QoS e garantias de desempenho fornecidas por cada subcarga.

3 AMBIENTE DE REDE AVALIADO

No Instituto Federal de Rondônia, *campus* Ariquemes, utilizam-se duas ferramentas de gerenciamento de redes que são: *Router OS Mikrotik*[1] e *pfSense*[2] . O ambiente conta com dois *firewalls* independentes, um deles gerencia o ambiente de rede acadêmico, e outro gerencia o ambiente administrativo. A diferença desse gerenciamento entre os dois ambientes é definida na forma de controle de tráfego e do filtro de conteúdo.

O *RouterOS Mikrotik* atua como *gateway*[3] da rede, sendo a ferramenta principal de gerenciamento desse ambiente, ofertando serviços como: gerenciamento de tráfego, *Firewall*[4] , *Webproxy*[5], entre outros. No serviço de gerenciamento de tráfego, aloca-se uma largura de banda para uma VLAN de perfil acadêmico de 30 Mbps. Todo o tráfego excedente é bloqueado, entrando em uma fila de espera, de forma que o excedente seja descartado ou espere o tráfego diminuir para então seguir seu fluxo. Essa ação evita que a VLAN interfira na alocação de banda de outras VLAN's da rede acadêmica ou na rede administrativa, por exemplo. Em contrapartida, dependendo da quantidade de requisições de acesso e do tamanho da fila, os usuários poderão ter uma tempo de espera maior do que sua expectativa de uso da internet, mediante esta limitação.

Essas técnicas de limitação são necessárias no modelo atual de gerenciamento de tráfego no ambiente avaliado, pois, baseado nas atividades cotidianas de monitoramento e ajuste nos controles de filas, sabe-se que sem fazer uso de limitadores de vazão a rede acadêmica facilmente consumiria toda a capacidade do serviço de conexão à *internet* contratada, impactando gravemente na produtividade da Rede administrativa. Na seção 3.1.1 haverá maior detalhamanto sobre este gerenciamento, atualmente adotado.

Os recursos do *pfSense* não são muito diferentes do *Mikrotik,* possuindo a capacidade de limitação de taxas e priorização de tipos de protocolos que são personalizáveis, mas que também são estáticos, ou seja, uma vez criado e definidas as personalizações, elas não são alteradas até que haja uma nova definição de parâmetros. Atualmente, essa ferramenta provê apenas o serviço de conexão VPN[6] interligando o *campus* à Reitoria.

---

1 Sistema operacional de roteadores de propriedade da empresa Mikrotik

2 Sistema operacional FreeBSD adaptado para assumir o papel de um firewall e/ou roteador de redes.

3 Um Gateway, ou ponte de ligação, é uma máquina intermediária destinada, geralmente, a interligar redes, separar domínios de colisão, ou mesmo traduzir protocolos.

4 Um firewall é um dispositivo de uma rede de computadores que tem por objetivo aplicar uma política de segurança a um determinado ponto da rede.

5 Serviço de rede que desempenha a função de intermediar a conexão do computador (local) à rede externa (Internet)

6 Rede Virtual privada.

Além dos exemplos citados, diversas soluções, inclusive as ultilizadas nesse ambiente, possibilitam automatizar ações como habilitar/desabilitar regras de controle de fluxo, tratamento de filas e QoS, que podem ser programadas mediante análise temporal dos fluxo de tráfego desta rede. Porém, apesar de automatizadas, essas técnicas são estáticas e resolvem em parte os problemas do desperdício de largura de banda em uma rede onde o máximo de dinamicidade que se pode alcançar, é definir que em determinados horários uma regra diferente seja aplicada para esta limitação/liberação de banda. Além do mais, qualquer variação dos parâmetros planejados podem ser afetados por essas ações automatizadas, podenso-se citar por exemplo o tráfego de rajada.

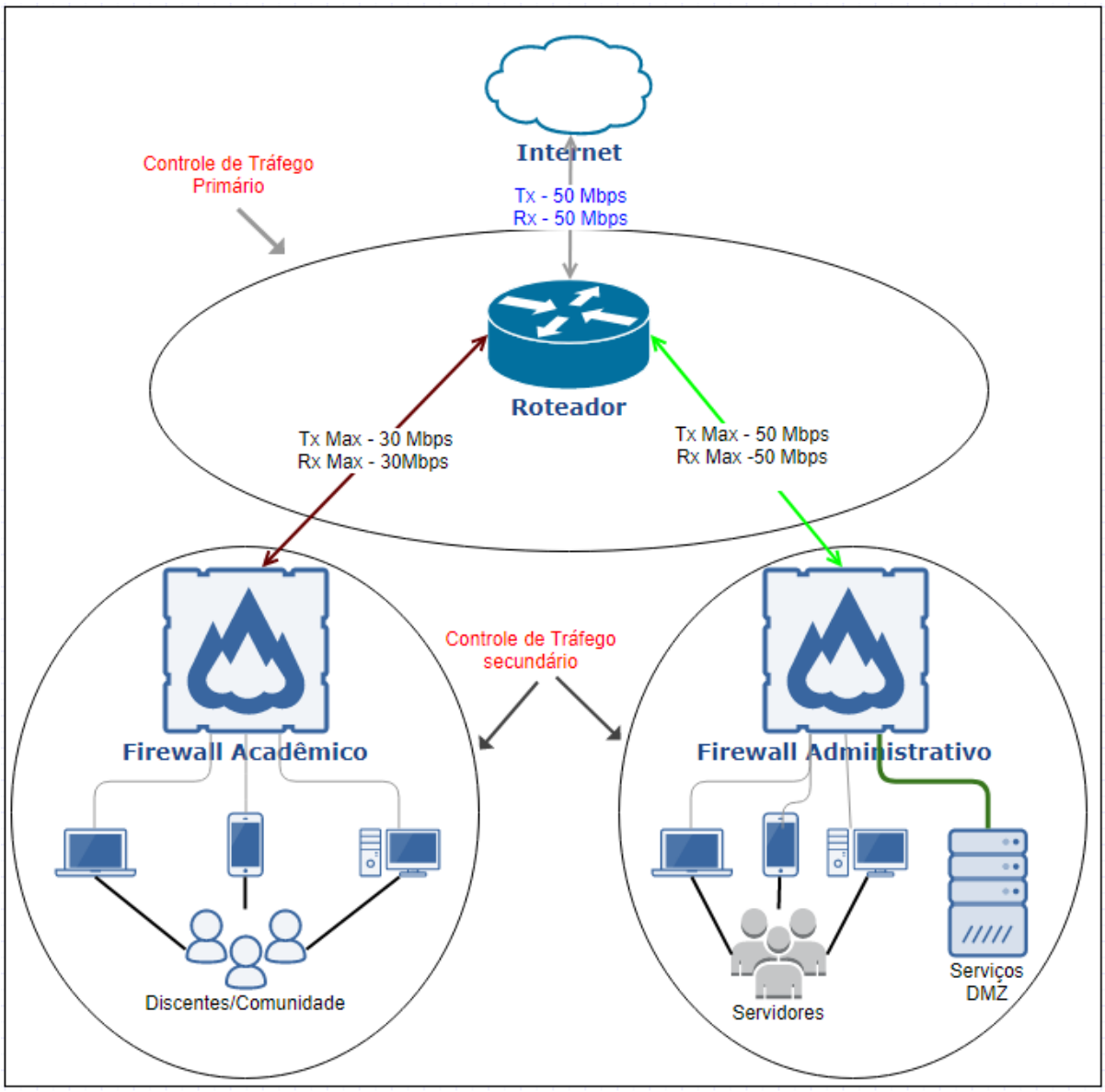
3.1 Investigação de tráfego

Nesta etapa realiza-se a investigação da rede objetivando a criação das políticas para os perfis de tráfego. Dados coletados mediante séries temporais são apresentados na seção a seguir.

3.1.1 Características do ambiente investigado

A topologia existente no ambiente investigado conta com uma estrutura de duas camadas de controle de tráfego: controle primário e controle secundário, conforme a Figura 8:

Figura 8 – Topologia atual - camadas de controle de tráfego



Elaborado pelo Autor

Conforme exemplifica a figura 8 o *campus* possui contratado o serviço de conexão à *internet* de 50Mbps, sendo o tráfego compartilhado entre os ambientes administrativo e acadêmico.

No controle primário identifica-se o roteador que é o responsável por compartilhar o tráfego de 50 Mbps entre os dois *firewalls*. Trata-te de uma *appliance* fornecida pelo provedor fabricada pela *Mikrotik®*, modelo RB-3011 e sistema operacional RouterOS, versão 6.35.4.

O controle da taxa de transmissão para a internet atualmente é estático, sendo configurada uma limitação de 30 Mbps para a rede acadêmica, já a rede administrativa não possui restrições, devido sua prioridade. Essa decisão está definida atualmente

pelo histórico de administração destes ambientes. A tabela 1 ilustra este controle:

Tabela 1 – Controle de taxas de tráfego - nível primário

Borda da Rede		
#	**Descrição**	**Download Max Limit**
0	Controle Geral	50Mbps
1	Rede Administrativa	50Mbps
2	Rede Acadêmica	30Mbps

O controle de tráfego secundário é realizado de forma independente pelos dois *firewalls*. Cada *firewall* também atua como *gateway*, utilizando a técnica de NAT, onde todo o tráfego da rede interna é mascarado, antes de sair para a internet. A configuração das filas da rede acadêmica está representada na tabela a seguir:

Tabela 2 – Controle de taxas de tráfego de nível secundário - Rede Acadêmica

Rede Acadêmica	
Sub-rede/vLAN	**Download Max Limit**
Eduroam - (Wireless)	10Mbps
Laboratório de Informática 1 (cabeado - 30 computadores)	10Mbps
Laboratório de Informática 2 (cabeado - 35 computadores)	10Mbps
Laboratório de Informática 3 (cabeado - 15 computadores)	10Mbps
Visitantes/Terceirizados	5Mbps

O controle de tráfego secundário da Rede Administrativa, está configurado da seguinte forma:

Tabela 3 – Controle de taxas de tráfego de nível secundário - Rede administrativa

Sub-rede / vLAN	Queues
Desktops Administrativos (cabeada/wireless)	15Mbps
Wireless para Funcionários	15Mbps
Datacenter	10Mbps

O controle de tráfego da rede administrativa fica desabilitado por padrão. Na situação de detecção de sobrecarga da rede o controle passa a ser habilitado manualmente pela equipe de TI.

A estrutura do controle de tráfego implantado atualmente no ambiente investigado tem se mostrado funcional e, de certa forma, provê algum compartilhamento entre os perfis de tráfego. Em contrapartida, percebe-se a existência de alguns problemas dessa metodologia adotada atualmente, que são:

- Controle de filas são rigorosos e inflexíveis, causando lentidão a quem está submetido a ele;
- Tráfego de rajada na rede administrativa impacta como um todo a rede, principalmente quando os controles de tráfego estão desabilitados;
- O ajuste das taxas de vazão dependem de constante monitoramento pela equipe de TI, o que é realizada de acordo com a demanda de atividades ou quando os usuários abrem chamados de atendimento reclamando de lentidão no acesso.

Para realização do monitoramento de tráfego destes ambientes usam-se dois sistemas, o Zabbix[7] e o TheDude[8], que coletam as estatísticas de trafego das interfaces de rede dos ativos, via protocolo SNMP, e apresentam seus resultados por meio gráficos, podendo-se selecionar um intervalo de tempo para sua análise.

Os problemas acima relacionados dão origem a um outro maior, que é objeto da presente pesquisa, qual seja o desperdício do serviço de conexão à internet.

3.1.2 Estudo de Caso

Todos os ativos de rede do *campus* são monitorados e utilizam por padrão o protocolo de gerenciamento simples de rede (SNMP). As coletas SNMP, como por

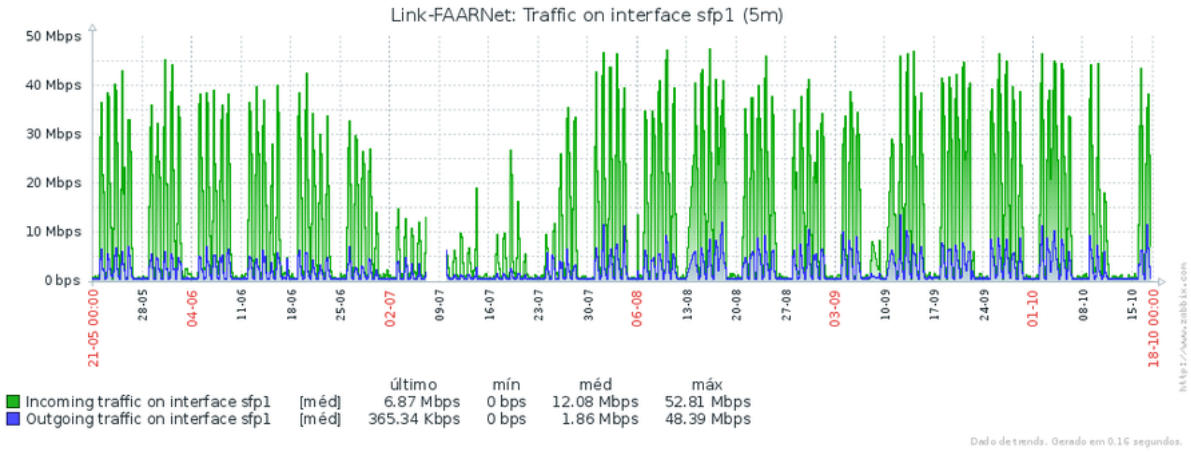
[7] Zabbix é um software que monitora diversos parâmetros de uma rede como a integridade e desempenho dos servidores.

[8] O The Dude é um software desenvolvido pela empresa Mikrotik de monitoramento de ativos via SNMP através de mapas.

exemplo, tráfego de interfaces, processamento, armazenamento de todos os ativos, são enviadas para um servidor de rede dedicado para armazenamento e tratamento desses dados pelo *Software Zabbix* , onde os dados coletados são tratados, possibilitando análises estatísticas temporais.

Baseado neste sistema de monitoramento, apresentamos as estatísticas do tráfego gerado por esta topologia. A figura 10 ilustra o consumo de taxa de transmissão pelo roteador de borda, a partir do dia 21/05/2017 até o 18/10/2017 (5 meses):

Figura 9 – Estatísticas de tráfego - vazão do enlace agregado

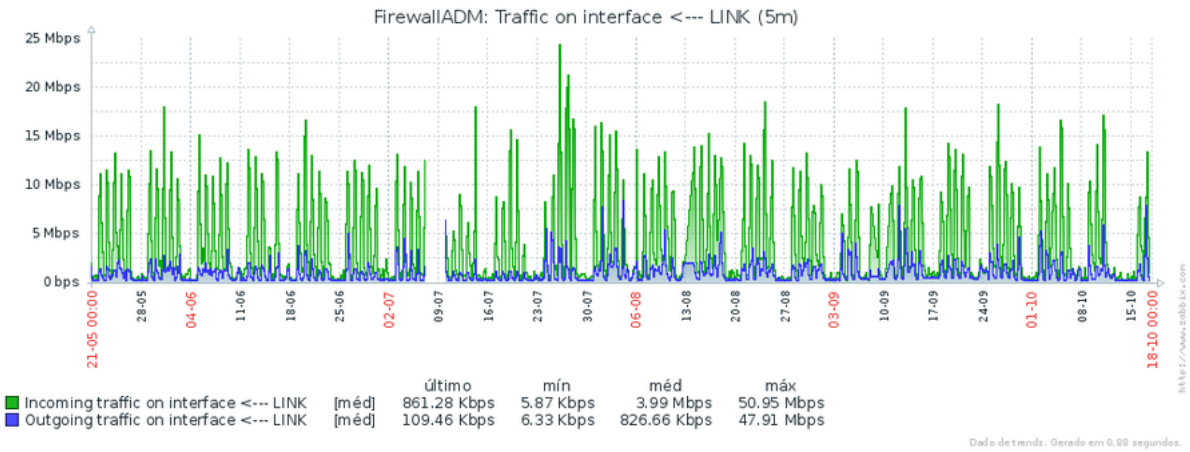


Elaborado pelo autor

Esta estatística de tráfego representa o consumo do tráfego primário, acumulando nos dois ambientes administrativo e acadêmico. Podemos observar que em praticamente todos os dias letivos, o tráfego atingiu seu limite de vazão (50 Mbps). Além disso, extrai-se como informação a média de tráfego agregado do período totalizando 12,03 Mbps.

A figura 10 apresenta as estatísticas do tráfego gerado pela rede administrativa coletadas no mesmo intervalo de tempo do tráfego agregado.

Figura 10 – Estatísticas de tráfego - vazão do enlace administrativo

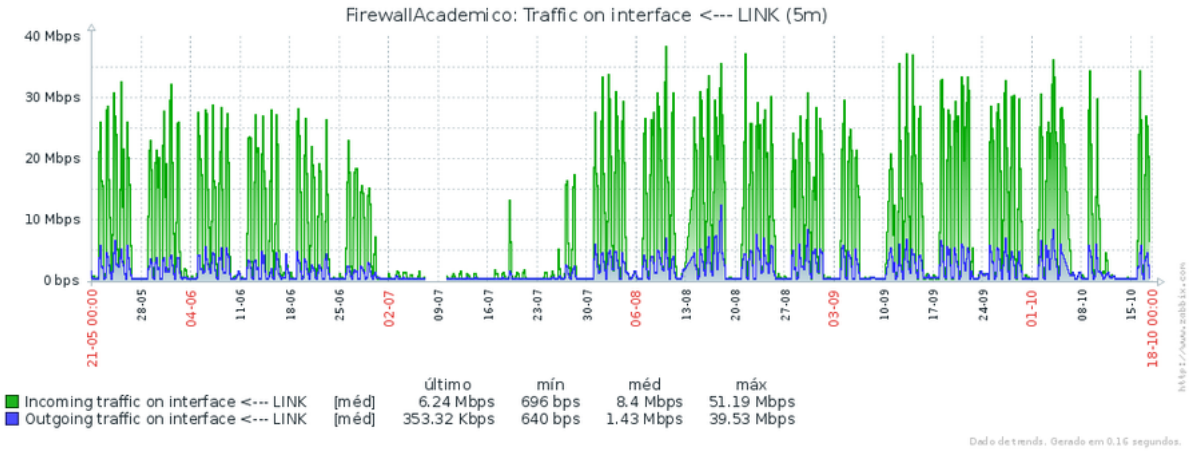


Elaborado pelo autor

Pode-se observar que o tráfego gerado ficou entre 10 e 15 Mbps em dias letivos e com média geral de 3,99 Mbps, representando uma média aproximada de 33% do tráfego total.

Dando sequência a esse levantamento, extrairam-se estatísticas de tráfego gerado pela rede acadêmica, apresentado na figura 11:

Figura 11 – Estatísticas de tráfego - vazão do enlace acadêmico



Elaborado pelo autor

Nesse caso nota-se que em dias letivos o consumo proveniente do ambiente acadêmico ficou em torno de 25 Mbps a 35 Mbps, e com média de 8,33 Mbps, representando um total aproximado de de 67 % do tráfego. A tabela 4 resume as médias gerais (dias letivos e não letivos) das estatísticas coletadas.

Tabela 4 – Resumo das estatísticas de média de tráfego

Perfil de Tráfego	Média
Tráfego Agregado	12,03 Mbps
Acadêmico	8,33 Mbps
Administrativo	3,99 Mbps

Elaborado pelo autor

Com o intuito de se obter uma média detalhada sobre o aproveitamento da taxa de vazão da conexão contratada, fez-se necessário coletar dados somente dos dias letivos. Para isso, foram selecionados os dias letivos dos meses de agosto, setembro e outubro de 2017. A tabela representada pela figura 12, reúne as estatísticas de tráfego geral e o percentual de consumo em relação ao serviço de conexão à internet contratado do mês de Agosto de 2017:

Figura 12 – Estatísticas de Tráfego agregado - Agosto

Agosto		
Dia	Média Mbps/Dia	Percentual de consumo
01/ago	35,16	70,32
02/ago	35,28	70,56
03/ago	36,26	72,52
04/ago	28,43	56,86
07/ago	27,79	55,58
08/ago	34,24	68,48
09/ago	36,08	72,16
10/ago	36,63	73,26
11/ago	28,56	57,12
14/ago	37,45	74,9
15/ago	36,2	72,4
16/ago	36,69	73,38
17/ago	35,09	70,18
18/ago	34,93	69,86
21/ago	27,86	55,72
22/ago	33,81	67,62
23/ago	37,17	74,34
24/ago	37,58	75,16
25/ago	27,25	54,5
28/ago	28,65	57,3
29/ago	30,07	60,14
30/ago	33,76	67,52
31/ago	23,43	46,86
	758,37	
Dias Letivos Total	23	
Média Mbps/Dia	32,97	
% de consumo	65,95	

A figura 13 exibe a estatística dos dias letivos do mês de Setembro de 2017:

Figura 13 – Estatísticas de Tráfego agregado - Setembro

Setembro		
Dia	**Média Mbps/Dia**	**Percentual de consumo**
01/set	27,19	54,38
04/set	29,18	58,36
05/set	34,95	69,9
06/set	30,71	61,42
11/set	20,02	40,04
12/set	33,74	67,48
13/set	39,06	78,12
14/set	36,07	72,14
15/set	31,02	62,04
18/set	37,35	74,7
19/set	37,2	74,4
20/set	38,17	76,34
21/set	35,01	70,02
25/set	36,74	73,48
26/set	41,89	83,78
27/set	38,04	76,08
28/set	35,7	71,4
29/set	29,57	59,14
	611,61	
Dias Letivos Total	**18**	
Média Mbps/Dia	**33,98**	
% de consumo	**67,96**	

Por fim, estatísticas de tráfego agregado, nos dias letivos do mês de Outubro de 2017 (Figura14):

Figura 14 – Estatísticas de Tráfego agregado - dias letivos de Outubro

Outubro		
Dia	**Média Mbps/Dia**	**Percentual de consumo**
02/out	38,91	77,82
03/out	35,14	70,28
04/out	37,28	74,56
05/out	36,86	73,72
06/out	24,05	48,1
09/out	34,48	68,96
10/out	29,28	58,56
11/out	13,25	26,5
16/out	34,9	69,8
17/out	32,67	65,34
18/out	37,28	74,56
19/out	30,91	61,82
20/out	29,62	59,24
23/out	32,53	65,06
24/out	33,14	66,28
25/out	36,43	72,86
26/out	34,21	68,42
27/out	31,14	62,28
30/out	36,93	73,86
31/out	38,16	76,32
Táfego Total	657,17	
Dias Letivos Total	**20**	
Média Mbps/Dia	**32,86**	
% de consumo	**65,72**	

Ao analisar as estatísticas de tráfego agregado, observa-se que a variação do consumo é mínima em dias letivos, e a média geral é apresentada no quadro a seguir:

Tabela 5 – Média geral de uso da conexão contratada.

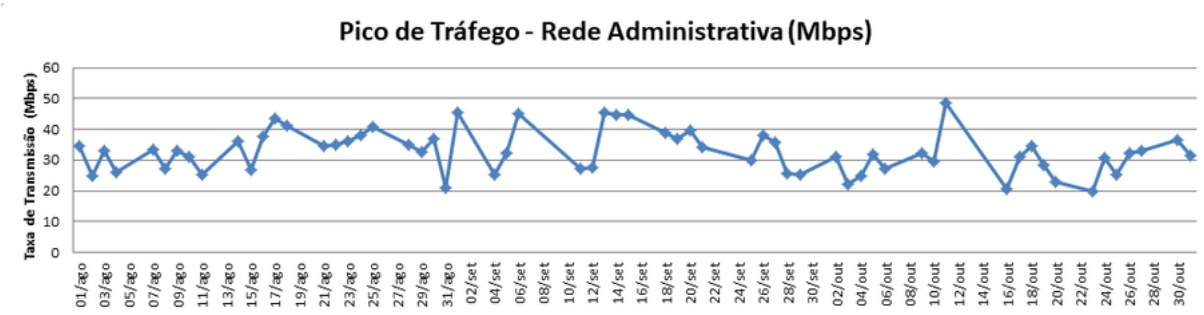
Média de consumo dos três meses	Média do percentual de Uso
33,27 Mbps	66,53 %

3.1.3 Pico de tráfego

Além da análise anteriormente realizada, existe a questão da política de priorização de tráfego e, organizacionalmente, é definido que a rede administrativa possui prioridade sobre qualquer necessidade de conexão à rede acadêmica. Salienta-se que essa política é definida pela importância dada ao acesso aos sistemas institucionais, aos sistemas do governo, aos sistemas acadêmicos, aos ambientes virtuais de aprendizagem etc.

Mediante à política institucional buscou-se entender o comportamento do tráfego proveniente do perfil administrativo, seu resultado está demonstrado na figura 16 :

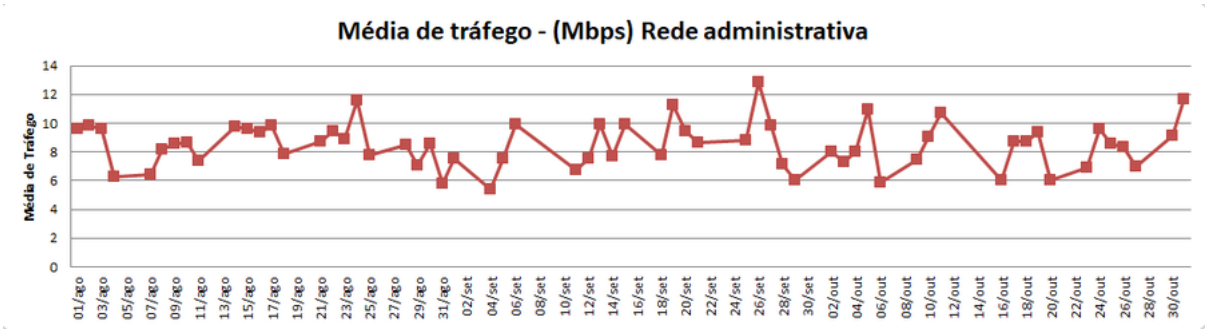
Figura 15 – Pico de Tráfego Administrativo



Elaborado pelo autor

Observa-se na figura acima 16 que a oscilação, entre os dias analisados, variou entre 20 Mbps e o limite da rede (50 Mbps). Outra informação que se pode agregar a esta análise é que, apesar de os picos diários atingirem o limite da conexão, a média de consumo desse ambiente nos dias letivos é consideravelmente inferior em relação aos picos, conforme a figura 16 :

Figura 16 – Média de tráfego - Rede Administrativa

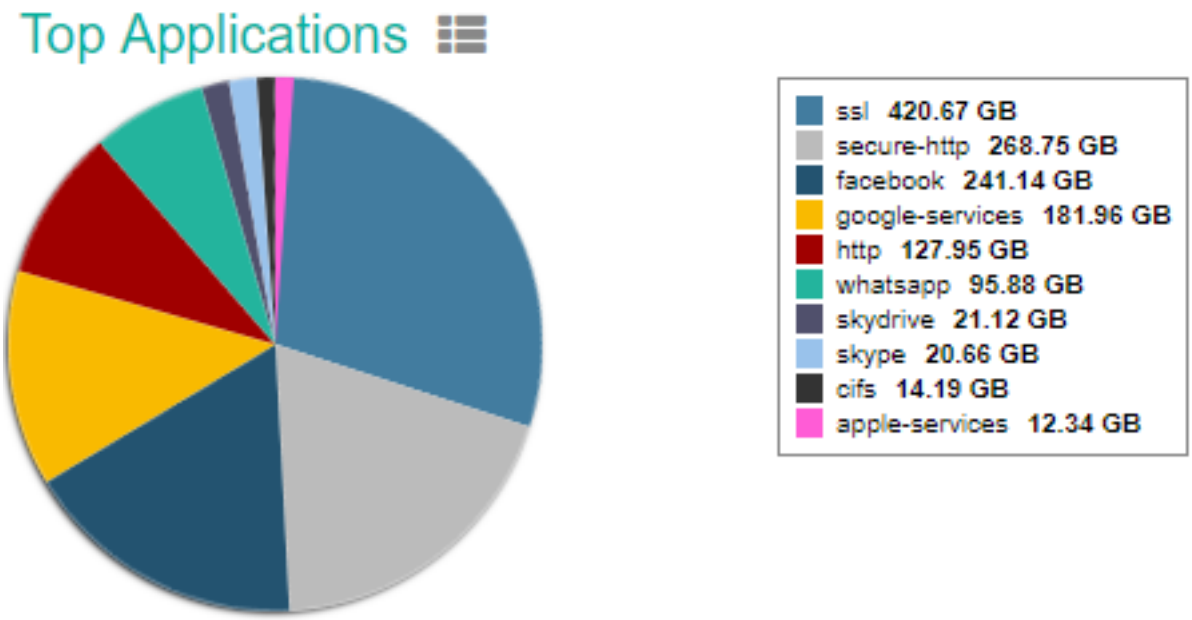


Elaborado pelo autor

Observa-se que a variação existente entre o dia letivo com menor e maior média de tráfego foi de 5,4 Mbps e 12,87 Mbps.

A seguir apresenta-se a estatística das principais aplicações na rede sem fio do *campus*, o que representa cerca de 70% do total de dispositivos conectados na rede como um todo (cabeada e sem fio).

Figura 17 – Estatística da rede sem fio



A figura 17 representa a estatística de acesso da rede sem fio do *campus,* com os dados referentes aos serviços mais consumidos do tráfego agregado. Percebe-se que os perfis de serviços de rede mais consumidos são comuns à maioria das redes mistas como sites de pesquisas, redes sociais, sistemas de comunicação, *streaming* de vídeo etc.

Baseado nas estatísticas apresentadas, pontua-se a seguinte situação:

- A carga de tráfego da rede acadêmica, sem a utilização de um controle secundário de taxas de transmissão, consumiria toda a capacidade de tráfego disponível;
- Apesar de possuir demanda de tráfego superior à taxa de vazão disponível pelo serviço contratado, 33 % do serviço de conexão à internet encontra-se subutilizado;
- Apesar de os picos de tráfego da rede administrativa atingirem o limite da capacidade da rede, o consumo médio de taxa de transmissão deste ambiente em dias letivos representa um total de 17% da taxa disponível;
- O método atual de gerenciamento de tráfego entre os ambientes administrativo e acadêmico possui sua eficiência limitada.

Sabe-se a importância de se manter a qualidade no serviço de acesso à internet para a comunidade acadêmica das instituições de ensino, e o foco dos administradores de redes dessas instituições se concentra em prover o máximo possível de qualidade no acesso à internet.

Necessita-se agregar otimização no uso dos recurso de rede nesses ambientes de ensino, observando com maior critério a questão do desperdício de banda e criar um mecanismo que possibilite aproveitar ao máximo a conexão que se tem contratada.

A proposta a seguir descreve a aplicação dos recursos disponíveis pelas Redes Definidas por *Software* para agregar uma otimização do uso da rede aos planos de controle e de dados desta rede, que seja personalizável de acordo com os interesses e prioridades institucionais, e que seja compatível com as infraestruturas existentes, sem necessidade de mudanças impactantes em sua topologia.

3.2 Proposta de Estratégia de QoS

No intuíto de se buscar a otimização da rede, definiu-se o objeto da proposta de solução que consiste em automatizar mecanismo de monitoramento e adotar uma estratégia de modelagem de tráfego com foco no combate ao desperdício, definição de prioridades e controle de tráfego excessivo.

Ao realizar o estudo sobre as tecnologias existentes na atualidade, foi identificada a oportunidade de adotar o método de QoS baseado em *DiffServ*, pensando em uma arquitetura topológica análoga a de um provedor de internet (ISP). Nesse caso, como o tráfego IP a ser tratado é misto (Figura 17)e a necessidade que se busca nesta dissertação é obter um melhor aproveitamento do tráfego do enlace contratado, será explorada a métrica de vazão e para isso usou-se a classe *Assured Forwarding* (AF) de *DiffServ* conforme explicado na seção 2.5.4.

3.3 Definição dos perfis de tráfego para o *campus*

A partir dos dados levantados no ambiente de estudo, foram definidos os perfis de tráfego para o *campus*, que deverão atuar de acordo com a tabela 7 :

Tabela 6 – Cálculo para definição de perfil QoS para Rede Administrativa:

Perfil QoS Administrativo	
Capacidade total de transmissão	50 Mbps

Perfil QoS Administrativo	
Percentual de consumo em relação a média total	33%
Média do Tráfego de Rajada (picos de tráfego em Mbps)	32,64
Prioridade de tráfego	Primária

Entende-se que, institucionalmente, o tráfego da rede administrativa do *campus* possui prioridade perante o perfil acadêmico, portanto o valor estipulado para garantia de banda do ambiente administrativo teve como base a média registrada dos picos de tráfego nos dias letivos, arredondando-se para 33 Mbps.

Desta forma, no perfil admnistrativo foi garantido que todos os fluxos com taxas inferiores ao valor de 33 Mbps, possuirão prioridade de vazão. Os fluxos que extrapolarem esta taxa não serão descartados diretamente. Os fluxos identificados com taxa igual ou superior ao definido, terão os rótulos de seus pacotes re-marcados em um (1) nível de prioridade e, dependendo da carga concorrente no meio de transmissão, estima-se que não sofrerão impacto sobre o desempenho.

Como esta modelagem envolve apenas dois domínios de rede, a regra de prioridade estipulada para o ambiente acadêmico será a diferença entre a capacidade máxima de tráfego e a regra de QoS definida para o perfil administrativo. Sendo assim, no ambiente acadêmico, todo o fluxo acima de 17 Mbps será tratado como excedente e terá seu DSCP remarcado, reduzindo em um nível seu *Assured Forwarding*(AF).

Tabela 7 – Cálculo para definição de perfil QoS para Rede Acadêmica:

Perfil QoS Acadêmico	
Prioridade	Secundária
Capacidade total de transmissão	50 Mbps
Percentual de consumo em relação à média	67%
Rótulo DSCP remarcado (diferença tráfego total - perfil adm)	17 Mbps

É importante ressaltar que, apesar de o tráfego gerado pela rede acadêmica possuir prioridade inferior, em nenhum momento ele será limitado por técnicas estáticas de enfileiramento. Os fluxos serão remarcados, podendo ou não serem descartados, dependendo do que esteja trafegando na rede administrativa. A partir deste novo método de compartilhamento de tráfego, este perfil não terá mais seu tráfego limitado estaticamente.

Além disso, um fator que não se tinha controle no método antigo é que o controle de tráfego secundário da rede administrativa só era ativado mediante a detecção de alguma anormalidade pela equipe de TI, podendo-se citar como hipótese: se algum dispositivo estivesse a utilizar todo o tráfego da rede, a rede acadêmica sofreria este impacto até que a anormalidade fosse detectada e a intervenção no sistema fosse realizada manualmente pela equipe do setor. No novo cenário, possibilita-se garantir QoS também para o ambiente acadêmico. O consumo administrativo pode chegar até o limite da rede porém, somente se não estiver nenhum tráfego acadêmico concorrendo, garantindo também, QoS para a rede acadêmica.

4 AMBIENTE DE EXPERIMENTAÇÃO

Nesta seção será apresentada a descrição do ambiente de experimentação criado a partir das premissas levantadas. Vale ressaltar que múltiplos ambientes foram testados, como exemplo: a tentativa de executar a experimentação em um ambiente físico onde foi dedicado um servidor de rede com quatro interfaces físicas e sistema operacional nativo. Problemas como a falta de compatibilidade de drivers, impossibilidade de restaurar configurações a um estado anterior impossibilitaram a continuidade dos testes no ambiente físico.

O ambiente experimental visou se aproximar ao máximo das características da rede real e os itens utilizados na proposta estão descritos na tabela a seguir:

Tabela 8 – Recursos utilizados para a experimentação

Descrição	Tipo	Quantidade	Versão
Servidor de rede Cisco UCS 200	equipamento	01	
vSphere ESXi 5.5	hypervisor	01	5.5
Linux Ubuntu	sistema Operacional	01	14.04.5
Mininet	*software*	01	2.2.1
Controlador Ryu	*software*	01	3.6
OfSoftSwitch	*software*	03	1.3
iperf	*software*	01	2.0.5
Css Materialize	*software*	01	1.0
AngularJs	*software*	01	1.3

Elaborado pelo autor

No ambiente de experimentação optou-se por utilizar as vantagens dos recursos de virtualização, dentre elas, não ser necessário dedicar um equipamento exclusivo para o experimento. Por representar um grau de risco à execução desses experimentos em ambientes de produção, cuidados foram tomados como, por exemplo: o isolamento de interfaces de rede, para que os testes não impactassem na dispobilidade dos outros serviços que compartilham do mesmo *hardware*, entre outros.

O servidor de rede, possui instalado o *HyperVisor vmWare ESXi* (VMWARE, 2017). No decorrer da execução das simulações, optou-se pelo uso de apenas um (1) *host* virtual, com a seguinte configuração de *hardware*: oito núcleos de processamento

(2,66Ghz) 16 GB de memória RAM, 1 Interface de rede isolada e disco de 32 GB de armazenamento.

O sistema operacional adotado foi o Linux Kubuntu (KUBUNTU, 2017) na versão 14.04. Optou-se por esta distribuição, por já possuir familiaridade no uso com bom desempenho, e ser a versão utilizada na versão disponibilizada pela SDN *Hub* (SDNHUB, 2017).

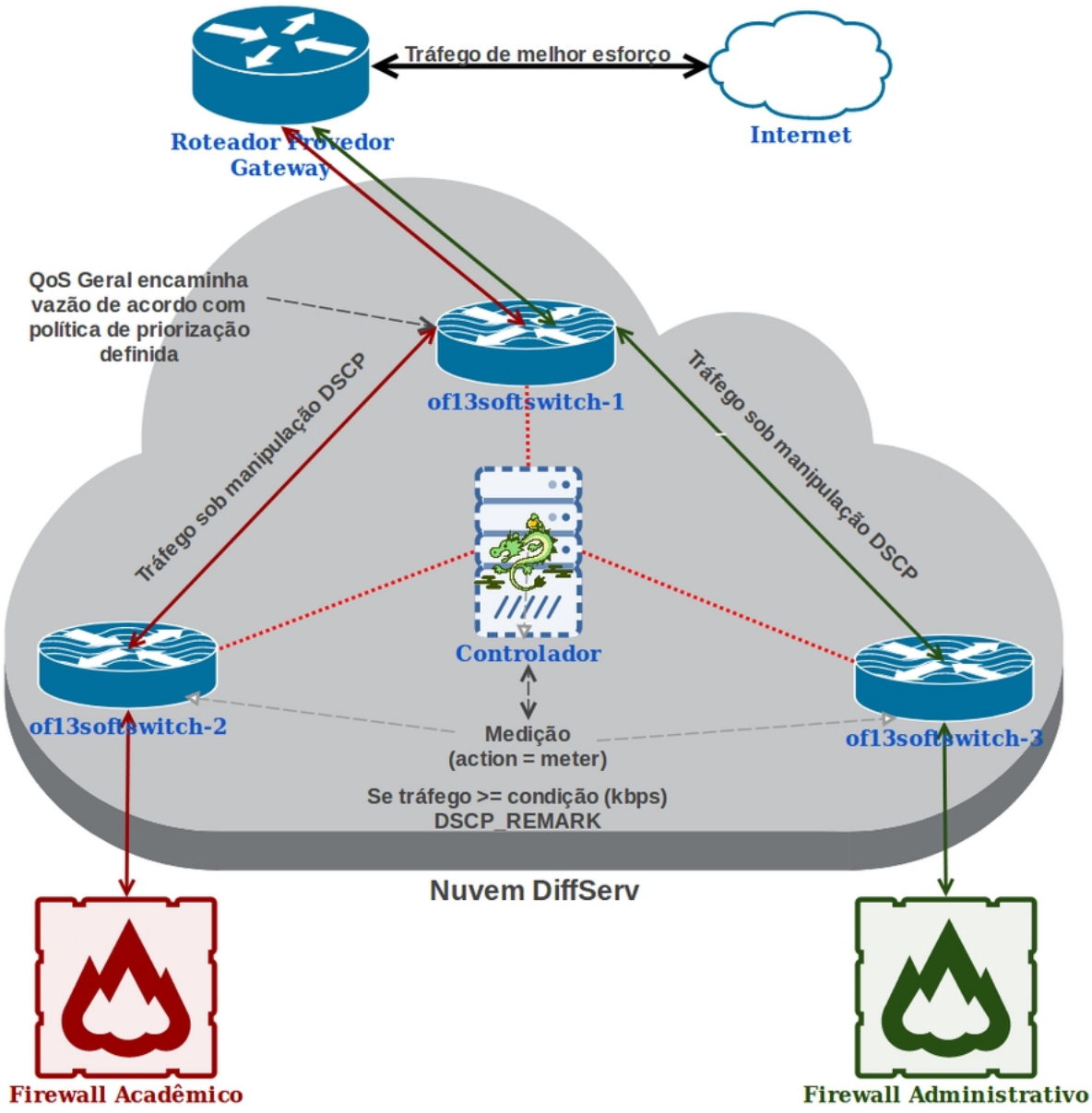
Para a emulação do *switch* virtual, optou-se por utilizar o *OfSoftSwitch* (CPQD, 2017a) seguindo a recomendação da comunidade Ryu, para o tratamento de *meters* no *OpenFlow* 1.3. Como ferramenta de geração de tráfego, optou-se por usa uma das mais populares nos experimentos de SDN, o iPerf (IPERF, 2017).

Para o desenvolvimento da interface *web* para a aplicação, utilizou-se a ferramenta CSS *Materialize* (MATERIALIZE, 2017) juntamente com o *framework java script* [1] AngularJS (ANGULAR.IO, 2017).

A estratégia proposta está representada de acordo com a figura18 :

[1] JavaScript é uma linguagem de programação interpretada muito ultilizada para o desenvolvimento de aplicações web.

Figura 18 – Topologia proposta - nuvem DiffServ



Elaborado pelo autor

Nessa nova topologia, substitui-se o monitoramento manual anterior por um conjunto de recursos virutais e a tomada de decisão foi definida por *software*. Para esse cenário, foi criado uma nuvem *DiffServ* permitindo que os *switches* pudessem aplicar as regras de QoS para o tráfego gerado. A descrição detalhada do ambiente experimental está presente na seção a seguir.

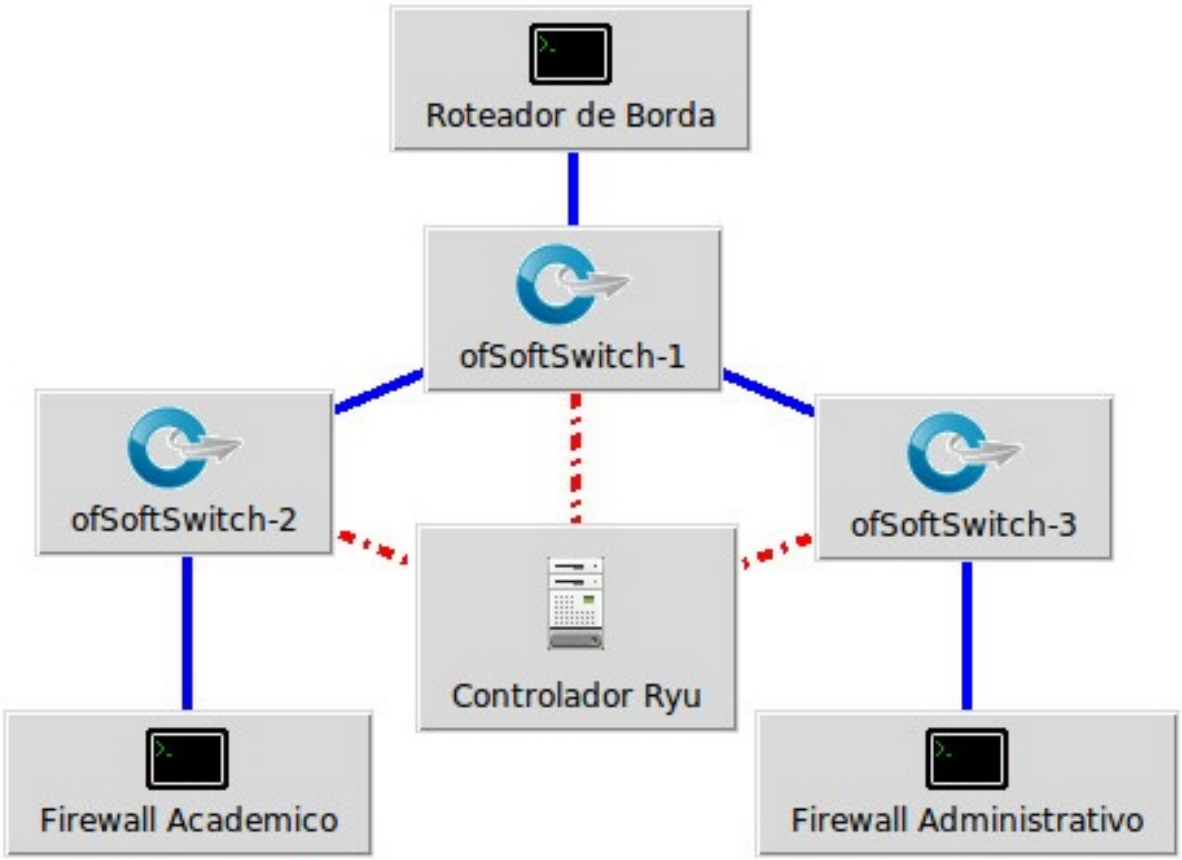
4.1 Especificação do Protótipo

A topologia proposta na seção anterior, foi transferida para o ambiente de emulação de redes, pela aplicação escrita em *python* da biblioteca padrão da documentação

do controlador Ryu (RYU PROJECT TEAM, 2017) e sua execução inicia o *mininet* com este ambiente pré configurado.

A figura 19 ilustra o ambiente virtual emulado pela aplicação gráfica do *mininet*, no caso, o *miniedit.py*:[2]

Figura 19 – Ambiente virtual emulado



Elaborado pelo autor

O ambiente emulado é formado pelos seguintes componentes:

- **Terminal-1 (Roteador de borda)**: componente que se conecta à porta um (1) do *ofSoftSwitch*-1. Representa o roteador da borda na topologia, que se conecta à internet e compartilha o serviço de conexão contratado. Todo o tráfego da nuvem *diffserv* terá ele como destino.
- **O ofSoftSwitch-1**: *Switch* de núcleo da nuvem *DiffServ.* Nele está instalado o *ofSoftSwitch* 1.3 a nível de usuário, assim como nos outros dois *switches*. Nele chegam os fluxos provenientes dos *switches* de borda da nuvem. Possui quatro interfaces virtuais onde a interface um (1) se conecta ao roteador de

[2] Interface gráfica do mininet que fornece os mesmo recuros da versão CLI, possibilitando a criação e visualização das topologias.

Borda (acesso à internet), a interface dois (2) se conecta ao *ofSoftSwitch*-2, interface três (3) se conecta ao ofSoftSwitch-3 e a interface 4 que se conecta ao controlador. Neste *switch* são instaladas as regras de QoS e chegam até ele os fluxos com o rótulo DSCP alterado. Ele processa estes fluxos e aplica o QoS, dando vazão ao tráfego para o roteador de borda. A comunicação entre *switches* acontece separadamente do tráfego dos *firewalls,* através de uma subrede exclusiva. A comunicação com o controlador também ocorre separadamente, por um canal exclusivo *OpenFLow.*

- **O ofSoftSwitch-2**: *Switch* da borda na nuvem DiffServ que recebe as requisições de tráfego da rede acadêmica. Possui 3 interfaces A interface 1 se conecta ao ofsoftswitch-1 (nucleo da nuvem), a interface 2 se conecta ao firewall administrativo e a interface 3 se conecta ao controlador. Todos os fluxos entrantes são medidos e recebem a manipulação do rótulo DSCP do cabeçalho IP de acordo com as políticas criadas, descritas posteriormente.

- **ofSoftSwitch-3**: *Switch* da borda na nuvem DiffServ que recebe as requisições de tráfego da rede administrativa. Assim como o ofsoftswitch-2, possui 3 interfaces. A interface 1 se conecta ao ofsoftswitch-1 (nucleo da nuvem), a interface 2 se conecta ao *firewall* administrativo e a interface 3 se conecta ao controlador. Todos os fluxos entrantes são medidos e recebem também a manipulação do rótulo DSCP do cabeçalho IP.

- **terminal-2 (*firewall* acadêmico)**: componente que simula o *gateway* da rede acadêmica. Se coneta ao *switch* 2 da borda da nuvem DiffServ. Este host tem como *gateway* o host 1(roteador de borda) e a comunicação entre eles ocorre através de uma sub-rede exclusiva.

- **terminal-3 (*firewall* administrativo)**: host que simula o *gateway* da rede administrativa. Se coneta ao *switch* 3 da borda da nuvem *DiffServ.* Este host tem como *gateway* o host 1(roteador de borda) e é integrante da mesma subrede onde está o host 1 e 2.

- **Controlador Ryu**: Controlador responsável por gerenciar esta nuvem DiffServ. Possui o controlador Ryu instalado a nível de usuário. Está conectado aos 3 *switches* através de um canal *OpenFLow.* Por ser um ambiente de testes, para este canal não foi definida uma comunicação criptografada. Neste executa-se a aplicação *qos_sample_topology.py* que cria esta topologia, estabelece os links entre os componentes e se posiciona como controlador da rede. O exemplo consta no anexo deste documento.

Nesta etapa, ainda não existe comunicação entre nenhum desdes componentes. Para isso acontecer, executa-se a aplicação *qos_simple_switch.py*. Esta aplicação, possui a função de um *switch* com suporte a QoS e ao *OpenFlow* 1.3. Além desta, executa-se no controlador a aplicação *rest_qos.py* que tem a finalidade de criar a interface REST API para manipulação dos parâmetros de QoS a serem utilizados. Vale ressaltar que estas aplicações foram implementadas e executadas com base nos exemplos presentes na documentação do controlador Ryu, apresentado anteriormente.

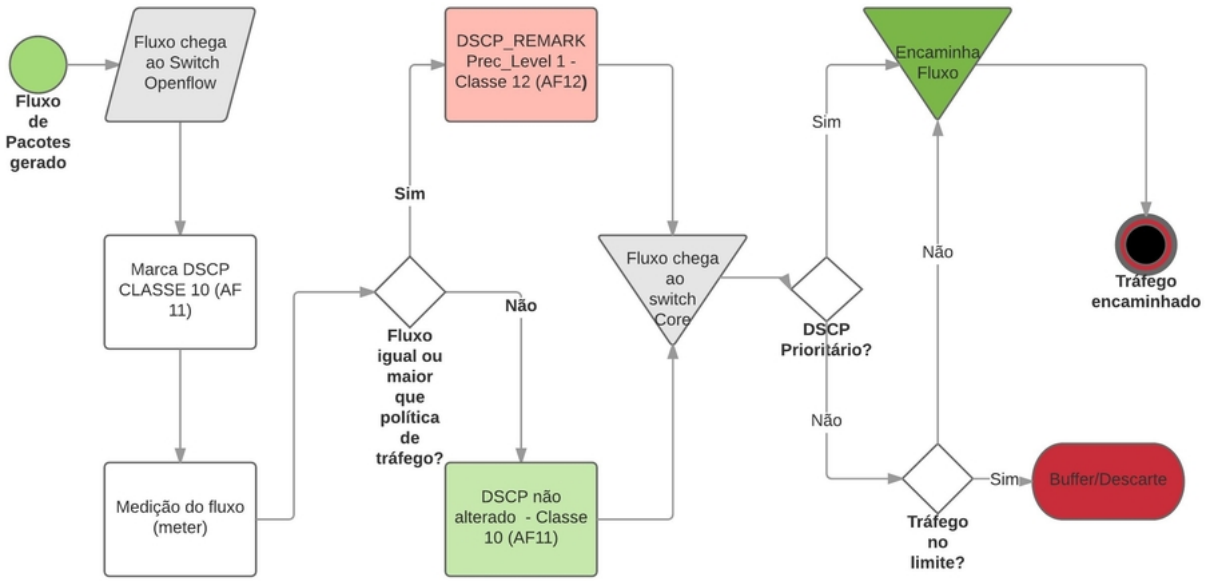
Após o ambiente de simulação ser criado e a comunicação entre os componentes ter sido estabelecida, chega se a etapa de aplicação da estratégia de QoS proposta. A interação entre os componentes descritos na figura 18 está descrita na seção a seguir.

4.2 Aplicação da proposta

Nesta seção, será aplicada a proposta de QoS através da configuração dos componentes da topologia proposta, de acordo com identificação dos perfis de tráfego apresentados na seção 3.3 e no ambiente virtual criado de acordo com a seção 4.1.

O fluxograma representado pela figura 20 apresenta a forma como os fluxos de pacotes são tratados nesta nova topologia:

Figura 20 – Fluxograma QoS



Elaborado pelo autor

Conforme a figura 20 apresenta, o fluxo é gerado e encaminhado ao *switch* de borda do domínio DiffServ (*Switch* 2 ou *Switch* 3). Na entrado do *switch*, todos os pacotes, em seu cabeçalho IP, recebem a marcação DSCP para a classe AF11 (baixa

possibilidade de descarte). Na próxima etapa, ainda no *switch* de entrada, este fluxo é medido. Neste *switch*, está configurado a regra de remarcação dos pacotes de acordo com o tamanho do fluxo definido pela política de tráfego. Caso o tamanho do fluxo seja igual ou maior do que a taxa definida, o DSCP é remarcado na saída do *switch* para a classe AF12(média probabilidade de descarte dos pacotes) e em seguida, encaminhado para o *switch* Core. Este é o que concentra as regras de QoS aplicando o controle de oferecendo a vazão ou descarte de pacotes. Desta forma, realiza o enfileiramento, priorizando os pacotes marcados com DSCP AF11 caso o enlace esteja congestionado.

Primeiramente, houve a necessidade de customizar o ofsoftswitch que, por padrão, possui sua taxa de transmissão de acordo com a capacidade da interface em que está instalado, ou até 1Gbps. Desta forma, foi alterado em seu código fonte o componente *netdev*, fixando a taxa máxima do dispositivo em 50Mbps, que é a limitação encontrada no ambiente da proposta.

Taxas de prioridades de vazão dos servidores de acordo com a política definida:

Tabela 9 – Política de tráfego a ser aplicada

Firewall Acadêmico - *Switch* 02			Firewall Administrativo - *Switch* 03	
Taxa máxima	50 Mbps		Taxa máxima	50 Mbps
Prioridade até	17 Mbps		Prioridade até	33 Mbps

Elaborado pelo autor

Para isso, foram criadas as regras de QoS de acordo com a tabela a seguir:

Tabela 10 – Especificação QoS - *Switch* de núcleo

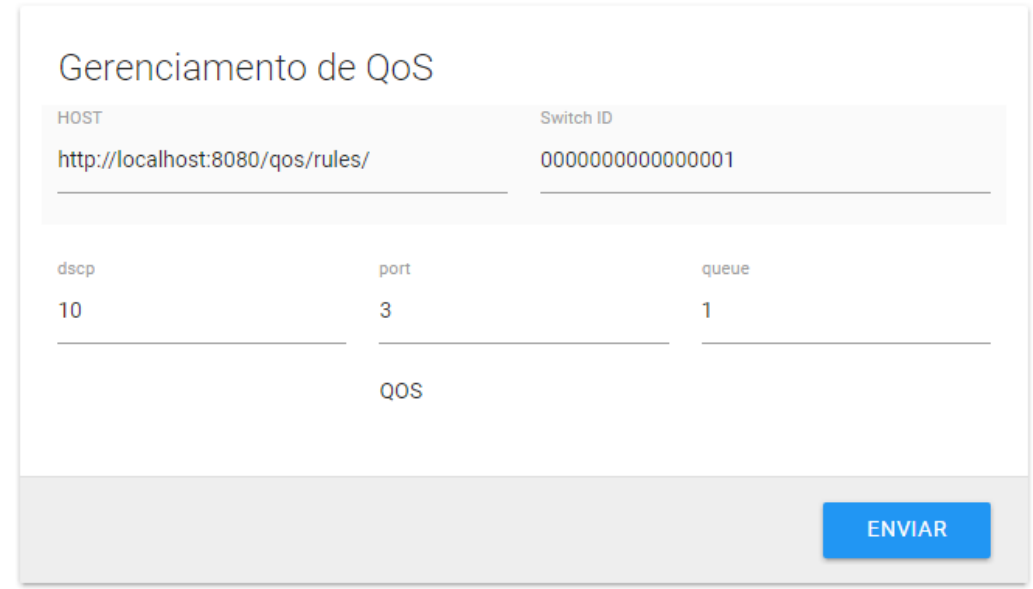
Especificação QoS - *Switch* 01			
DSCP	**Queue**	**QoS ID**	**Interface**
0	1	1	s1-eth1
10	3	2	s1-eth1
12	2	3	s1-eth1
0	1	4	s1-eth2
10	3	5	s1-eth2
12	2	6	s1-eth2

Elaborado pelo Autor

Para configurar o *switch* core, necessita-se primeiramente, criar uma regra de QoS, que recebe um número como identificador (QoS ID). Esta QoS fica associada ao ID de uma fila e ao valor DSCP criando um grupo de prioridade. Desta forma, quando um pacote chega ao *switch* de núcleo, ele irá verificar em qual QoS será associado, para a tomada de decisão de descarte ou encaminhamento.

Devido aos comandos serem muito extensos e complexos, foi desenvolvida uma aplicação web com interfaces para a consulta e inserção das regras QoS, visando facilitar a visualização e manipulação dos *switches,* conforme demonstra a figura 21 :

Figura 21 – Interface web de gerenciamento de QoS



switch_id: 0000000000000001

qos-id	type	queue	priority	port	dscp
1	IPv4	1	1	2	
2	IPv4	3	1	2	10
3	IPv4	2	1	2	12

Elaborado pelo Autor

Para os *switches* 2 e 3 as regras de medição/remarcação (se for o caso) são implantadas também via REST API conforme a tabela 11 :

Para configurar os *switches* de borda, necessita-se primeiramente criar regras de QoS independentes para cada *switch*, sendo necessário especificar um identificador para o QoS e o identificador para o Medidor. Posteriormente, associa-se a este medidor a ação de remarcar os pacotes juntamente com a taxa de tráfego em que se deve realizar.

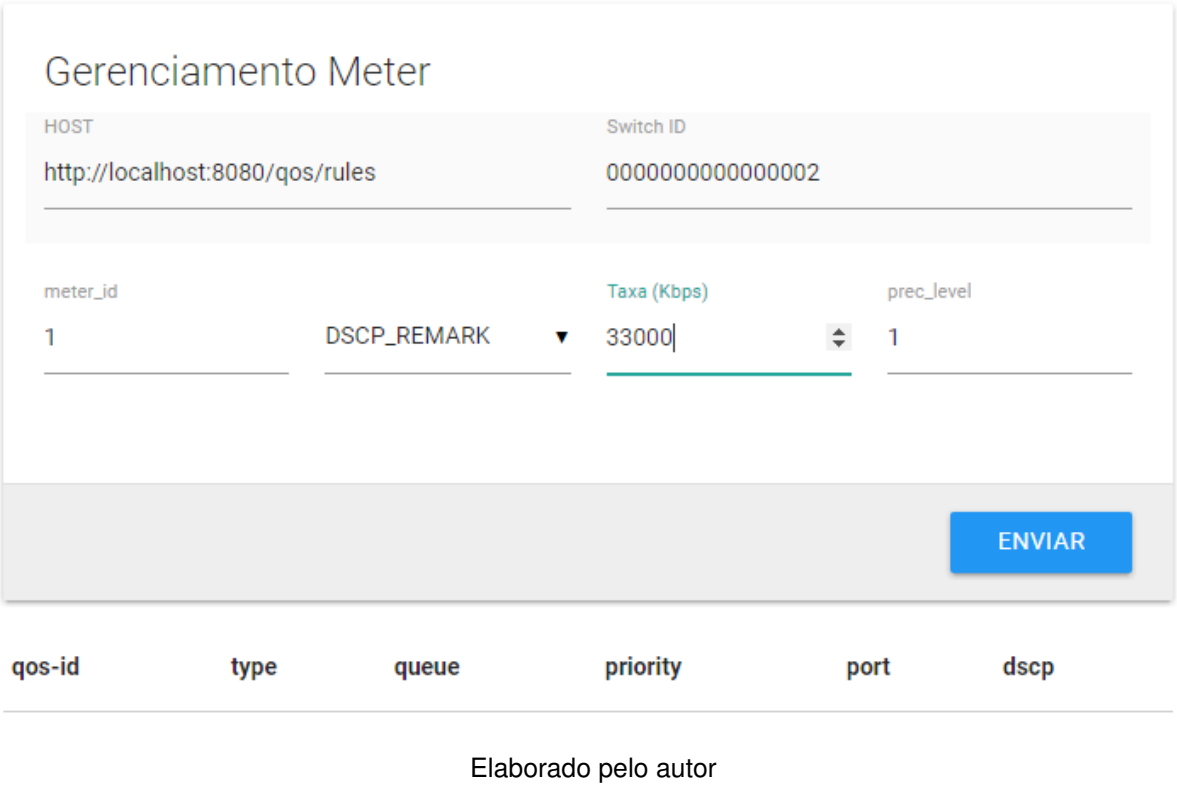
A configuração destas segue o mesmo método citado anteriormente, onde foi desenvolvido uma interface web via REST API para manipular as configurações, conforme a figura 22 :

Tabela 11 – Especificação das taxas para remarcação do DSCP

Especificação QoS - Remarcação DSCP						
Switch	**Meter ID**	**QoS ID**	**DSCP**	**Taxa de medição**	**Ação**	**DSCP-novo**
Sw2-eth1	1	1	10	17 Mbps	DSCP Remark	12
Sw3-eth1	1	1	10	33 Mbps	DSCP Remark	12

Elaborado pelo Autor

Figura 22 – Interface web para gerenciamento de QoS/Meter



Elaborado pelo autor

Os exemplos dos comandos executados encontram-se nos anexos deste documento.

5 AVALIAÇÃO

Nesta etapa deste trabalho, é realizada a avaliação da proposta implementada. Através da análise do ambiente realizada anteriormente, foram definidos quatro cenários virtuais.

Como prova de conceito, foram realizadas 30 repetições para cada um destes testes. Resultados baseados em métricas de QoS como vazão, perda e latência serão apresentados além do ponto chave que consiste em verificar se a proposta proporciona melhor aproveitamento no uso do enlace contratado.

5.1 Ambiente sem suporte à QoS

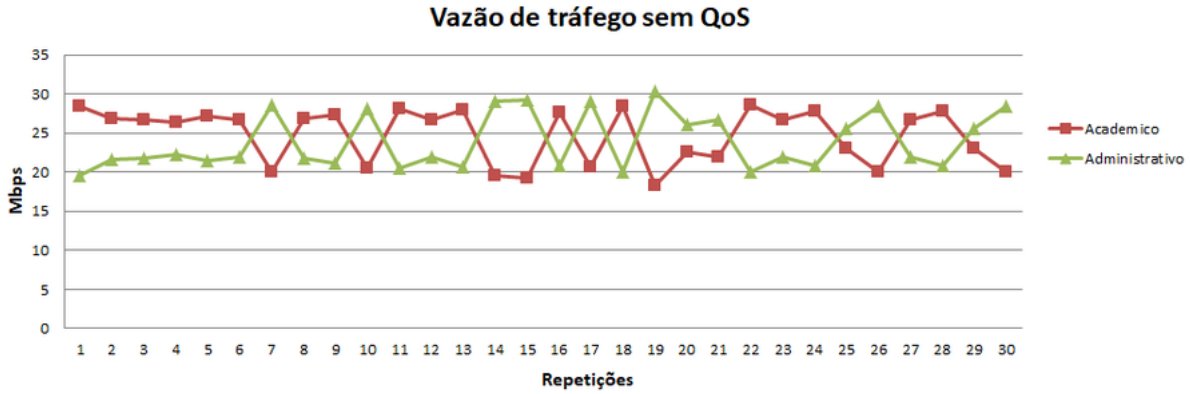
Antes de iniciar as simulações sobre a estratégia adotada, foi realizado uma simulação no mesmo ambiente virtual, mas ainda sem as regras de QoS implementadas. A tabela a seguir exibe os parâmetros utilizados para este primeiro teste:

Tabela 12 – Parâmetros de simulação: cenário sem QoS

Ambiente	Tráfego Gerado	Tempo	Taxa de tráfego gerado	Repetições	DSCP-Remark
Acadêmico	895 Mbytes	150 segundos	50Mbps	30	–
Administrativo	895 Mbytes	150 segundos	50Mbps	30	–

O gráfico da figura 23, ilustra o resultado desta simulação, baseado pelas taxas de vazão dos dois ambientes.

Figura 23 – Simulação do Ambiente sem QoS



Elaborado pelo autor

Pode-se perceber que sem a estratégia de QoS implementada, o trafego é de maior esforço e a disputa por vazão ocorre por igual. Caso houvesse necessidade de priorização de tráfego neste caso, não seria possível oferecer qualidade de serviço. A tabela a seguir ilustra o resultado médio das métricas avaliadas para este cenário.

Tabela 13 – Resultados do teste - 50/50 sem QoS

Ambiente de Rede	Média de Vazão	Taxa média de transferência	Latência	Perda
Acadêmico	442 Mbytes	24,72 Mbps	1,88 ms	50,6 %
Administrativo	426 Mbytes	23,88 Mbps	1,17 ms	52,3 %

Analisando o resultado das métricas da simulação deste ambiente, observa-se que os valores ficaram similares, e o ponto crítico de ambos ambientes é a perda. Ambos ambientes tiveram descarte próximo da metade do tráfego gerado pois a carga gerada foi o dobro suportado pelo enlace. Para ambos, a latência apresentou níveis baixos.

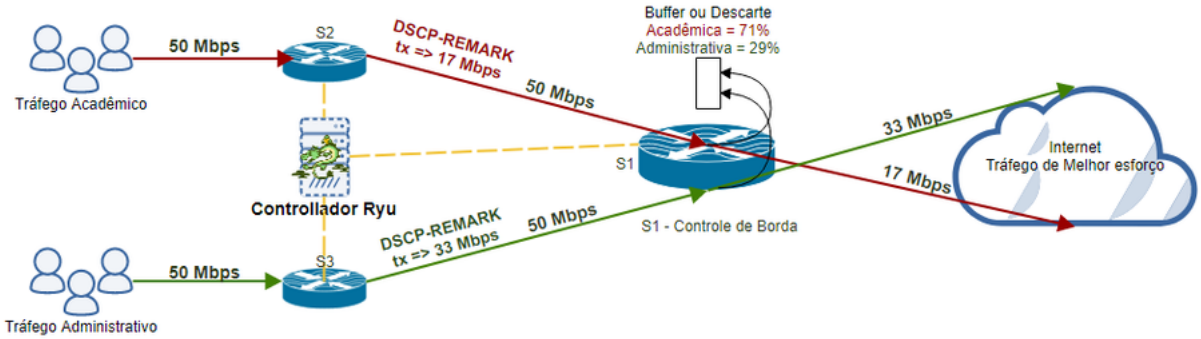
No ambiente real, este cenário é vivenciado, caso o controle de tráfego primário (8) não esteja habilitado. O método existente de controle de tráfego, só possui duas opções: Ou limita-se a taxa de vazão da rede acadêmica em um valor determinado para que o restante de tráfego possa ser garantido a rede administrativa, ou esta opção apresentada, sem limitação de banda.

A seguir, serão apresentados os cenários com simulações usando QoS no controle do tráfego.

5.2 Cenário 1 - Tráfego máximo com suporte à QoS - 50/50

O teste realizado neste cenário consistiu em simular um ambiente extremo de tráfego onde os dois perfis de rede geravam tráfego na quantidade limite da rede (50/50) ficando a cargo do sistema implementado administrar os fluxos e atender aos requisitos de prioridade de vazão. A figura 24 ilustra o ambiente simulado deste cenário:

Figura 24 – Cenário 2 - Tráfego máximo



Elaborado pelo autor

Desta forma, gerou-se tráfego de 50Mbps a partir da rede acadêmica concomitantemente com a rede administrativa. Devido a limitação do meio de transmissão (50Mbps), pressupõe-se que metade deste tráfego será descartado (tráfego agregado de 100 Mbps) e a vazão será de acordo com as regras de priorização.

A tabela a seguir apresenta os parâmetros utilizados para simulação do ambiente através da geração de tráfego máximo:

Tabela 14 – Parâmetros de simulação cenário 01

Ambiente	Tráfego Gerado	Tempo	Taxa de tráfego gerado	Repetições	DSCP-Remark
Acadêmico	880 Mbytes	150 segundos	50Mbps	30	17 Mbps
Administrativo	880 Mbytes	150 segundos	50Mbps	30	33Mbps

Como o objetivo é verificar a priorização do tráfego via *software* através das regras de QoS implantadas, os parâmetros estipulados para a simulação são iguais. A tabela 19, apresenta o resultado da simulação sob as métricas de QoS:

Tabela 15 – Resultados do teste - cenário 01

Ambiente de Rede	Média de Vazão	Taxa média de transferência	Latência	Perda	Perda(Bytes)

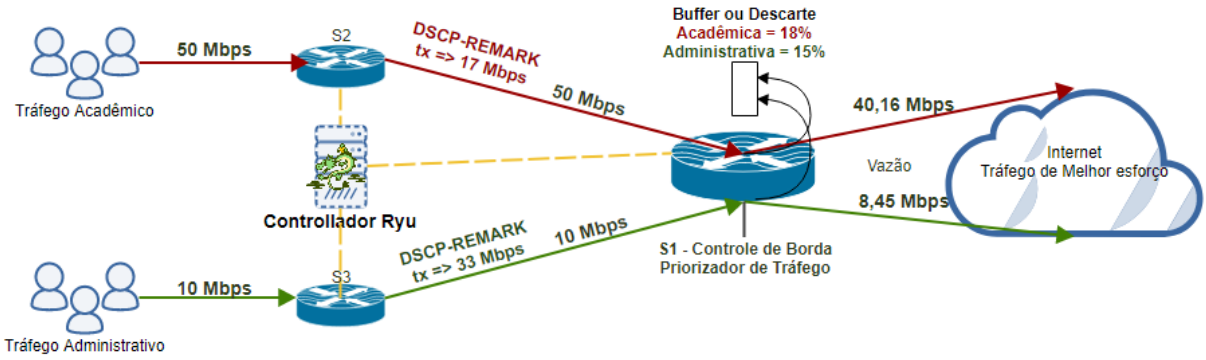
Ambiente de Rede	Média de Vazão	Taxa média de transfe-rência	Latência	Perda	Perda(Bytes)
Acadêmico	251 Mbytes	14 Mbps	3,08 ms	71%	629 Mbytes
Administrativo	617 Mbytes	34 Mbps	2,62 ms	29%	263 Mbytes

Explicando os dados da tabela 19, observa-se que o monitoramento e remarcação dos pacotes ocorreram conforme esperado. O objetivo do QoS é de garantir qualidade de serviço para a rede administrativa até que a ocupação do tráfego atingisse 33 Mbps (70% da capacidade total) comparando a taxa média de transferência (34 Mbps) e o percentual de perda (29%) em relação aos 880 Mbytes gerados 19.

5.3 Cenário 2 - Tráfego ocioso da rede administrativa

O teste realizado neste cenário, consistiu em simular um ambiente em que ocorre a sobra de taxa de tráfego gerado pela rede administrativa. A figura 25 ilustra o ambiente simulado deste cenário:

Figura 25 – Cenário 2 - Tráfego ocioso da rede administrativa



Elaborado pelo autor

Desta forma, gerou-se tráfego de 10 Mbps a partir da rede administrativa e 50 Mbps da rede acadêmica, concomitantemente, totalizando 60 Mbps. Neste caso, apenas pacotes da rede acadêmica terão seus rótulos DSCP re-marcados, pois o tráfego gerado está acima dos 17Mbps da regra de QoS. Devido à limitação física do meio de transmissão, pressupõe-se apesar de possuir prioridade com valor inferior ao da rede administrativa, a rede acadêmica não ficará limitada ao seu percentual de medição e poderá aproveitar a ociosidade gerada pela rede administrativa.

A tabela 19 apresenta os parâmetros utilizados para simulação do ambiente através da geração de tráfego máximo:

Tabela 16 – Parâmetros de simulação cenário 02

Ambiente	Tráfego Gerado	Tempo	Taxa de tráfego gerado	Repetições	DSCP-Remark
Acadêmico	715 Mbytes	150 segundos	50 Mbps	30	17 Mbps
Administrativo	179 Mbytes	150 segundos	10 Mbps	30	33 Mbps

Como o objetivo é verificar a priorização do tráfego via *software* através das regras de QoS implantadas, os parâmetros estipulados para a simulação são iguais. A seguir, apresenta-se o resultado da simulação sob as métricas de QoS:

Tabela 17 – Resultados do teste - cenário 02

Ambiente	Média de Vazão	Taxa média de transferência	Latência	Perda
Acadêmico	718 Mbytes	40,16 Mbps	0,55 ms	18,3 %
Administrativo	151 Mbytes	8,54 Mbps	2,18 ms	15,4 %

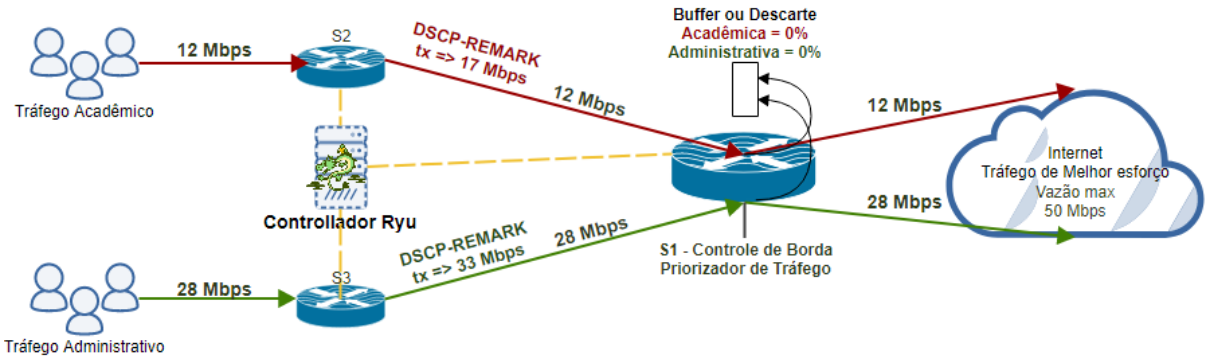
A tabela 19 demonstra que o pressuposto foi alcançado de forma que a largura de banda que não estava sendo usada pela rede administrativa, foi aproveitada pela rede acadêmica. Como os pacotes gerados pela rede administrativa só teriam sua prioridade reduzida após a medição detectar algum fluxo superior a 33 Mbps, não sofreram nenhuma alteração, mantendo seu rótulo DSCP com o valor 10 e prioridade total. Mesmo assim, nos testes pode-se perceber que ocorreu uma perda de 15,4% de pacotes, levando a entender que a alta carga de tráfego da acadêmica, teve uma influência na administrativa.

Levando em consideração o cenário 1 e cenário 2, nota-se que a alta carga de tráfego afeta o desempenho de ambos ambientes, na mesma proporção.

5.4 Cenário 3 - Ambiente ocioso

Como existe momentos em que o tráfego gerado nos dois ambientes, fica constantemente inferior ao nível de medição, foi realizado uma simulação deste cenário, conforme a figura 26 :

Figura 26 – Cenário 3 - Tráfego abaixo da regra QoS



Elaborado pelo autor

A tabela a seguir apresenta os parâmetros utilizados para simulação do ambiente de baixo volume de tráfego:

Tabela 18 – Parâmetros de simulação - cenário 04

Ambiente	Tráfego Gerado	Tempo	Taxa de tráfego gerado	Repetições	DSCP-Remark
Acadêmico	215 Mbytes	150 segundos	12 Mbps	30	17 Mbps
Administrativo	501 Mbytes	150 segundos	28 Mbps	30	33 Mbps

O objetivo deste cenário foi verificar se esta estratégia de QoS influenciaria no desempenho da rede em casos que o volume de tráfego não atingisse o parâmetro de re-marcação DSCP. A tabela apresenta o resultado da avaliação das métricas de QoS neste ambiente e podemos concluir que o desempenho não foi afetado.

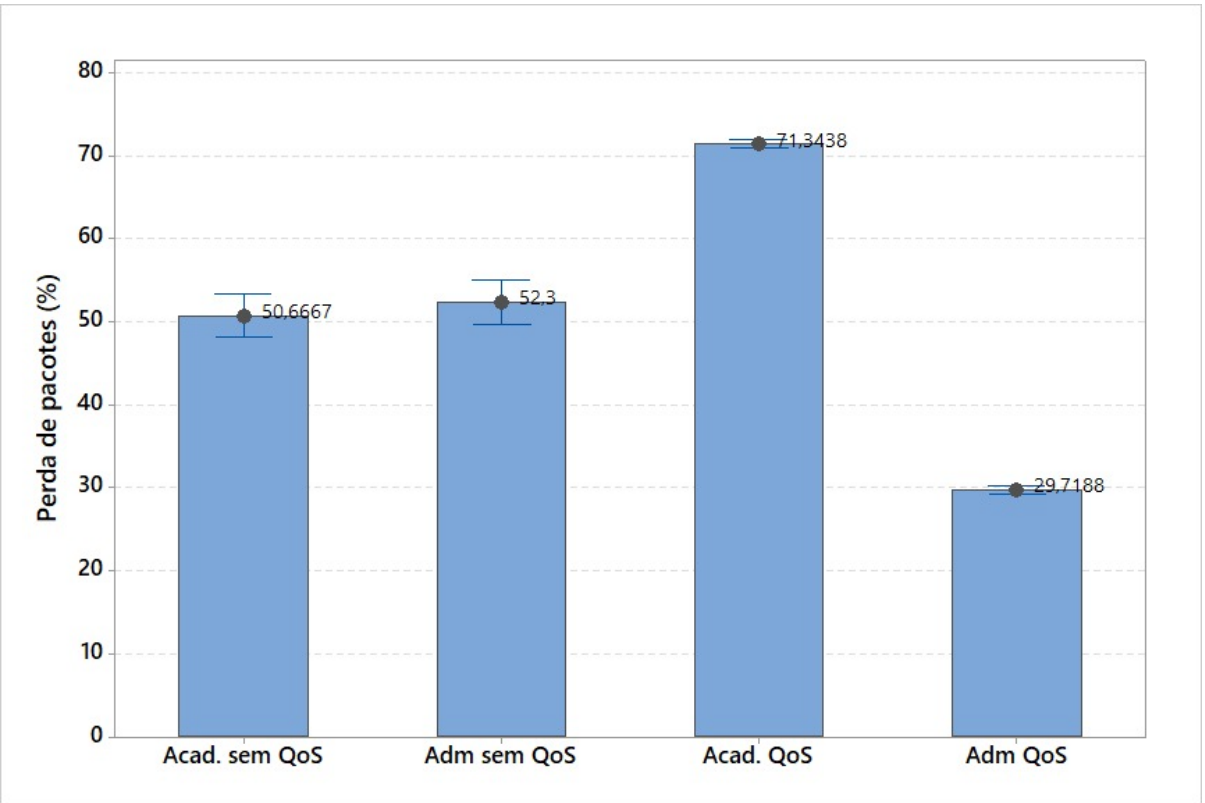
Tabela 19 – Resultados do teste - cenário 03

Ambiente	Média de Vazão	Taxa média de transferência	Latência	Perda
Acadêmico	215 Mbytes	12 Mbps	0,1 ms	0,03 %
Administrativo	501 Mbytes	28 Mbps	0,04 ms	0,02 %

5.5 Resultados

A figura a seguir, ilustra a comparação entre os 2 ambientes, Sem QoS e cenário 1, em relação à perda de pacotes:

Figura 27 – Perda de pacotes - Intervalo de confiança entre ambiente comum e cenário 1 - com QoS

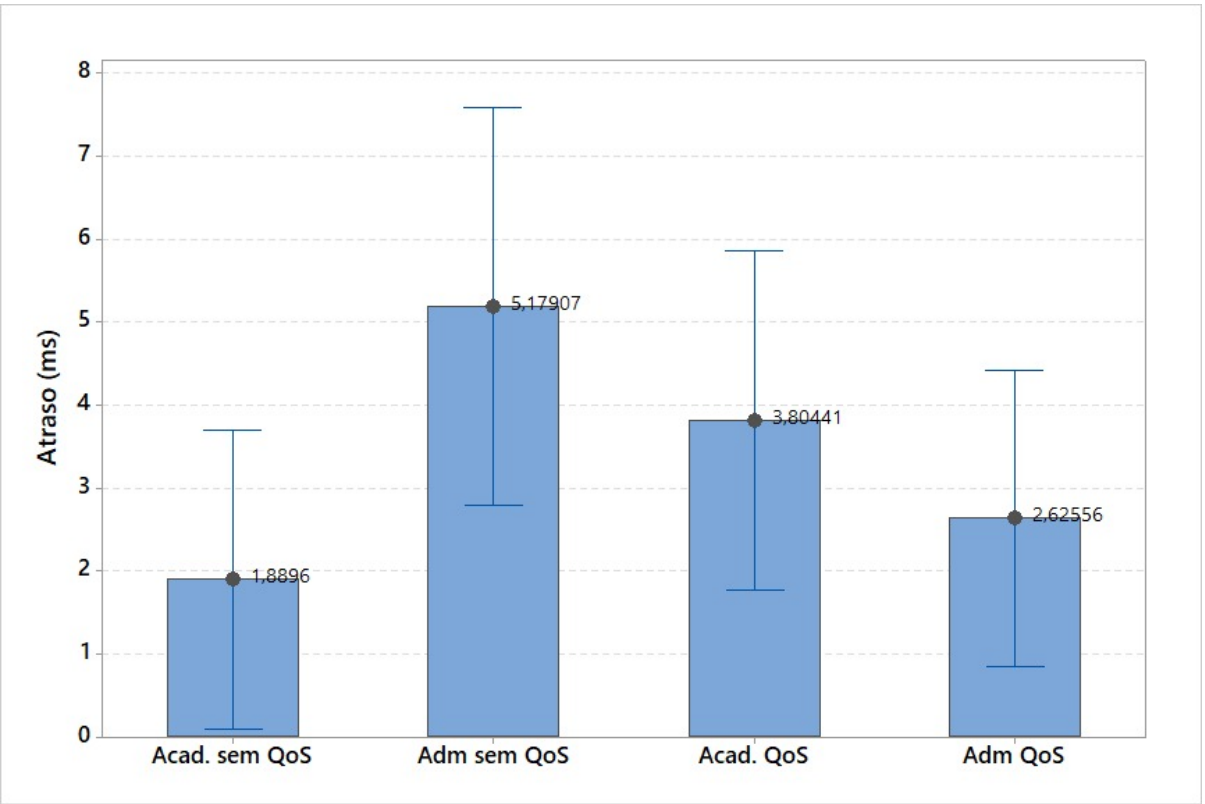


Elaborado pelo autor

Como o tráfego gerado fora o dobro do suportado pelo enlace, aproximadamente metade dos pacotes foram perdidos. O gráfico exibido na figura 27 demonstra a proporção de perda entre os dois ambientes (academico e administrativo) e os dois cenários (com e sem QoS). Vale lembrar que este valor definido como garantia de QoS para a rede administrativa, foi baseado na média dos, apresentada na seção em que foi realizado o estudo de caso (7).

Ainda sobre a figura 27, ressalta-se que , apesar de a rede acadêmica possuir prioridade inferior, não foi completamente comprometida pelo tráfego prioritário da rede administrativa e obteve vazão proporcional ao valor estipulado no perfil de QoS Acadêmico, no caso, 14Mbps (7).

Figura 28 – atraso médio entre os ambientes

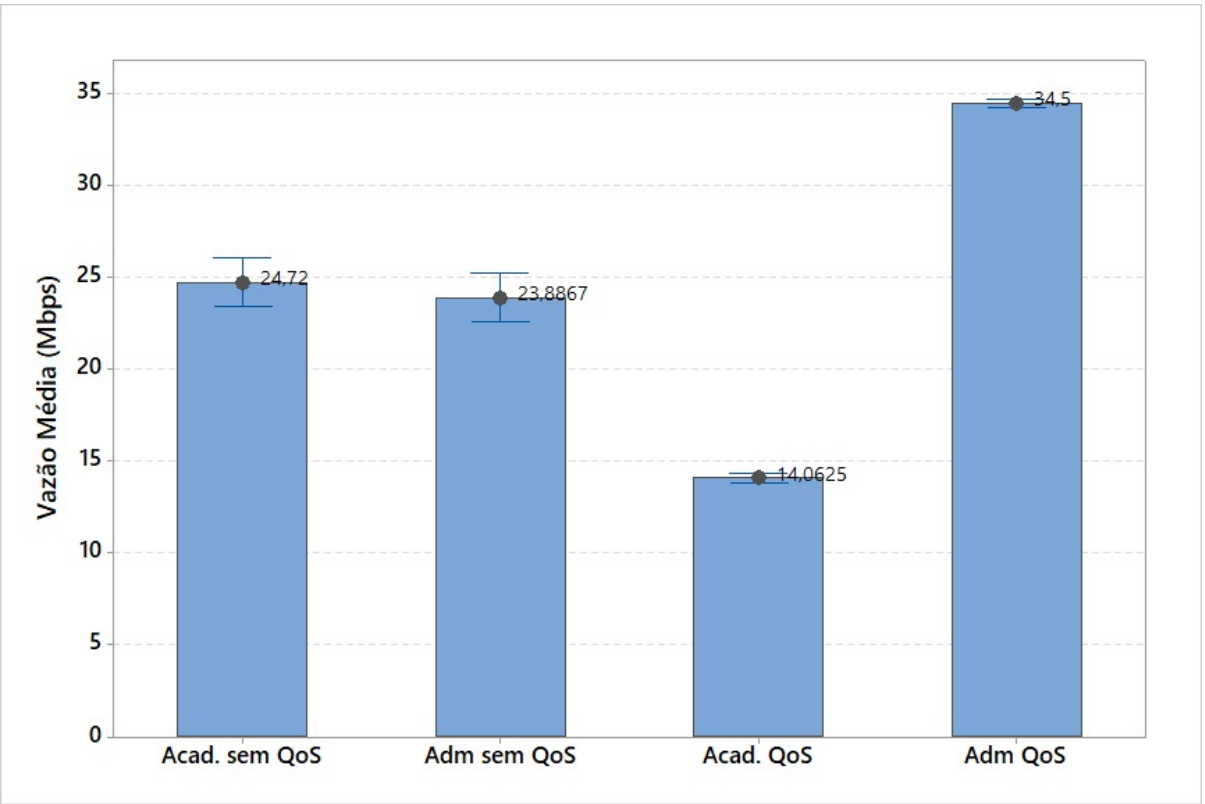


Elaborado pelo autor

Em relação à figura 28 , não houve inserção significativa de atraso ocasionado pelo processo de remarcação, como pode ser observado nos intervalos de confiança das comparações feitas sem e com QoS de cada perfil.

O gráfico exibido pela figura 29 apresenta a comparação entre os ambientes acadêmico e administrativo em relação à taxa de vazão alcançada, primeiramente sem QoS e posteriormente com QoS configurado:

Figura 29 – Vazão média - Comparação entre ambiente comum e cenário 1 - com QoS



Elaborado pelo autor

Mediante a comparação realizada, observa-se que, no cenário sem QoS, a disputa pela vazão ocorreu de forma padrão para os dois ambientes. Já no cenário com QoS configurado, observa-se o funcionamento do controle de QoS de acordo com os valores estabelecidos pela política de tráfego.

6 CONCLUSÃO

Como contribuição apresentada por este trabalho, uma nova estratégia para gerenciar a qualidade de serviço em redes de *campus* foi apresentada. Para isso, foi realizado um estudo sobre o tráfego circulante nesta rede de forma que pode-se definir perfis de tráfego e pontos críticos a serem solucionados que consistiram em priorizar tráfego e aproveitar ao máximo a taxa de vazão do acesso à internet contratada.

Realizou-se um levantamento teórico sobre o gerenciamento/compartilhamento de tráfego em ambientes sobrecarregados. Observou-se a possibilidade de agregar inteligência, principalmente no âmbito das redes definidas por *software*, através da aplicação de técnicas de qualidade de serviço. Dentre elas, optou-se pela adoção dos serviços diferenciados, por meio da criação de uma nuvem *DiffServ*, pois atendia bem o que o problema exigia que era obter uma melhor utilização do enlace contratado de acesso à internet, independente do tipo de aplicação que circula na rede.

Foi proposta uma nova estratégia para tratamento deste tráfego, adotando técnicas de QoS associadas aos medidores que monitoram constantemente os fluxos de tráfego e, baseado na política de priorização, que imediatamente realizam intervenções em busca de um melhor aproveitamento da taxa de vazão rede. As premissas adotadas priorizavam o tráfego para a rede administrativa quando detectado o congestionamento na rede, e liberavam a vazão para perfis não prioritários, quando detectada a ociosidade no meio de transmissão.

Cenários com as mesmas características observadas no estudo foram experimentados, e os resultados obtidos demonstram que por meio do monitoramento constante do tráfego, é possível ampliar em até 33% a utilização do serviço de conexão à internet contratado, sem deixar de garantir a qualidade de serviço aos perfis de tráfego prioritários. Foi desenvolvida uma interface amigável que possibilita aos administradores de redes, manipular as regras de QoS forma mais rápida.

6.1 Trabalhos Futuros

Como o presente trabalho consistiu-se na análise e proposta de nova estratégia para o compartilhamento de taxa de transmissão, pretende-se como trabalhos futuros, migrar a proposta atual para um ambiente de produção que já foi apresentado e aprovado pela gestão do *campus*. Além disso, pretende-se expandir a área de atuação da nuvem *DiffServ* para atender ao controle de tráfego secundário da rede, como laboratórios de informática, rede sem fios, equipe gestora, entre outros perfis de usuários. Pretende-se associar um maior número de técnicas de tratamento de QoS, que além de possibilitar um aproveitamento melhor da taxa de tráfego do enlace

contratado, ofereçam também uma melhor qualidade de experiência (QoE) aos usuários dessas redes. Pretende-se ampliar o desenvolvimento da interface de administração, agregando recursos de gerenciamento mais específicos, inclusive no monitoramento mais aperfeiçoado. Desta forma, esta solução poderá facilmente ser adotada em outras unidades do Intituto Federal de Rondônia ou em ambientes de rede que possuam as mesmas características.

REFERÊNCIAS

AIRHEADS COMMUNITY. *SDN Apps*. 2017. Disponível em: <http://community.arubanetworks.com/t5/SDN-Apps/ct-p/SDN-Apps>. Acesso em: 08/11/2017.

ANGULAR.IO. *AngularJs - One framework. Mobile & desktop.* 2017. Disponível em: <https://angular.io/>. Acesso em: 17/11/2017.

BOLEY, J. M.; JUNG, E. S.; KETTIMUTHU, R. Adaptive QoS for data transfers using software-defined networking. 2017.

COMER, D. E. *Interligação de Redes com TCP/IP*. 5. ed. [S.l.]: Elsevier, 2006. v. 1.

COMER, D. E. *Interligação de Redes com TCP/IP*. 6. ed. [S.l.]: Elsevier, 2015. v. 1. ISBN 9788535278637.

CPQD. *OpenFlow Software Switch 1.3*. 2017. Disponível em: <https://github.com/CPqD/ofsoftswitch13>. Acesso em: 09/08/2017.

CPQD. *Software Switch OpenFlow 1.3*. 2017. Disponível em: <https://github.com/CPqD/ofsoftswitch13>. Acesso em: 10/08/2017.

DAS, T. et al. Insights on SDN migration trajectory. v. 2015-Septe, p. 5348 – 5353, 2015.

ERICSSON TRAFFIC LAB. *OpenFlow Software Switch 1.1*. 2012. Disponível em: <https://github.com/TrafficLab/of11softswitch>. Acesso em: 10/08/2017.

GELBERGER, A.; YEMINI, N.; GILADI, R. Performance analysis of Software-Defined Networking (SDN). p. 389 – 393, 2013.

HONG, C. et al. Achieving high utilization with software-driven WAN. p. 15 –, 2013. Disponível em: <http://dl.acm.org/citation.cfm%3Fdoid%3D2486001.2486012>.

HU, C.; WANG, Q.; DAI, X. SDN over IP: Enabling Internet to Provide Better QoS Guarantee. p. 46 – 51, 2015.

IETF - RFC 2474. *Definição do campo de serviços diferenciados (campo DS) nos cabeçalhos IPv4 e IPv6*. 1998. Disponível em: <https://tools.ietf.org/html/rfc2474>. Acesso em: 12/06/2017.

IETF-RFC 4594. *Diretrizes de Configuração para Classes de Serviços DiffServ*. 2006. Disponível em: <https://tools.ietf.org/html/rfc4594%23section-1.5.2>. Acesso em: 09/06/2017.

IETL - RFC 7426. *Software-Defined Networking (SDN)*: Layers and Architecture Terminology. 2015. Disponível em: <https://tools.ietf.org/html/rfc7426>. Acesso em: 22/07/2017.

IPERF. *iPerf - The ultimate speed test tool for TCP, UDP and SCTP*. 2017. Disponível em: <https://iperf.fr/>. Acesso em: 10/08/2017.

JARSCHEL, M. et al. Interfaces, attributes, and use cases: A compass for SDN. v. 52, n. 6, p. 210 – 217, 2014.

KUBUNTU. *Linux Kubuntu*. 2017. Disponível em: <https://kubuntu.org/>. Acesso em: 10/09/2017.

KUMAR, H. User Control of Quality of Experience in Home Networks using SDN. p. 1 – 6, 2013.

LARA, A.; KOLASANI, A.; RAMAMURTHY, B. Network Innovation using OpenFlow: A Survey. PP, n. 99, p. 1 – 20, 2013. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm%3Farnumber%3D6587999>.

MATERIALIZE. *Um framework front-end moderno e responsivo baseado em Material Design*. 2017. Disponível em: <http://materializecss.com/>. Acesso em: 09/11/2017.

MININET.ORG. *Mininet Overview*. 2017. Disponível em: <https://www.opennetworking.org/platforms/mininet/>. Acesso em: 09/08/2017.

NIPPON TELEGRAPH CORPORATION. *Welcome to RYU the Network Operating System(NOS)*. 2017. Disponível em: <http://ryu.readthedocs.io/en/latest/index.html>. Acesso em: 10/08/2017.

OPEN NETWORKING FOUNDATION. *Especificação OpenFlow 1.3*. 2017. Disponível em: <https://www.opennetworking.org/software-defined-standards/specifications/>. Acesso em: 22/08/2017.

OPEN NETWORKING FOUNDATION. *ONOS Features*. 2017. Disponível em: <https://onosproject.org/features/>. Acesso em: 30/07/2017.

OPEN NETWORKING FOUNDATION. *Open Network Operating System (ONOS)*. 2017. Disponível em: <https://www.opennetworking.org/platforms/onos/>. Acesso em: 09/08/2017.

OPEN NETWORKING FOUNDATION. *Software-Defined Networking (SDN) Definition*. 2017. Disponível em: <https://www.opennetworking.org/sdn-definition/>. Acesso em: 12/06/2017.

OPEN NETWORKING FOUNDATION. *Transforming Networks into Agile Platforms for Service Delivery*. 2017. Disponível em: <https://www.opennetworking.org/sdn-definition/>. Acesso em: 12/06/2017.

OPENDAYLIGHT.ORG. *ODL - Visão geral da Pltforma*. 2017. Disponível em: <https://www.opendaylight.org/what-we-do/odl-platform-overview>. Acesso em: 09/08/2017.

RASTOGI, A.; BAIS, A. Comparative analysis of software defined networking (SDN) controllers - In terms of traffic handling capabilities. p. 1 – 6, 2016.

RFC 7426. *Software-Defined Networking (SDN)*: Layers and Architecture Terminology. 2015. Disponível em: <https://tools.ietf.org/html/rfc7426>. Acesso em: 22/07/2017.

RYU PROJECT TEAM. *RyuBook 1.0*. 2017. Disponível em: <https://osrg.github.io/ryu-book/en/html/rest_qos.html>. Acesso em: 16/03/2017.

SCOTT-HAYWARD, S. Design and deployment of secure , robust , and resilient SDN Controllers. p. 1 – 5, 2015.

SDNHUB. *SDN Hub*. 2017. Disponível em: <http://sdnhub.org/>. Acesso em: 12/06/2017.

VENNE, F. C. Exploring DSCP modification pathologies in mobile edge networks. In: . [S.l.: s.n.], 2017.

VMWARE. *vSphere Hypervisor*. 2017. Disponível em: <https://www.vmware.com/br/products/vsphere-hypervisor.html>. Acesso em: 02/09/2017.

WANG, Y.; ZHANG, Y.; CHEN, J. Pursuing Differentiated Services in a SDN-Based IoT-Oriented Pub/Sub System. n. 2, p. 906 – 909, 2017.

APÊNDICES

APÊNDICE A – GERAÇÃO DE TRÁFEGO NO AMBIENTE SIMULADO

O script a seguir auxiliou na simulação dos cenários, onde ocorria a geração do tráfego de acordo com as repetições definidas. o Exemplo abaixo simula o teste com a geração de tráfego de 50Mbps. Para os outos cenários, foi necessário apenas alterar este parâmetro.

```
#!/bin/bash
for cont in {1..30};
do
echo “medição cenário sem QoS - TCP sessão $cont de 30“;
iperf -c 10.0.0.1 -b 50M -t 10 >> /home/ubuntu/Documents/50-50-33-17-3.log ;
sleep 5;
echo “finalizando coleta $cont”
done
echo “fim da medição do cenário Sem QoS TCP”;
exit;
```

ANEXOS

ANEXO A – AMBIENTE SIMULADO

O código a seguir foi escrito em *python* e tem a finalidade de criar na ferramenta Mininet o ambiente da proposta. Este modelo utilizado está disponível na documentação do Ryu (RYU PROJECT TEAM, 2017).

```
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.topo import Topo
from mininet.node import UserSwitch
from mininet.node import RemoteController

        class SliceableSwitch(UserSwitch):
def __init__(self, name, **kwargs):
UserSwitch.__init__(self, name, '', **kwargs)

        class MyTopo(Topo):
def __init__( self ):
"Create custom topo."
# Initialize topology
Topo.__init__( self )
# Add hosts and switches
host01 = self.addHost('h1')
host02 = self.addHost('h2')
host03 = self.addHost('h3')
switch01 = self.addSwitch('s1')
switch02 = self.addSwitch('s2')
switch03 = self.addSwitch('s3')
# Add links
self.addLink(host01, switch01)
self.addLink(host02, switch02)
self.addLink(host03, switch03)
self.addLink(switch01, switch02)
self.addLink(switch01, switch03)

        def run(net):
s1 = net.getNodeByName('s1')
s1.cmdPrint('dpctl unix:/tmp/s1 queue-mod 1 1 80')
s1.cmdPrint('dpctl unix:/tmp/s1 queue-mod 1 2 120')
s1.cmdPrint('dpctl unix:/tmp/s1 queue-mod 1 3 800')

        def genericTest(topo):
```

```
net = Mininet(topo=topo, switch=SliceableSwitch,
controller=RemoteController)
net.start()
run(net)
CLI(net)
net.stop()
        def main():
topo = MyTopo()
genericTest(topo)
        if __name__ == '__main__':
main()
```

ANEXO B – CONFIGURAÇÕES DO SWITCH 01 - NÚCLEO

Configura o controlador para direcionar o pacote entrante na **porta 2** do switch para a **Fila 1** caso seu campo **DSCP seja 0**.

curl -X POST -d '{"match": {"ip_dscp": "0", "in_port": "2"}, "actions":{"queue": "1"}}' http://localhost:8080/qos/rules/0000000000000001

Configura o controlador para direcionar o pacote entrante na **porta 2** do switch para a **Fila 2** caso seu campo **DSCP seja 12**.

curl -X POST -d '{"match": {"ip_dscp": "12", "in_port": "2"}, "actions":{"queue": "2"}}' http://localhost:8080/qos/rules/0000000000000001

Configura o controlador para direcionar o pacote entrante na **porta 2** do switch para a **Fila 3** caso seu campo **DSCP seja 10**.

curl -X POST -d '{"match": {"ip_dscp": "10", "in_port": "2"}, "actions":{"queue": "3"}}' http://localhost:8080/qos/rules/0000000000000001

Configura o controlador para direcionar o pacote entrante na **porta 3** do switch para a **Fila 1** caso seu campo **DSCP seja 0**.

curl -X POST -d '{"match": {"ip_dscp": "0", "in_port": "3"}, "actions":{"queue": "1"}}' http://localhost:8080/qos/rules/0000000000000001

Configura o controlador para direcionar o pacote entrante na **porta 3** do switch para a **Fila 1** caso seu campo **DSCP seja 0**.

curl -X POST -d '{"match": {"ip_dscp": "10", "in_port": "3"}, "actions":{"queue": "3"}}' http://localhost:8080/qos/rules/0000000000000001

Configura o controlador para direcionar o pacote entrante na **porta 3** do switch para a **Fila 1** caso seu campo **DSCP seja 0**.

curl -X POST -d '{"match": {"ip_dscp": "12", "in_port": "3"}, "actions":{"queue": "2"}}' http://localhost:8080/qos/rules/0000000000000001

ANEXO C – CONFIGURAÇÃO DO SWITCH 02 - ACADÊMICO

Instala as entradas de medição no **switch 2.**

curl -X POST -d '{"match": {"ip_dscp": "10"}, "actions":{"meter": "1"}}' http://localhost:8080/qos/rules/0000000000000002

Comando para **remarcar campo DSCP** quando medição atingir **17 Mbps.**

curl -X POST -d '{"meter_id": "1", "flags": "MBPS", "bands":[{"type":"DSCP_REMARK", "rate": "17", "prec_level": "1"}]}' http://localhost:8080/qos/meter/0000000000000002

ANEXO D – CONFIGURAÇÃO DO SWITCH 03 - ADMINISTRATIVO

Instala as entradas de medição no **switch 3.**

curl -X POST -d '{"match": {"ip_dscp": "10"}, "actions":{"meter": "1"}}' http://localhost:8080/qos/rules/0000000000000003

Comando para **remarcar campo DSCP** quando medição atingir **33 Mbps.**

curl -X POST -d '{"meter_id": "1", "flags": "MBPS", "bands":[{"type":"DSCP_REMARK", "rate": "33", "prec_level": "1"}]}' http://localhost:8080/qos/meter/0000000000000003