# Analisando métodos e oportunidades em redes definidas por software (SDN) para otimizações de tráfego de dados

Leandro Andrioli <sup>1</sup> Rodrigo da Rosa Righi <sup>2</sup> Mateus Rauback Aubin <sup>3</sup>

Resumo: Redes de computadores são dinâmicas e requerem constante atualização e monitoramento de operações para atender ao crescente volume de dados trafegados. Isso gera diversos problemas de custo, bem como no gerenciamento e ajuste de desempenho para oferecer uma qualidade de serviço (QoS) granular, balancear a carga dos dados e controlar a ocorrência de congestionamentos. Como alternativa, um novo paradigma de rede programáveis vem sendo utilizado sob o nome de Redes Definidas por Software (SDN). A SDN consiste no desacoplamento do plano de dados e controle da rede, onde um controlador programável é responsável por gerenciar regras para o encaminhamento dos dados para diversos dispositivos. Assim, o hardware que fica no fluxo de dados da rede simplesmente endereça o encaminhamento dos pacotes de forma rápida de acordo com essas regras. Nesse contexto, esse artigo conduz um estudo sobre diferentes métodos e abordagens que estão sendo utilizados na literatura para solucionar problemas na otimização do tráfego de dados na rede através do uso de SDN. Em particular, o presente estudo se diferencia de outras revisões sobre SDN pois foca em questões como QoS, balanceamento de carga e controle de congestionamento. Por fim, além da revisão do estrado-da-arte de SDN perante as áreas mencionadas, apresenta-se também um levantamento de desafios futuros e oportunidades de pesquisa na área.

**Palavras-chave:** Adaptabilidade. Balanceamento de carga. Controle de congestionamento. Qualidade de serviço. Openflow. Redes definidas por software.

Abstract: Computer networks are dynamic and require constant updating and monitoring of operations to meet the growing volume of transmited data. This generates a number of cost issues as well as performance management and tuning to provide a granular quality of service (QoS), data load balancing and bottleneck control. As an alternative, a new paradigm of programable network has been used under the name of Software-Defined Networks (SDN). SDN consists in the decoupling of the data and control planes in the network, where a programmable controller is responsible for managing routing rules to transfer data between different network devices. Thus, the hardware which controls the network's data plane simply addresses the forwarding of the packets according to these rules. In this context, this article conducts a study on different methods and approaches that are being used in the literature to solve problems in the optimization of data traffic in the network through the use of SDN. In particular, this study differs from other reviews of SDN because it focuses on issues such as QoS, load balancing, and congestion control. Lastly, in addition to the state-of-the-art review of these areas, a discussion about future research challenges and opportunities in the SDN area is also presented.

**Keywords:** Adaptability. Load balancing. Congestion control. Quality of service. Openflow. Software-defined networks.

http://dx.doi.org/10.5335/rbca.v9i4.6948

 $<sup>^1\</sup>mathrm{Universidade}$  do Vale do Rio dos Sinos - Unisinos, São Leopoldo (RS), Brasil leandro.andrioli@outlook.com

 $<sup>^2 \</sup>text{Universidade}$  do Vale do Rio dos Sinos - Unisinos, São Leopoldo (RS), Brasil rrighi@unisinos.br

<sup>&</sup>lt;sup>3</sup>Universidade do Vale do Rio dos Sinos - Unisinos, São Leopoldo (RS), Brasil maubin@edu.unisinos.br

## 1 Introdução

A capacidade das redes de computadores, aliadas a problemas de infraestrutura, e o crescente volume de tráfego de dados são limitantes hoje em dia para a execução de serviços escaláveis e que tenham a necessidade de alta disponibilidade, bem como desempenho adequado. Essa quantidade de dados é proveniente de diversas fontes, como o grande número de usuários que enviam e recebem informações, sensores inteligentes e aplicações que realizam transmissões de grandes volumes de dados [1]. As redes convencionais utilizam algoritmos especiais implementados em componentes dedicados para controlar e monitorar o fluxo de dados na rede, gerenciando o encaminhamento dos pacotes e reavaliando, em intervalos regulares, o estado da ligação entre os dispositivos da rede.

Na rede convencional, no momento da recepção de um pacote por um dispositivo de rede, seja ele um roteador ou chaveador, é utilizado um conjunto de regras incorporado no seu próprio *firmware* com o objetivo de encontrar o dispositivo de destino para otimizar o percurso de encaminhamento do pacote [2]. Esses dispositivos de rede utilizam diversos protocolos complexos e custosos em termos de gerenciamento e tempo, onde os administradores de rede, por sua vez, são responsáveis pela configuração de políticas para responder a uma grande variedade de eventos da rede e das aplicações envolvidas. Em particular, os administradores precisam transformar, manualmente, as políticas de alto nível em comandos de configuração de baixo nível enquanto se adaptam às mudanças nas condições da rede. Muitas vezes, é necessário realizar essas tarefas através ferramentas proprietárias e limitadas quanto a funcionalidades.

O gerenciamento da rede e ajuste fino de desempenho é bastante desafiador no cenário tradicional, tornandose também propenso a erros devido a má interpretação humana a respeito de eventos da rede[3]. Nesse contexto, as ideias vindas de programabilidade do paradigma de redes definidas por software (SDN, do inglês Software-Defined Network) ressurgem com a popularização do protocolo OpenFlow, eliminando a rigidez presente nas redes tradicionais. Com ela ocorre um desacoplamento das camadas de controle e dados, onde todo o gerenciamento da rede se torna programável em um controlador, enquanto cabe ao hardware de rede (chaveadores ou roteadores, por exemplo) simplesmente a tarefa de encaminhamento de pacotes de acordo com as regras que estão estabelecidas pelo software [4].

Uma SDN permite que o comportamento da rede seja mais flexível, bem como adaptável e gerenciável conforme às necessidades de cada contexto de negócio. Como comentado, OpenFlow é um protocolo padronizado para comunicação entre chaveadores e o controlador SDN. Foi proposto como uma forma da comunidade científica testar novas ideias e experimentos. Inicialmente o OpenFlow foi implantado na rede de um campus acadêmico [5], mas hoje em dia é utilizado em diversas universidades dos EUA e Europa [6]. Além disso, a industria também adotou esse protocolo nas suas implementações de SDN como uma estratégia para aumentar as funcionalidades na rede e reduzir os custos e a complexidade do hardware de rede.

Diversas revisões da literatura foram realizadas com foco em SDN. Em sua maioria, o foco das pesquisas é voltado para: (I) conceitos, desafios e oportunidades futuras para SDN [1, 2, 7]; (II) redes programáveis [3]; (III) inovações no próprio protocolo Openflow [4]; (IV) taxonomia para classificação das revisões da literatura em SDN [8]; (V) soluções e problemas causados ao atualizar fluxos e regras na rede SDN [9, 10]. Diferente dos trabalhos supracitados, o presente artigo apresenta uma revisão de literatura referente a métodos e abordagens que estão sendo utilizados pelos pesquisadores para solucionar problemas de otimização do tráfego de dados nas redes através do uso de SDN. A expressão otimização de tráfego é observada perante a análise de qualidade de serviço (QoS), balanceamento de carga e controle de congestionamento. Neste sentido, a contribuição do artigo aparece na discussão do estado-da-arte de SDN na visão dessas três áreas e numa análise de desafios de pesquisa e oportunidades de SDN com foco em otimização e desempenho de rede.

O presente artigo está organizado em 7 seções. Após a introdução, a Seção 2 expõem os critérios para escolha dos artigos utilizados na presente revisão de literatura. Na seção 3, apresenta-se o conceito e arquitetura de SDN, explorando também características e a arquitetura do protocolo OpenFlow. A seção 4 mostra pesquisas recentes relacionadas aos métodos e abordagens para otimização do tráfego na rede usando conceitos de SDN. Uma análise de desafios e oportunidades de pesquisa na área são apresentadas na Seção 5. Por fim, a Seção 6 reapresenta as principais conclusões da pesquisa, salientando também a sua contribuição.

## 2 Critérios para escolha de artigos

A fim de selecionar os artigos para a presente revisão de literatura, foi realizada uma análise a qual foi conduzida respondendo perguntas relacionadas com otimização do tráfego na rede através do uso de SDN da seguinte forma: (I) Como redes definidas por software são definidas atualmente? (II) Como redes definidas por software estão sendo aplicadas para otimização do tráfego na rede? (III) Quais as técnicas e abordagens que vêm sendo utilizadas para controlar QoS, balancear a carga e evitar congestionamentos na rede através do uso de SDN? (IV) Quais são os principais desafios e oportunidades em aberto a serem enfrentados e superados na utilização de redes definidas por software para otimizar as operações na rede?

Com base nessas questões da análise, definiram-se os termos mais pertinentes de pesquisa. São eles: (I) software-defined networking ou SDN; (II) quality of service ou QoS; (III) load balancing; (IV) congestion control ou congestion avoidence. Os termos de pesquisa foram escolhidos seguindo as seguintes etapas [11]:

- Definir os principais termos de pesquisa;
- Identificar sinônimos alternativos relacionados com os termos principais;
- Utilizar os operadores OR para incorporar sinônimos alternativos e AND para vincular os termos principais.

As bases de dados utilizadas como fonte da pesquisa foram: (I) Google Scholar; (II) IEEE Xplore e; (III) ACM Digital Library. O processo de filtragem de artigos adotou os seguintes critérios: (I) ano de publicação: adotou-se um intervalo de 3 anos para a coleta de material, ou seja, de 2014 a 2016; (II) tipo de publicação: revista científica ou anais de conferências; (III) contexto de publicação: O contexto da pesquisa foi levado em conta para a inclusão de itens nesta pesquisa, onde optou-se por pesquisa acadêmica; (IV) idioma: Apenas artigos na língua inglesa. A pesquisa inicial incluiu artigos dos últimos três anos, e em seguida, foram utilizadas as referências para a revisão do assunto e para determinar os desafios e oportunidades relacionadas a SDN e otimização do tráfego de dados. No início da pesquisa foram encontrados 694 trabalhos e após as etapas de remoção de impurezas conforme os critérios supracitados, remoção de artigos duplicados, filtro por título, filtro por abstract, filtro por introdução e filtro por leitura completa, chegamos aos 24 artigos selecionados, conforme pode ser observados na Tabela 1.

## 3 Redes definidas por Software

Redes definidas por software, também conhecidas por SDN, são um tópico de pesquisa recente e que trás consigo um novo paradigma para redes quando comparado ao modelo de redes tradicional [3]. SDN desacopla as camadas de controle e dados. Com isso, o componente de software fica responsável por todo o plano de controle da rede, enquanto cabe ao hardware apenas a tarefa de encaminhamento de pacotes de acordo com as regras que estão estabelecidas pelo software [4]. Dessa maneira, SDN oferece uma ampla rede de abstração que simplifica o gerenciamento da rede e reduz os custos e a complexidade de hardware [4] [3]. Nesse alto nível de abstração, há um ambiente no qual toda a engenharia de tráfego, roteamento de dados e controle de acesso podem ser implementados de forma rápida e eficiente, sem a necessidade da criação de políticas de baixo nível [28]. Como consequência, a complexidade dos dispositivos da rede é reduzida, uma vez que apenas uma simples interface é exposta para o administrador realizar as configurações, liberando-o de atuar manualmente em dispositivos de diferentes fornecedores [3]. A Figura 1 apresenta uma comparação entre uma arquitetura de rede tradicional frente uma arquitetura de rede SDN.

A arquitetura padrão SDN é composta por três camadas principais: (I) Camada de Aplicação; (II) Camada de Controle; (III) Camada de Infraestrutura. A camada de infraestrutura é formada por um conjunto de elementos da rede, como por exemplo switches e roteadores, que juntos representam o plano de dados com a responsabilidade de encaminhamento de pacotes conforme as instruções do plano de controle [25]. A camada central, conhecida por camada de controle, é o *core* da rede definida por software, a qual é formada por um controlador centralizado que permite que as funções da rede sejam programáveis de uma maneira rápida e fácil. Esse controlador oferece uma visão global de toda a infraestrutura da rede para a camada de aplicação e utiliza a *southbound interface* para configurar o encaminhamento de dados nos switches da rede através da instalação de regras em cada um dos

Tabela 1: Trabalhos analisados para a presente revisão de literatura.

Artigo	Modalidade
An sdn-based multipath qos solution [12]	Qualidade de Serviço
Qamo-sdn: Qos aware multipath tcp for software defined optical networks [13]	Qualidade de Serviço
Experimental demonstration of latency-aware software defined networking for	Qualidade de Serviço
networking for openflow-based intra-datacenter optical interconnect	
networks [14]	
Minimizing latency of critical traffic through sdn [15]	Qualidade de Serviço
Application-oriented bandwidth and latency aware routing with openflow	Qualidade de Serviço
network [16]	
An adaptive routing scheme for heterogeneous data-flows using openflow [17]	Qualidade de Serviço
Congestion control using openflow in software defined data center	Controle de Congestionamento
networks [18]	
A joint approach to multipath routing and rate adaptation for congestion	Controle de Congestionamento
control in openflow software defined network [19]	
A congestion avoidance algorithm in sdn environment [20]	Controle de Congestionamento
Otcp: Sdn-managed congestion control for data center networks [21]	Controle de Congestionamento
A novel proposal to effectively combine multipath data forwarding for	Controle de Congestionamento
data center networks with congestion control and load balancing using	-
software-defined networking approach [22]	
An openflow-based elastic solution for cloud-cdn video streaming service [23]	Balanceamento de Carga
Load balancing in openflow-enabled switches for wireless access traffic	Balanceamento de Carga
aggregation [24]	_
Pushing sdn to the end-host, network load balancing using openflow [25]	Balanceamento de Carga
Minimum cost load balanced multipath routing protocol for low power	Balanceamento de Carga
and lossy networks [26]	_
Optimized congestion aware energy efficient traffic load balancing	Balanceamento de Carga
scheme for routing in wireless sensor networks [27]	_
Software-defined networking: Challenges and research opportunities for	Survey
future internet [1]	
A survey on software-defined network and openflow: From concept to	Survey
implementation [2]	
A survey on software-defined networking [7]	Survey
A survey of software-defined networking: Past, present, and future of	Survey
programmable networks [3]	
Network innovation using openflow: A survey [4]	Survey
A survey and a layered taxonomy of software-defined networking [8]	Survey
Rules placement problem in openflow networks: A survey [9]	Survey
The problems and solutions of network update in sdn: A survey [10]	Survey

Plano de Controle
Plano de Dados

Plano de Dados

(a)

Plano de Controle
Plano de Dados

(b)

Plano de Dados

Plano de Dados

Plano de Dados

Plano de Dados

Figura 1: Comparação entre as arquiteturas de (a) rede tradicional; (b) rede SDN

switches [25] [2]. Atualmente, o protocolo predominante para esse canal de comunicação entre o controlador e os componentes da rede é o OpenFlow [29]. Na seção 4 será provido detalhes sobre o protocolo OpenFlow.

Por fim, no topo das camadas, encontra-se a camada de aplicação, a qual é formada pelas aplicações e serviços da rede. Considera-se aplicações e serviços da rede, módulos que de alguma forma implementam uma lógica para funcionalidades de engenharia de tráfego, roteamento de dados, balanceamento de carga, tolerância a falhas, etc [25]. Na Figura 2 podemos observar a organização das três camadas principais da SDN. Como resultado final, a SDN resulta em melhorias na rede relacionadas a otimização de desempenho, controle e gerenciamento de políticas granulares e simplificação no provisionamento dos recursos. Nesse modelo, os administradores têm o poder de controlar todo o fluxo de dados, alterando, em tempo de execução, a forma de tratamento e roteamento dos dados nos switches a partir de um local centralizado, através de um módulo de aplicação que se comunica com o controlador, sem a necessidade de configurar cada um dos dispositivos manualmente [30].

## 3.1 Modelos de Design

SDN suporta diferentes modelos, os quais possuem arquiteturas distintas, requisitos e formas de controle. Podemos citar como principais os modelos: (I) centralizado x distribuído; e (II) políticas reativas x proativas, que serão detalhados agora [3].

#### 3.1.1 Centralizado x Distribuído

A SDN suporta tanto o modelo centralizado, quando o distribuído. O modelo centralizado é baseado em um único controlador central que tem uma visão global de todos os elementos da rede, realizando todo o gerenciamento e controle sobre eles [1]. Com isso, toda a inteligência e lógica fica centralizada em um único ponto. Esse modelo é suportado pelo *Open Network Fundation*, e o OpenFlow é o principal protocolo para o gerenciamento nessa formato de administração [30].

Aplicações de Negócios

API
API
Camada de Controle

Controladora

Interface de Controle
do Plano de Dados

Camada de Infraestrutura

Dispositivo de Rede

Dispositivo de Rede

Dispositivo de Rede

Figura 2: Principais camadas da arquitetura SDN.

Por ser centralizado, tem como principais vantagens uma menor complexidade de implantação, um ponto único de gerenciamento e melhor controle sobre os estados da rede, devido ao conhecimento global da rede. Porém, essa modalidade também possui pontos negativos. O primeiro é em relação a escalabilidade, onde todas funcionalidades encontram-se em um único ponto que requer maior poder computacional, consumo de energia e que consequentemente torna o armazenamento de regras limitado [1, 9]. Ainda, a constante busca por elementos da rede e suas estatísticas produzem uma sobrecarga no controlador, o que aumenta os tempos de resposta. Além disso, outro ponto de preocupação é relacionado a tolerância a falhas, que se torna um problema ao se ter um único controlador como ponto único de falha. Nesse aspecto, de questão de tolerância a falhas, o protocolo Open-Flow, em versões mais recentes, permite a conexão de múltiplos controladores no mesmo switch, o que oferece a possibilidade de acionar um controlador de backup em caso de falhas [3].

Já no modelo distribuído, elimina-se o único ponto de falha, pois o controle e gerenciamento passa a ser realizado através diferentes controladores. Isso faz com que o modelo distribuído se torne mais robusto, escalável e com capacidade de reagir rapidamente aos eventos dos rede. Apesar dessas vantagens, a complexidade de implantação nesse modelo se torna maior, pois é difícil obter uma visão global da rede e manter os controladores sincronizados. Existem grandes desafios relacionados a encontrar formas de sincronização dos eventos locais e distribuídos para prover a visão global da rede aos controladores, principalmente devido ao *overhead* de comunicações [1]. Atualmente, entre as iniciativas que procuram encontrar soluções para se ter uma lógica centralizada com um plano de controle distribuído destacam-se as pesquisas realizadas por [31, 32].

#### 3.1.2 Políticas Reativas x Proativas

No controle de políticas reativas, os elementos da rede responsáveis pelo encaminhamento de pacotes, consultam o controlador a cada decisão que precisa ser tomada. Isso acontece por exemplo na chegada de um pacote relativo a um fluxo ainda não mapeado no switch. Isso faz com que o atraso aumente para cada primeiro pacote de novos fluxos, pois no momento da chegada é necessário que o pacote seja encaminhado para o controlador realizar a tomada de decisão/ação necessária. Após esse processo, os próximos pacotes desse fluxo podem seguir a mesma ação sem a necessidade de consultar o controlador. Nesse modelo, em alguns casos, não ocorre um grande impacto, porém, em grandes redes geograficamente distribuídas, isso pode gerar um impacto negativo considerável [3]

Em contra partida, no modelo proativo, o controlador instala nos switches as regras necessárias, evitando que os equipamentos realizem consultas frequentes no controlador durante a descoberta de novos fluxos [3]. Logo, limita-se a fazer uma pesquisa apenas na tabela de fluxo local do switch. Como consequência, a forma proativa

OpenFlow Switch
Canal Protocolo OpenFlow Seguro
SSL
hw Tabela de Fluxos

Figura 3: Arquitetura com os principais componentes do OpenFlow [5].

elimina qualquer latência induzida pela consulta de um controlador em cada fluxo.

# 4 ONF e o Protocolo OpenFlow

A Open Network Foundation (ONF) é uma organização fundada em 2011 que dedica-se à promoção da adoção de redes definidas por software (SDN) através do desenvolvimento de padrões abertos [30]. Como uma forma de padronização para a comunicação entre o plano de controle e plano de dados, a ONF propôs o protocolo OpenFlow, o qual se tornou o primeiro protocolo para este fim a ser utilizado em SDN [19]. Trata-se de um protocolo aberto que permite a implementação dos conceitos de SDN para o hardware e software [2]. Atualmente, está entre os mais populares protocolos para essa finalidade e de forma geral descreve como aplicações podem programar a tabela de fluxos de diferentes switches [4] [5].

Como pode-se observar na Figura 3, a arquitetura do OpenFlow é formada por três componentes principais [2, 29, 4]: (I) switches: o Openflow gerencia e infere na tabela de fluxo de diferentes switches e roteadores. Essa tabela de fluxos é uma lista de entrada de fluxos, os quais possuem um campo de correspondência, contadores e instruções/ações; (II) controlador: é um programa responsável por manipular a tabela de fluxo do dispositivo de rede através do protocolo OpenFlow, onde realiza operações de atualização, inclusão ou remoção de regras fluxos. Esse processo pode ser estático, onde o controlador é uma unidade de software simples rodando em um computador que estabelece regras estáticas durante a execução, ou dinâmico, onde o estabelecimento de regras para o caminho dos pacotes é alterado durante a execução; (III) canal seguro: é uma interface que conecta o controlador com todos os switches da rede. É através desse canal que o controlador realiza o gerenciamento, que inclui o recebimento e envio de pacotes de controle entre os switches envolvidos.

Quando um pacote chega no switch, ele é computado *on-the-fly* e comparado com o campo de correspondência de cada uma das regras presentes na tabela de fluxos desse switch [3]. Se ocorrer correspondência em alguma das regras, o pacote é processado de acordo com a ação especificada para a regra em questão. Caso contrário, o pacote é encapsulado e enviado para o controlador decidir qual ação deve ser tomada [4]. A grande maioria dos switches com OpenFlow tem suporte a, pelo menos, as seguintes regras: (I) enviar o pacote para determinada porta; (II) encapsular e enviar o pacote para o controlador; (III) descartar o pacote.

Atualmente, diferentes versões de especificação do OpenFlow estão disponíveis. A primeira versão foi a 0.2.0 lançada em março de 2008. Posteriormente foram lançadas outras versões até chegar na 1.0.0 [33] em dezembro de 2009, que se tornou a mais implantada. Após essa versão, ainda foram lançadas as versões 1.1.0 [34], 1.2 [35], 1.3 [36], 1.4 [37], até chegar na versão corrente 1.5 [38].

## 5 Otimização do Tráfego da Rede

É sabido que o tráfego de dados na Internet encontra-se numa forte crescente, e esse volume de informações é proveniente de diversas fontes, tais como usuários da Internet, sensores inteligentes e dispositivos móveis [39]. As redes de comunicação acabam sofrendo com sua inflexibilidade e incapacidade de se adaptar às diferentes exigências de tráfego transmitido [40]. É necessário a existência de estratégias eficientes capazes de gerenciar e tratar esse volume de informações de forma otimizada e eficiente. Nesse contexto, a SDN é um promissor e poderoso conceito que introduz novas dimensões de flexibilidade e adaptabilidade nas redes de comunicação [40]. Ela faz com que a rede se torne programável, oferecendo o controle e a visão centralizada de toda a rede, criando oportunidades para a otimização do tráfego da rede que pode ser extraído através do controle de congestionamento, gerenciamento de qualidade de serviço e balanceamento da carga com um roteamento de dados adaptativo e inteligente para engenharia de tráfego.

## 5.1 SDN aplicada na QoS

A Qualidade de Serviço (QoS) fornece garantias de desempenho para aplicações através de métricas como largura de banda, atraso e a perda de pacotes [12]. De forma geral, a ideia é ter a capacidade para fornecer prioridades diferentes para as aplicações, garantindo um determinado nível de desempenho a um fluxo de dados. Nos dias de hoje, as arquiteturas de Internet oferecem apenas o serviço *best effort*, portanto não é possível garantir QoS para aplicações [12]. Iniciativas como a da Internet Engineering Task Force (IETF), têm explorado arquiteturas QoS como IntServ [1] e DiffServ [2], porém não tiveram sucesso na sua adoção. Um dos principais motivos da falta de sucesso dessas arquiteturas é a ausência de conhecimento relativo ao estado global da rede [2]. Em contrapartida, com o protocolo OpenFlow, é possível oferecer um suporte granular na QoS, permitindo especificar a forma de lidar com fluxos individualizados, considerando as diferentes métricas da rede.

Em [13] é apresentada a arquitetura QAMO-SDN para redes de fibra óptica empregando o novo protocolo de transporte MPTCP com um algoritmo de provisionamento de QoS para SDN em datacenters na nuvem. A QAMO-SDN tem capacidade de ser adaptativo e auto configurável de forma dinâmica com base na situação corrente da rede. Em [12], os autores apresentam o projeto de HiQoS, uma proposta que oferece garantias de utilização da largura de banda através do uso de SDN. HiQoS utiliza diferentes rotas entre uma fonte e destino, além de um mecanismo de filas para garantir a largura de banda para diferentes tipos de tráfego. Quando o tráfego de dados de determinado tipo de aplicação é maior que a largura da banda alocada para a fila, um novo caminho menos utilizado é adicionado para o tráfego dos dados. Resultados constatam que HiQoS reduz atrasos de transmissão dos dados e o tempo de resposta de servidores, aumentando assim a taxa de transferência do sistema.

Em [14] é proposta a arquitetura LaSDN. Uma arquitetura ciente da latência que utiliza o OpenFlow em redes intra-datacenter. A arquitetura é capaz de organizar os serviços com sensibilidade na latência corrente ao introduzir uma estratégia de agendamento de serviço ciente da latência (LaSS). Dessa forma, LaSDN organiza e acomoda as aplicações com a QoS necessária considerando o fator de latência e oferecendo maior disponibilidade e capacidade de resposta para as demandas intra-datacenter. Na proposta de [15], a latência também é utilizada como métrica para gerenciar o atraso do tráfego de dados importantes. Uma solução que realiza análises periódicas para encontrar um caminho com o menor atraso e realizar o encaminhamento dos dados ponto a ponto através desse caminho é proposto. Os autores de [16] propõem um roteamento ciente da latência e da largura da banda para diferentes serviços. A ideia é alocar rotas distintas para dois tipos de aplicações: aplicações dependentes da largura da banda e aplicações dependentes da latência. Eles oferecem uma API onde a aplicações comunica com o controlador para registrar a necessidade de qual o tipo de rotas. Com base nessa preferência, o controlador insere os fluxos adequados nos respectivos switches.

Posteriormente, em [17] foi proposto um controlador que utiliza OpenFlow e leva em conta diferentes necessidades além da latência e largura da banda, como confiabilidade e jitter. Nessa proposta, as aplicações enviam o perfil que desejam em um campo do cabeçalho do protocolo OpenFlow. Esse perfil se refere a uma combinação de diferentes pesos a serem considerados para largura de banda, latência e perda de pacotes. Após a geração dos pesos para cada caminho, com base nesses parâmetros, utiliza-se o algorítmo de Dijkstra para gerar o caminho de menor custo e então atualizar os fluxos nos switches necessários. Tanto em [16] e [17] a análise de caminhos é calculada e alterada de forma completa, ou seja, não ocorre uma análise e alteração individualizada dos links para cada caminho.

#### 5.2 SDN aplicada no controle de congestionamento

O controle de congestionamento nas redes é uma tarefa muito importante. Congestionamentos na rede ocorrem quando muitos pacotes estão trafegando em uma mesma parte da sub-rede, sendo superior a sua capacidade. Como consequência ocorrem atraso e perda de pacotes, levando a uma queda acentuada de desempenho. Um controle de congestionamento deve reduzir o atraso de transmissão e consumo de largura de banda, fazendo com que ocorra uma otimização para a taxa de transferência e consequentemente a performance da rede [18].

[18] apresentam um método para controlar o congestionamento em data centers e, dessa maneira, otimizar a performance da rede. No método proposto, o reconhecimento de congestionamentos nos links é descoberto através da constante verificação de estatísticas das portas dos switches. Ao verificar um fluxo com mais de 70% de uso, com base nas estatísticas correntes da rede, é iniciado o recálculo de um novo caminho com mais recursos livres. De forma similar, [19] propõem um controle de congestionamento utilizando Openflow com uma combinação de diferentes rotas aliada com um recurso para adaptação da taxa de entrega. Pode-se concluir que essa combinação produz uma melhor performance que o modelo tradicional de único caminho e o modelo de diferentes rotas sem adaptação da taxa de entrega. Os autores de [20] propõem um algoritmo para controlar fluxos de congestionamentos em redes SDN. Visa-se prever e inferir na rede antes da ocorrência de perda de pacotes. Para esse propósito, ao atingir 70% do uso da rede a proposta recalcula uma nova rota para o tráfego dos dados com base na utilização das portas do switch. Observa-se que a solução é capaz de reduzir os congestionamentos em ambientes de alto congestionamento. Com isso, seria possível substituir o *slow start* do algoritmo de controle de congestionamento TCP pelo algoritmo proposto.

O Omniscient TCP (OTCP), apresentado por [21], é uma abordagem que utiliza a SDN para computar parâmetros de controle de congestionamento específicos do ambiente baseado nas propriedades disponíveis da rede que se encontram no controlador. Voltado para rede interna de data centers, o objetivo é abordar as deficiências de TCP sob cargas de trabalho agregadas, alcançando alta taxa de transferência e latência baixa e estável. Para isso, ocorre configurações dos parâmetros de congestionamento do TCP para cada rota ponto a ponto considerando a topologia, latência, taxa de transferência e o buffer. Em [22] é apresentado um método que visa integrar o balanceamento de carga dinâmico e o esquema de múltiplos caminhos com o controle de congestionamento através do uso da pura abordagem de SDN aplicada para redes de data centers. Embora alguns cenários não tenham sido testados, pode-se observar que a abordagem tem certas vantagens e melhorias em relação às abordagens existentes. Com a introdução do SDN, o algoritmo dinâmico proposto, pode atualizar e reagir prontamente ao desequilíbrio de carga e ao congestionamento de tráfego.

## 5.3 SDN aplicada para balanceamento de carga

Em redes tradicionais, a melhor forma de realizar o balanceamento da carga de trabalho é um questão muito importante. Hoje, servidores de balanceamento de carga são usados para satisfazer a demanda de grande volume de dados, porém essa técnica necessita de grande capital de investimento e oferece baixa escalabilidade e flexibilidade. Isso acaba dificultando o suporte a demandas com carga de trabalho altamente dinâmicas [41].

Em [23] é sugerida uma abordagem elástica para gerenciar um serviço de transmissão de vídeo em diferentes servidores que são executados na nuvem. A estratégia emprega balanceamento de carga baseado no OpenFlow junto a um mecanismo de gerenciamento elástico responsável por aumentar e diminuir a quantidade de servidores de transmissão ativos através de uma estratégia de elasticidade horizontal. Conforme a demanda corrente de usuários, utilização de recursos (uso de memória, CPU e taxa de transferência) e as limiares das políticas de elasticidade, o mecanismo aumenta ou diminui os servidores de transmissão na nuvem e recalcula o direcionamento do tráfego de dados através dos recursos do OpenFlow.

Os autores de [24] propõem um algoritmo de balanceamento de carga para o controlador OpenDaylight [42] focando no gerenciamento de carga nas portas do switch. O algoritmo coleta as estatísticas de carga de tráfego das portas do switch e, no momento em que a quantidade de bytes excede um determinado limiar, ele gera um novo fluxo para ser instalado na tabela de fluxo do switch. Dessa forma ocorre um balanceamento de carga entre as portas que tem acesso ao *backhaul*. Em [25], os autores apresentam abordagens para levar SDN a um dispositivo final (computador, smartphone, etc.) considerando o cenário de balanceamento de carga através de múltiplas interfaces de rede. Tem como diferencial a ideia de levar o balanceamento de carga para o lado do cliente, através

da instalação e configuração o protocolo OpenFlow e um controlador SDN no aparelho do cliente.

[26] propõem um algoritmo proativo para roteamento com diferentes caminhos chamado de MRPL. O método visa alcançar um balanceamento de carga através de um programa linear que utiliza heurística para implementá-lo de forma distribuída. Como resultado, é possível equilibrar o consumo de energia, maximizar o tempo de vida da rede, e minimizar o custo total da transmissão para redes de baixo consumo de energia. Também relacionado a eficiência energética, [27] apresenta o esquema OCAEE-LB para balanceamento de carga e redução de congestionamentos ciente da eficiência energética para roteamento de dados em redes de sensores sem fio. O esquema utiliza as informações adquiridas durante o processo de descoberta de rotas e considera uma métrica de roteamento composto para determinar o status de congestionamento de um nó para aplicar ou não o balanceamento de carga de tráfego.

# 6 Desafios e oportunidades de pesquisa

A utilização de redes SDN vem crescendo não só em aplicações de pesquisa, mas também na indústria. Existem diversas oportunidades ainda em aberto para otimização do tráfego na rede através do uso de SDN. Dentre alguns destes desafios, pode-se destacar principalmente os relacionados à escalabilidade, disponibilidade, adaptabilidade e confiabilidade. Conforme a rede SDN expande, os controladores de rede se tornam um gargalo no que diz respeito ao desempenho devido à grande quantidade de mensagens de sinalização recebidas e pedidos de encaminhamento. Para solucionar esse e outros problemas relacionados a escalabilidade, pode-se utilizar uma arquitetura de controle descentralizada com planos de controle integralmente distribuídos. Arquiteturas que consideram as formas de comunicação entre os controladores e a utilização de estratégias de programação que empreguem algoritmos para evitar colisões nas mensagens são fundamentais. Outra possibilidade, é utilizar um modelo de hierarquia para coordenar os controladores, no qual quanto mais próximo da raiz, mais decisões globais são tomadas [2].

SDN oferece um conhecimento global dos dados correntes na topologia da rede. Nesse contexto, é possível aplicar técnicas de aprendizado de máquina para a rede se tornar mais inteligente e auto adaptativa. Decisões podem ser tomadas com base no tipo e característica do tráfego que é recebido. Dessa forma, o desenvolvimento de algoritmos de aprendizado de máquina otimiza os recursos, aumenta a performance e segurança da rede juntando o mundo das redes com a inteligência artificial. Com o crescimento da heterogeneidade de aplicações e requisitos nas redes é importante encontrar formas de satisfazer simultaneamente diferentes métricas de QoS. A SDN oferece um roteamento inteligente e junto com recursos de virtualização, pode-se separar a rede em fatias ou zonas para isolar as diferentes necessidades por recursos. Cada necessidade com a política de QoS mais apropriada para sua natureza. Por exemplo, para uma transmissão de vídeo em tempo real, pode-se tolerar perda de pacotes, porém é importante uma alta largura de banda para transmissão dos dados, enquanto uma rede de sensores de captação de dados em uma indústria, essa situação pode se inverter. Dessa forma, mecanismos transparentes que combinem métricas de rede como taxa de transmissão, largura de banda, *jitter*, atraso e perda de pacotes são esperados.

Observa-se que a atenção e aplicação da grande maioria das pesquisas é voltada para redes de data centers, ambientes de computação em nuvem, redes de sensores e transmissão de vídeo. Diferentes aplicações podem ser exploradas através do uso de SDN. Esforços voltados para outras aplicações que envolvem grande volumes de dados e diferentes tipos de necessidades, como cidades inteligentes e redes móveis, serão importantes. Por fim, a transição entre arquiteturas de rede tradicionais para arquiteturas baseadas em SDN também é uma questão em aberto. Apesar do surgimento de alguns dispositivos de rede com suporte nativo ao OpenFlow, é difícil substituir todo o hardware que compõe a malha de rede global. Dessa forma, são necessários mecanismos, protocolos e interfaces que permitem a coexistência entre ambas as arquiteturas para viabilizar e acelerar a adoção, a implantação e a globalização de redes definidas por software.

# 7 Conclusão

SDN é uma tecnologia que está crescendo e ganhando força, tanto no meio acadêmico como na indústria. Através do desacoplamento do plano de controle e de dados, torna-se possível programar a rede de forma centralizada e eficiente, sem precisar lidar com linguagens proprietárias de baixo nível. Isso simplifica a rede e permite

que o tráfego de dados seja controlado de forma inteligente e com ações rápidas considerando as condições atuais da rede. Como resultado, ocorre uma otimização do tráfego de dados na medida em que existem políticas de qualidades de serviço granulares, balanceamento de carga dinâmico e controle de congestionamentos.

Nesse artigo, examinamos diferentes métodos e abordagens que estão sendo utilizados pelos pesquisadores para solucionar problemas na otimização do tráfego de dados nas redes através do uso de SDN. Exploramos trabalhos que propõem diferentes formas para lidar com qualidade de serviço, controle de congestionamento e balanceamento de carga. Por fim, apontamos os desafios e oportunidades de pesquisa futuras dentro desse contexto, que atualmente estão relacionados principalmente à escalabilidade, disponibilidade, adaptabilidade, confiabilidade e segurança.

#### Referências

- [1] HAKIRIA, A. et al. Software-defined networking: Challenges and research opportunities for future internet. *Computer Networks*, v. 75, p. 453–471, oct 2014.
- [2] HU, F.; HAO, Q.; BAO, K. A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, v. 16, p. 2181–2206, may 2014.
- [3] NUNES, B. A. A. et al. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, v. 16, p. 1617–1634, feb 2014.
- [4] LARA, A.; KOLASANI, A.; RAMAMURTHY, B. Network innovation using openflow: A survey. *IEEE Communications Surveys & Tutorials*, p. 493–512, fev 2014.
- [5] MCKEOWN, N. et al. Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, v. 38, p. 69–74, mar 2008.
- [6] OPENFLOW deployments. 2016. Disponível em: <a href="http://archive.openflow.org/wp/current-deployments/">http://archive.openflow.org/wp/current-deployments/</a>>. Acesso em 30 set. 2016.
- [7] XIA, W.; NIYATO, D. A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, v. 17, p. 27–51, mar 2015.
- [8] JARRAYA, Y.; MADI, T.; DEBBABI, M. A survey and a layered taxonomy of software-defined networking. *IEEE Communications Surveys & Tutorials*, v. 16, p. 1955–1980, apr 2014.
- [9] NGUYEN, X.-N. et al. Rules placement problem in openflow networks: A survey. *IEEE Communications Surveys & Tutorials*, v. 18, p. 1273–1286, dec 2015.
- [10] WANG, S.; LI, D.; XIA, S. The problems and solutions of network update in sdn: A survey. *Computer Communications Workshops (INFOCOM WKSHPS)*, p. 474–479, aug 2015.
- [11] BRERETONA, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, v. 80, p. 571–583, apr 2007.
- [12] JINYAO, Y. et al. Hiqos: An sdn-based multipath qos solution. *China Communications Magazine Co. Ltd.*, v. 12, p. 123–133, jun 2015.
- [13] TARIQ, S.; BASSIOUNI, M. Qamo-sdn: Qos aware multipath tcp for software defined optical networks. *Consumer Communications and Networking Conference (CCNC)*, p. 485–491, jul 2015.
- [14] YANG, H. et al. Experimental demonstration of latency-aware software defined networking for openflow-based intra-datacenter optical interconnect networks. *Information and Communications Technology*, mar 2013.
- [15] LLOPIS, J. M.; PIECZERAK, J.; JANASZKA, T. Minimizing latency of critical traffic through sdn. *Networking, Architecture and Storage (NAS)*, p. 420–423, aug 2016.

- [16] U-CHUPALA, P. et al. Application-oriented bandwidth and latency aware routing with openflow network. *Cloud Computing Technology and Science (CloudCom)*, feb 2015.
- [17] GOEL, P.; SRINIVASAN, L.; VARMA, V. An adaptive routing scheme for heterogeneous data-flows using openflow. *Cloud Computing in Emerging Markets (CCEM)*, nov 2015.
- [18] GHOLAMI, M.; AKBARI, B. Congestion control using openflow in software defined data center networks. *19th International ICIN Conference Innovations in Clouds, Internet and Networks*, Paris, p. 1–5, mar 2016.
- [19] HERTIANA, S. N.; KURNIAWAN, A. A joint approach to multipath routing and rate adaptation for congestion control in openflow software defined network. *Wireless and Telematics (ICWT)*, nov 2015.
- [20] SONG, S. et al. A congestion avoidance algorithm in sdn environment. *Information Networking (ICOIN)*, p. 420–423, jan 2016.
- [21] JOUET, S.; PERKINS, C.; PEZAROS, D. Otcp: Sdn-managed congestion control for data center networks. *Network Operations and Management Symposium (NOMS)*, p. 209–225, apr 2016.
- [22] MALLIK, A.; HEGDE, S. A novel proposal to effectively combine multipath data forwarding for data center networks with congestion control and load balancing using software-defined networking approach. *Recent Trends in Information Technology (ICRTIT)*, apr 2014.
- [23] REGO, P. A. L. et al. An openflow-based elastic solution for cloud-cdn video streaming service. *Global Communications Conference (GLOBECOM)*, dec 2015.
- [24] SELVI, H. et al. Load balancing in openflow-enabled switches for wireless access traffic aggregation. *Network Operations and Management Symposium (NOMS)*, p. 1013–1014, apr 2016.
- [25] AL-NAJJAR, A.; LAYEGHY, S.; PORTMANN, M. Pushing sdn to the end-host, network load balancing using openflow. *International Conference on Pervasive Computing and Communication Workshops* (*PerCom Workshops*), Sydney, p. 1–6, mar 2016.
- [26] MOGHADAM, M. N.; TAHERI, H.; KARRARI, M. Minimum cost load balanced multipath routing protocol for low power and lossy networks. *Wireless Networks*, v. 20, nov 2014.
- [27] SUNITHA, G. P.; KUMAR, B. P. V.; KUMAR, S. M. D. Optimized congestion aware energy efficient traffic load balancing scheme for routing in wireless sensor networks. *Information Processing (ICIP)*, p. 696–701, dec 2015.
- [28] ELDEFRAWY, K.; KACZMAREK, T. Byzantine fault tolerant software-defined networking (sdn) controllers. *Computer Software and Applications Conference (COMPSAC)*, p. 208–213, aug 2016.
- [29] FOUNDATION, O. N. *OpenFlow*. 2016. Disponível em: <a href="http://archive.openflow.org/">http://archive.openflow.org/</a>. Acesso em 18 set. 2016.
- [30] FOUNDATION, O. N. *ONF*. 2016. Disponível em: <a href="https://www.opennetworking.org/">https://www.opennetworking.org/</a>>. Acesso em 18 set. 2016.
- [31] TOOTOONCHIAN, A.; GANJALI, Y. Hyperflow: a distributed control plane for openflow. *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, p. 3–6, apr 2010.
- [32] KOPONEN, T. et al. Onix: A distributed control platform for large-scale production networks. *OSDI*, v. 10, p. 1–14, oct 2010.
- [33] FOUNDATION, O. N. *OpenFlow Switch Specification Version 1.0.0 (Wire Protocol 0x01)*. 2016. Disponível em: <a href="http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf">http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf</a>>. Acesso em 18 set. 2016.
- [34] FOUNDATION, O. N. *OpenFlow Switch Specification Version 1.1.0 (Wire Protocol 0x02).* 2016. Disponível em: <a href="http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf">http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf</a>>. Acesso em 18 set. 2016.

- [35] FOUNDATION, O. N. *OpenFlow Switch Specification Version 1.2 (Wire Protocol 0x03)*. 2016. Disponível em: <a href="https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.2.pdf">https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.2.pdf</a>. Acesso em 18 set. 2016.
- [36] FOUNDATION, O. N. *OpenFlow Switch Specification Version 1.3 (Wire Protocol 0x04)*. 2016. Disponível em: <a href="https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf">https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf</a>>. Acesso em 18 set. 2016.
- [37] FOUNDATION, O. N. *OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)*. 2016. Disponível em: <a href="https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/spec-v1.4.0.pdf">https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/spec-v1.4.0.pdf</a>>. Acesso em 18 set. 2016.
- [38] FOUNDATION, O. N. *OpenFlow Switch Specification Version 1.5.0 (Protocol version 0x06)*. 2016. Disponível em: <a href="https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf">https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf</a>>. Accesso em 18 set. 2016.
- [39] SASAKI, S. et al. A proposal of qos-aware power saving scheme for sdn-based networks. *International Conference on Network-Based Information Systems*, p. 71–77, dec 2015.
- [40] DURNER, R.; BLENK, A.; KELLERER, W. Performance study of dynamic qos management for openflow-enabled sdn switches. *International Symposium on Quality of Service (IWQoS)*, p. 177–182, jun 2015.
- [41] ZHONG, H. et al. An efficient sdn load balancing scheme based on variance analysis for massive mobile users. *Mobile Information Systems*, oct 2015.
- [42] FOUNDATION, O. *OpenDaylight Controller*. 2016. Disponível em: <a href="https://www.opendaylight.org/">https://www.opendaylight.org/</a>>. Acesso em 28 set. 2016.