

Análise de Características Topológicas Impactantes para a Alocação de Controladores em uma Rede SD-WAN: caso RNP

Vicente Padrenosso^a, Marcelo Vasconcelos^b, André Mendes^c

^{a,b,c}Instituto Militar de Engenharia (IME)

^avicente@ime.eb.br, ^bvasconcelos@ime.eb.br, ^candre.chaves@ime.eb.br

RESUMO: Encontrar uma solução viável e ótima para o problema de alocação de controladores SDN, é uma tarefa desafiadora. Nesse viés, este trabalho foi desenvolvido com o objetivo de expor as características reais de uma rede, que possam impactar na escolha do posicionamento de um controlador SDN. Ademais, são apresentados os experimentos com a ferramenta POCO (Pareto-based Optimal Controller-placement) e com os controladores ONOS e Floodlight, que serviram de comparativo para auxiliar na tomada de decisão quanto ao melhor posicionamento do controlador dentro da rede. Para isso, foi utilizada a topologia da rede Ipê, da RNP (Rede Nacional de Pesquisa), como base desta pesquisa. Em síntese, os resultados demonstraram que devem ser considerados os aspectos, não só estáticos da rede, mas também as características dinâmicas e os aspectos relacionados ao modelo e ao fim a que se destina a utilização dos controladores, fatores que podem impactar no posicionamento.

ABSTRACT: Finding a viable and optimal solution to the SDN controller allocation problem is a challenging task. In this bias, this work was developed with the objective of exposing the real characteristics of a network, which may impact the choice of positioning an SDN controller. Furthermore, the experiments with the POCO tool (Pareto-based Optimal Controller-placement) and with the ONOS and Floodlight controllers are presented, which served as a comparison to assist in decision making regarding the best positioning of the controller within the network. For this, the topology of the Ipê network, from RNP (National Research Network), was used as the basis for this research. In summary, the results showed that not only the static aspects of the network should be considered, but also the dynamic characteristics and aspects related to the model and the purpose for which the controllers are used, factors that can impact the positioning.

PALAVRAS-CHAVE: Controladores. Floodlight. ONOS. POCO. SDN.

KEYWORDS: Controllers. Floodlight. ONOS. POCO. SDN.

1. Introdução

Nos últimos anos, as Redes Definidas por Software (*Software Defined Networking*, SDN), ganharam muita atenção, e vem ganhando mais força com os cenários atuais. Diante disso, flexibilidade, programabilidade e escalabilidade são alguns dos benefícios trazidos pelas redes definidas por software, em comparação à infraestrutura de rede tradicional.

No trabalho de Nunes [1], a rede SDN é definida como um paradigma de rede, em que o hardware de encaminhamento (plano de dados) é dissociado das decisões do controlador (plano de controle). Na SDN,

a inteligência da rede é logicamente centralizada em controladores baseados em software, e os dispositivos de rede tornam-se simples encaminhadores de pacotes, que podem ser programados através de interfaces abertas compatíveis, por exemplo, com o protocolo OpenFlow, [2].

O controlador em uma SDN pode ser considerado o “cérebro” da rede, conforme citado no trabalho de Kreutz [3]. Atua como um ponto de controle estratégico na rede SDN, gerenciando o controle de fluxo para os switches e roteadores. Assim, à medida que as organizações implantam mais redes SDN, os controladores são encarregados de gerenciar a interligação entre os domínios da SDN, usando interfaces de protocolos abertos, como o OpenFlow.

OpenFlow, proposto inicialmente pela Universidade de Stanford, surgiu da necessidade de os pesquisadores executarem protocolos experimentais na rede acadêmica. É a arquitetura mais conhecida e aceita para redes SDN, e possui um conjunto de especificações bem definidas. Trata-se de um protocolo aberto, que permite que um controlador gerencie dispositivos de rede e dite o comportamento deles. A implementação de padrões abertos simplifica o projeto e operação da rede, permitindo que os pesquisadores realizem experimentos sem a necessidade de fabricantes para expor o funcionamento interno de seus produtos, ou desenvolvedores para escrever um software de controle específico do fornecedor.

Na arquitetura estabelecida pelo protocolo OpenFlow, além da figura do controlador, também existe o papel do switch OpenFlow. O switch OpenFlow tem a garantia da confiabilidade na troca de mensagens com o controlador por meio do protocolo SSL (*Secure Socket Layer*). Além disso, a interface do protocolo OpenFlow garante a padronização das mensagens enviadas pelo controlador ao switch, de modo a definir o comportamento do encaminhamento de pacotes, de acordo com a tabela de fluxos. A tabela de fluxos é composta por regras, e cada regra consiste em ações associadas a fluxos. As entradas desta tabela são atualizadas pelo controlador, então, por meio dessa tabela o switch executa o encaminhamento de pacotes.

Ainda, segundo Lange [4], na arquitetura OpenFlow, um controlador logicamente centralizado gerencia os switches da rede, fornecendo-lhes regras para estabelecer o seu comportamento nas operações com pacotes. Portanto, a posição de cada controlador na rede afeta objetivos concorrentes, como latência entre controladores, latência entre switch-controlador e resiliência.

Nesse sentido, uma questão importante durante a implantação de uma SDN é o posicionamento do controlador na rede, ou seja, decidir onde posicionar uma visão geral sobre a rede, que permita gerenciar recursos de forma simples e eficaz, considerando o aproveitamento da capacidade programável para obter respostas eficientes com o mínimo de latência entre os nós e o controlador, alcançando a máxima taxa de transferência entre eles.

A primeira abordagem sobre o assunto, foi feita por Brandon Heller [5], seguido por outros pesquisadores que adicionaram diferentes questões em suas observações. Entretanto, a maioria das pesquisas na área de SDN sobre o posicionamento do controlador concentra grande parte dos seus esforços na análise de redes sem considerar as suas características reais, por exemplo, o tráfego na rede e o tipo de controlador que será implementado. Portanto, as diferentes demandas de uma rede em produção e a carga individual causada por diferentes nós conectados a um controlador ainda é pouco considerada em detalhes.

Ante o exposto, as principais contribuições deste trabalho estão em investigar, de forma comparativa, quais características topológicas possam impactar no posicionamento de controladores em redes SD-WAN (*Software Defined – Wide Area Network*) reais, por meio de um estudo sobre a utilização do método exaustivo implementado pela ferramenta POCO (*Pareto-based Optimal COntroller*) [6], que consiste em, de forma resumida, calcular as distâncias geográficas em relação a todas as posições de uma topologia contendo n nós, a fim de encontrar as melhores posições para o controlador dentro do conjunto de todas as soluções possíveis, onde n é o número de nós da rede. Ademais, nos trabalhos relacionados à ferramenta POCO, não foram encontradas informações exatas quanto ao número máximo de nós suportados.

Essa ferramenta é implementada em Matlab e disponível como software de código aberto. A abordagem POCO avalia exaustivamente todas as possibilidades para o posicionamento do controlador em uma topologia e calcula a compensação de latência e resiliência para cada posição. Dessa forma, o algoritmo não fornece nenhuma recomendação específica para um determinado posicionamento, mas retorna um conjunto de posicionamentos baseados no método *Pareto-based Optimal Placement* [7], que permite aos administradores de rede a escolha de um posicionamento apropriado. Assim, essa estratégia dá mais flexibilidade na implantação de uma rede SD-WAN.

Então, para alcançar os objetivos propostos, os resultados definidos pela ferramenta POCO são comparados no estudo de caso da rede Ipê, da RNP (Rede Nacional de Ensino e Pesquisa), inserida em um ambiente virtualizado, considerando as suas características reais de conexão e tráfego, e também a análise dos controladores ONOS e Floodlight ativos nessa topologia, conforme será apresentado na **seção 4** - Metodologia aplicada.

Como resultado, é demonstrado que nos estudos sobre o posicionamento dos controladores, somente as considerações sobre as distâncias geográficas e latências definidas pela ferramenta POCO não são suficientes. Logo, os testes provaram que as características de latência em conjunto com as informações de largura de banda e tráfego também são fatores importantes na definição do posicionamento, além disso, os testes também mostraram que os aspectos relacionados ao modelo e à utilização dos controladores devem ser considerados.

Após essa introdução, serão apresentadas a fundamentação teórica e a motivação na **seção 2**, na **seção 3** os trabalhos relacionados, na **seção 4** a metodologia aplicada, na **seção 5** os experimentos realizados e a análise dos resultados, por fim, a conclusão e trabalhos futuros na **seção 6**.

2. Motivação

De acordo com trabalhos mais recentes, como o realizado por Tamal Das [8], que apresenta um *survey* com uma profunda pesquisa sobre o problema do posicionamento do controlador SDN, e classifica extensivamente outros trabalhos em várias perspectivas, é possível verificar que ainda a maioria das pesquisas na área de SDN sobre o posicionamento dos controladores concentra grande parte dos seus esforços na análise de redes, considerando somente suas características estáticas, ou seja, dados como cálculos das distâncias geográficas e latências relacionadas a essas distâncias, conforme observado também no trabalho de Hock sobre a ferramenta POCO [6]. Além disso, as características

da rede e a carga individual causada por diferentes nós conectados a um controlador ainda é pouco considerada em detalhes.

Em primeiro lugar, um objetivo principal para o bom posicionamento do controlador, é minimizar as latências entre os nós e os controladores na rede. No entanto, olhar apenas para os atrasos e medidas de desempenho estáticas não é suficiente. Um controlador SDN é um ponto estratégico de controle na rede. Assim, decidir onde alocar os controladores SDN, quais switches serão controlados por cada controlador, quantos controladores são necessários para que não haja um ponto limitante, e quais controladores tem o melhor desempenho de acordo com o posicionamento em cada cenário são questões importantes, a fim de minimizar o custo de recobrimento da rede e garantir um bom desempenho.

3. Trabalhos relacionados

Vários estudos foram realizados com o objetivo de tratar o problema de posicionamento dos controladores, outros estudos trabalharam no sentido de comparar controladores SDN de forma isolada, focando apenas no desempenho do controlador. Porém, poucos estudos auxiliaram na tomada de decisão para selecionar um controlador com o melhor desempenho, levando em conta o seu posicionamento dentro de uma rede com características reais.

O problema de posicionamento do controlador na arquitetura SDN foi introduzido por Heller [5], onde foi realizada uma otimização em relação à latência dos nós até o controlador designado. Nesse artigo, o autor cita que para as WANs (*Wide Area Networks*), o melhor posicionamento depende, entre outras métricas, da latência.

No campo do problema de posicionamento, Heller [5] mostra o desempenho da rede variando a posição dos controladores na rede. Assim, encontrar a localização e o número de controladores que serão necessários é uma tarefa desafiadora em uma arquitetura de rede como a SD-WAN.

No trabalho de Stanislav Lange [4], é apresentado o POCO (*Pareto-based Optimal COntroller placement*) [6],

uma estrutura para encontrar o posicionamento ideal do controlador de forma que a conectividade entre os switches e o controlador seja maximizada levando-se em consideração a capacidade do controlador. Primeiro, os autores propõem um algoritmo de força bruta, mas é válido apenas para pequenas redes. Em seguida, os autores propõem heurísticas para resolver o problema de posicionamento de controladores em grandes redes. Para tanto, utilizam um algoritmo baseado no *Pareto Simulated Annealing* [9]. Neste trabalho, os autores se concentram em maximizar a resiliência da rede. Onde eles consideraram a colocação de controladores em uma rede SDN dinâmica, na qual existem variações de latência entre os controladores e seus switches. Porém, não consideram a posição de controladores levando em consideração a alocação dinâmica da rede.

O posicionamento precisa ser escolhido de maneira cuidadosa. O *framework* POCO, tem a capacidade de lidar com topologias de pequeno e médio porte, que fornecem a solução em segundos. Entretanto, para redes de larga escala, a avaliação exaustiva necessita de uma quantidade considerável de esforço computacional e uso de memória. É neste contexto, que se torna necessária a busca por uma solução computacional mais próxima aos ambientes reais.

O trabalho de Rastogi e Bais [10], realiza uma análise comparativa em termos da capacidade do tráfego. O objetivo central do trabalho consiste em apresentar uma análise entre dois controladores, denominados Pox e Ryu, respectivamente, em termos da capacidade de manuseio de tráfego. O emulador Mininet foi utilizado para emular o ambiente dos controladores SDN, e assim monitorar o desempenho do tráfego. Porém, o trabalho não considerou as características reais da rede, como por exemplo a capacidade e ocupação dos links.

Já sobre a comparação dos controladores, o trabalho realizado por Amin Tootoonchian [11], um dos primeiros estudos comparativos dos controladores

SDN, considerou um número limitado de controladores (NOX, NOX-MT, Beacon e Maestro) focando apenas no desempenho do controlador. Com o avanço das tecnologias tais controladores já são considerados ultrapassados.

O trabalho de Bondkovskii [12], faz uma comparação qualitativa entre dois controladores SDN de código aberto, o OpenDaylight e o Open Network Operation System (ONOS). O estudo foca na interface Northbound destes dispositivos.

Pesquisas mais recentes, como a realizada por Lusani Mamushiane^[13], estuda e avalia o desempenho de alguns controladores populares de código aberto como ONOS, Ryu, Floodlight e OpenDaylight em termos somente de latência como métrica, usando uma ferramenta de *benchmarking* OpenFlow chamada Cbench.

Também, no trabalho realizado por Tamal Das [8] é apresentado um *survey* com uma profunda pesquisa sobre o problema do posicionamento do controlador SDN, que classifica extensivamente os trabalhos existentes em várias perspectivas.

No trabalho de Ola Salman [14] é realizada uma avaliação qualitativa de controladores SDN de código aberto (MUL, Beacon, Maestro, ONOS, Ryu, OpenDaylight, Floodlight, NOX, IRIS, Libfluid-based e POX). As métricas avaliadas são latência e taxa de transferência desempenhadas sobre um número variável de switches. Os resultados obtidos sugerem que MUL e Libfluid-based têm o melhor desempenho de taxa de transferência, enquanto o Maestro mostrou melhor desempenho de latência. Outros trabalhos relacionados são comparados na **tabela 1**.

Neste trabalho serão comparados os controladores ONOS e Floodlight, que serão avaliados em dois cenários de posicionamento em uma rede SD-WAN, conforme a metodologia que será apresentada na **seção 4**.

Trabalhos Relacionados - Múltiplos Objetivos													
Trabalho	Posicionamento	Latência Switch-Controlador	Latência entre Controladores	Capacidade do Controlador	Resiliência	Confibilidade	Balancamento de Carga	Traáfego Switch-Controlador	Número de Controladores	Análise de Topologia Real	Aplicação do Controlador	Public	Qualis
[6]	✓	✓	✓							✓		IEEE	A1
[4]	✓	✓	✓	✓			✓			✓		IEEE	A1
[15]	✓	✓	✓	✓			✓					IEEE	A1
[16]	✓	✓					✓	✓				IEEE	A1
[17]	✓	✓	✓	✓	✓					✓		IEEE	A1
[18]	✓	✓	✓	✓	✓					✓		IEEE	A1
[19]	✓	✓					✓			✓		IEEE	A1
[20]	✓	✓	✓	✓			✓			✓		Springer	A1
[21]	✓	✓	✓	✓			✓			✓		IEEE	A1
[22]	✓	✓	✓	✓			✓			✓		IEEE	A1
[23]	✓	✓	✓	✓			✓			✓		IEEE	A1
[24]	✓	✓	✓				✓			✓	✓	IEEE	A1

Tab. 1: Trabalhos relacionados.

4. Metodologia aplicada

Este trabalho tem como base o estudo do método exaustivo utilizado na ferramenta POCO (Pareto-based Optimal Controller) para computar o posicionamento ótimo de controladores SDN na rede Ipê, da RNP.

Também foi utilizado o Mininet [25], como emulador de redes SDN empregado em um ambiente virtualizado para análise e comparação do impacto do posicionamento utilizando os controladores ONOS e Floodlight na operação de redes habilitadas com OpenFlow. A **figura 1**, mostra um exemplo da topologia da RNP na interface gráfica do POCO.

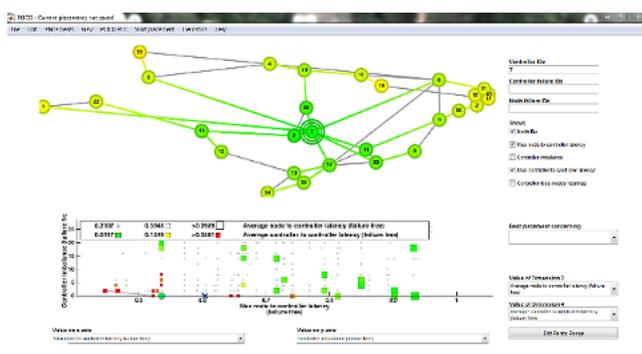


Fig. 1 – Exemplo da interface gráfica do POCO.

O emulador Mininet foi escolhido, principalmente, por ser possível emular redes reais e também pelos

seguintes motivos: oferece suporte a vários controladores SDN; os hosts são Linux com características “reais” que podem executar, potencialmente, qualquer programa que rode em Linux; possibilita análise de tráfego através da captura de pacotes usando, por exemplo, o Wireshark ou TCPDUMP; permite testes de conectividade com ping / fping ou de medidas de desempenho com iPerf; é possível criar links com largura de banda (*bandwidth*) e latências específicas, além de desativar e ativar os links.

Existe uma série de controladores disponíveis na literatura. Entre os mais conhecidos estão o NOX, POX, Floodlight, OpenDaylight (ODL), *Open Network Operating System* (ONOS) e RYU. Contudo, em uma perspectiva de implementação prática, é muito difícil determinar qual controlador terá o melhor desempenho em qualquer tipo de rede. Na pesquisa realizada por Zhu [26], é possível verificar uma boa análise comparativa sobre os recursos de diversos controladores.

Nesse sentido, os critérios para a escolha dos controladores analisados nesta pesquisa incluem: interfaces (protocolos e versões suportadas), REST API, interface gráfica do usuário (GUI), modularidade, sistema operacional (SO) suportado, parcerias de desenvolvimento e suporte, documentação, linguagem de programação, virtualização, área de aplicação e arquitetura. Então, foram escolhidos os controladores ONOS e Floodlight.

Todo o ambiente de testes foi configurado em uma máquina virtual (VM), criada com o sistema operacional Ubuntu 16.04.05 LTS, contendo a versão do kernel padrão 4.4.0-87-generic, memória RAM de 8 GB, 8 processadores e o espaço de armazenamento interno de 30 GB. Essa VM foi configurada em um desktop com Windows 10 Home - 64 bits, processador Intel Core i7-9700 e 16 GB de memória RAM.

A topologia escolhida para os experimentos foi a rede Ipê, da RNP (Rede Nacional de Ensino e Pesquisa). A RNP é primeira rede de acesso à Internet no Brasil, integra mais de 800 instituições de ensino e pesquisa no país, conta com 28 nós posicionados em todos os Estados do território nacional beneficiando a mais de 3,5 milhões de usuários, evoluiu das linhas telefônicas

para conexões em fibra óptica de altíssima capacidade, de 64 Kb/s para mais de 100 Gb/s. Todas as informações utilizadas nos experimentos referentes às conexões entre os links, largura de banda e tráfego de entrada e saída entre os nós foram coletadas do panorama de tráfego da topologia real da RNP, disponíveis no próprio site da instituição, conforme exemplo mostrado na **figura 2**.

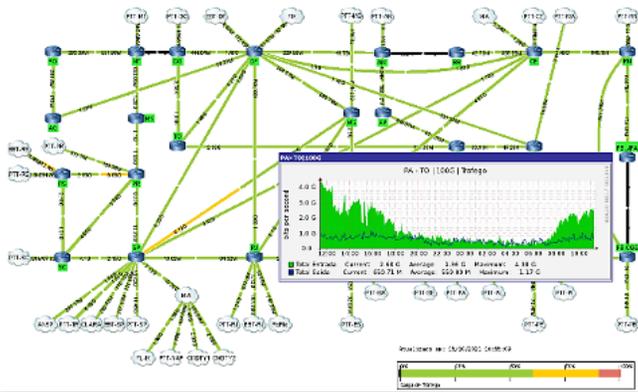


Fig. 2 – Panorama de conexões e tráfego da RNP.

Inicialmente, foi realizado um estudo das funcionalidades principais da ferramenta POCO para definição do posicionamento do controlador em uma rede SD-WAN. Além disso, também foram comparados os controladores ONOS e Floodlight, que foram avaliados em dois cenários de posicionamento, onde no primeiro cenário foram consideradas somente as informações de distâncias e latências, ou seja, os links foram testados livres de tráfego, utilizando a capacidade total do enlace. No segundo cenário, foram incluídos os tráfegos de ocupação dos enlaces diretamente conectados, de acordo com as informações reais do panorama de tráfego da RNP. Então, foram avaliadas todas as 28 possíveis posições para o controlador na topologia da RNP, e esses posicionamentos foram definidos como: (i) posição ótima e (ii) pior posição. Por posição ótima, entende-se o posicionamento no qual a latência entre os nós e o controlador é minimizada. Por pior posição, entende-se o posicionamento para o qual se tem maior latência entre os nós e o controlador.

Dessa forma, os resultados definidos pelos testes das posições com os controladores foram comparados com os resultados das posições definidas pelo método

exaustivo da ferramenta POCO, que só considerou o primeiro cenário, sem tráfego. Essas comparações não consideraram apenas métricas estáticas como, por exemplo, as distâncias entre os nós da rede, mas também medidas quantitativas, como transferência de dados, *jitter* e perda de pacotes no cenário com tráfego.

5. Testes e análise de resultados

Com todo o ambiente de teste configurado, foram realizados os testes com as topologias criadas, integrando os controladores nos cenários sem tráfego (cenário 1) e com tráfego (cenário 2). Desse modo, para cada teste foi posicionado um host (srv) para simular o controlador em cada uma das 28 posições da topologia da RNP e foi executado o comando *ffping* para obter as informações de latência a partir do controlador (srv) nessa determinada posição, com destino a todos os outros hosts nas demais posições.

Para obter melhores resultados estatísticos e alcançar maior eficácia nas comparações, foram consideradas 30 amostras dos testes com o *ffping* ($n = 30$) e nível de confiança de 95%. Então, neste caso, foram executados 30 testes com o *ffping* para cada uma das posições da topologia, e em cada teste foram enviados 30 pacotes ICMP para cada destino.

Para a execução dos testes no cenário 2, onde foi considerado também o tráfego na rede (ocupação dos links), foi utilizado o programa *iPerf* para a simulação do tráfego, considerando as capacidades dos enlaces e os valores de tráfego real extraídos do panorama de tráfego da topologia da RNP. Após isso, também foi executado o teste com o *ffping*, da mesma forma como no cenário sem tráfego.

Iperf é uma ferramenta para medições ativas da largura de banda máxima alcançável em redes IP, podendo realizar injeção de pacotes (tanto TCP quanto UDP) para medir o desempenho de redes de computadores. Ela suporta o ajuste de vários parâmetros relacionados a tempo, buffers e protocolos (TCP, UDP, SCTP com IPv4 e IPv6). Para cada teste, ela relata a largura de banda, perda e outros parâmetros.

Assim, os resultados dos testes com os controladores no cenário sem tráfego, considerando a melhor e a pior

posição, foram comparados com os resultados obtidos pelos cálculos da ferramenta POCO. Logo, esses resultados dos controladores puderam confirmar os posicionamentos estabelecidos pela ferramenta POCO ou mostraram novos posicionamentos, considerando outros fatores como, por exemplo, a capacidade dos links. Já os testes no cenário com tráfego mostraram a importância de se considerar métricas como as capacidades dos *links*, a capacidade de transferência de dados, o *jitter* e a perda de pacotes, além de somente considerar a distância entre os nós.

Para a execução dos testes com a ferramenta POCO, com a finalidade de comparar mais resultados, foram utilizados os posicionamentos e informações da topologia da RNP que estava operacional em três diferentes períodos (novembro de 2020, março de 2021 e julho de 2021), visto que, algumas alterações foram observadas nestas topologias, por exemplo, notou-se que alguns links deixaram de existir ou suas capacidades foram ampliadas, e outros novos links foram estabelecidos.

5.1 Posicionamentos POCO

Como primeiros resultados, foram examinadas as posições definidas pela ferramenta POCO para as três topologias citadas. Para isso, foi considerado o cenário livre de falhas, com apenas um controlador ($k = 1$). Nesses resultados é possível perceber que os valores das latências médias são idênticos ou muito próximos para as mesmas posições das topologias testadas. Portanto, nota-se que os resultados foram mantidos com o método exaustivo utilizado pela ferramenta POCO para os cálculos das latências baseados somente nas distâncias geográficas. Por isso, pode-se concluir que mesmo com as alterações nos links das topologias, mas sem alterações nas posições físicas dos nós, os resultados não demonstraram alterações significativas, ou seja, as distâncias e latências foram mantidas, conforme mostrado na **figura 3**.

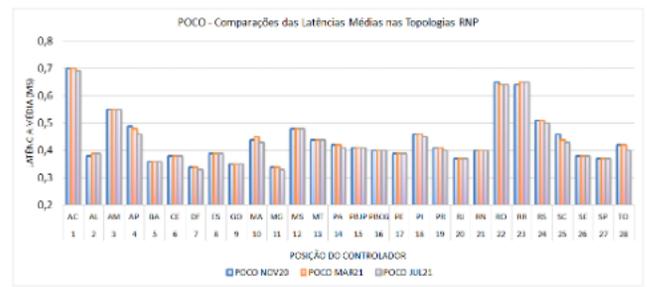


Fig. 3 – Classificação das posições POCO.

Então, para as três topologias testadas, a posição 7 (DF) foi definida como a melhor posição, com a menor latência média, e a posição 1 (AC) como a pior posição, com a maior latência. As definições das posições estão representadas por círculos maiores nas **figuras 4 e 5**.

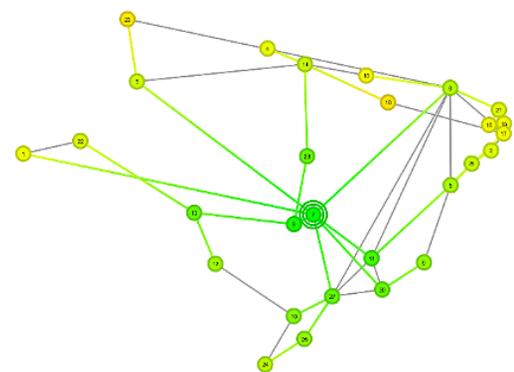


Fig. 4 – Classificação da melhor posição POCO.

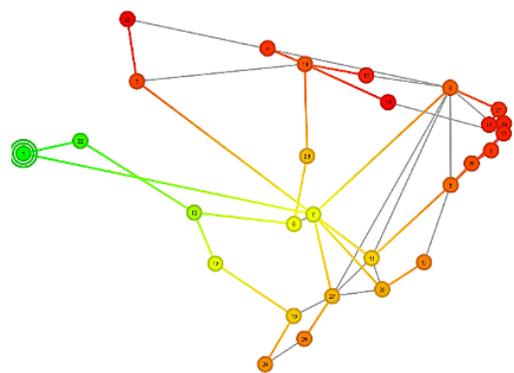


Fig. 5 – Classificação da pior posição POCO.

Na **figura 4**, também é possível perceber que o nó definido com a menor latência (7-DF), também é o nó mais centralizado na topologia, portanto, o de menor distância entre dois pontos, considerando somente as distâncias geográficas entre os nós com links estabelecidos. Também, verifica-se que o nó 7 (DF), tem uma boa quantidade de enlaces redundantes, o que facilita a ligação com os demais nós da rede. Sob outra perspectiva, o motivo do nó 1 (AC) ter sido definido como a pior posição pode ser justificado pela sua posição geográfica ser a mais distante em relação aos demais nós da topologia. Dessa forma, torna-se mais um indicativo de que a distância geográfica entre os nós têm bastante peso nas decisões por meio da ferramenta POCO, o que pode não ser suficiente para a decisão do posicionamento do controlador.

5.2 Classificações dos controladores

Após a classificação das posições pela ferramenta POCO, as topologias também foram testadas no Mininet utilizando os controladores ONOS e Floodlight, considerando os dois cenários propostos. Também, foram realizados testes considerando métricas críticas de desempenho, como quantidade de dados transmitida, *jitter* e perda de dados.

5.2.1 Canário 1 - Sem ocupação dos links

Nesta seção, são mostradas as comparações dos resultados dos testes de latência entre a ferramenta POCO e os controladores no primeiro cenário, sem tráfego. Para execução desses experimentos, foi utilizada a topologia que estava operacional no mês de novembro de 2020, quando foram iniciados os experimentos.

Para a realização do primeiro teste, já no Mininet com o cenário e a topologia pronta, os links foram utilizados livres de tráfego, com a capacidade total do enlace. Então, para cada teste foi posicionado um host (srv) para simular o controlador em cada uma das 28 posições e foi executado o comando *ffping* para obter as informações de latência a partir do controlador (srv) nessa determinada posição com destino a todos os outros hosts nas demais posições.

Para obter melhores resultados estatísticos e alcançar maior eficácia nas comparações, foram consideradas 30 amostras dos testes com o *ffping* (n = 30) e nível de confiança de 95%. Para chegar a esse número de amostras (30 pings), foram executados testes com 10, 20, 30, 40, 50 e 100 pings com o controlador ONOS entre duas posições da topologia de estudo desta pesquisa. Assim, foi constatado o melhor caso, conforme mostrado na **figura 6**. Então, nos resultados apresentados a seguir, foram enviados 30 pacotes ICMP para cada destino.

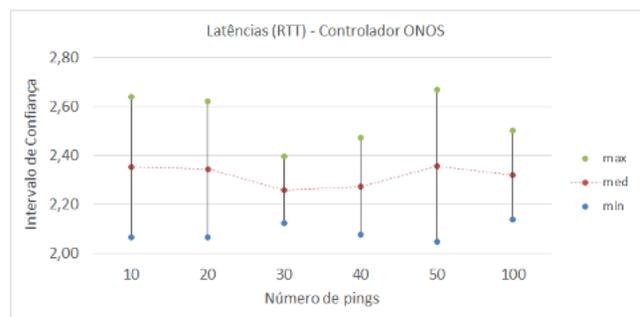


Fig. 6 – Topologia Nov 20 - sem ocupação dos links (cenário 1).

Para exemplificar o que foi descrito anteriormente, segue o comando que foi executado a partir do host controlador: *ffping -s -g 10.0.0.201 10.0.0.228 -c 30*, conforme parâmetros da **tabela 2**.

Tab. 2: Parâmetros do *ffping*.

-s	Imprime as estatísticas ao final do teste
-g	Gera uma lista de alvos a partir um range de IP, com os endereços IP inicial e final
-c	Especifica o número de pacotes a serem enviados para cada destino. Neste teste foram enviados 30 pacotes ICMP para cada destino

O range de IPs (-g 10.0.0.201 10.0.0.228) significa que o comando foi disparado do controlador com IP 10.0.0.200 para todos os hosts no range que varia na sequência do host 1, conectado na posição 1 (switch 1), com IP 10.0.0.201, até o host 28, na posição 28 (switch 28), com IP 10.0.0.228.

Após testadas as 28 posições, os resultados das latências médias foram comparados com os resultados calculados pela ferramenta POCO, conforme mostrado no gráfico

Neste cenário, com tráfego, foi exigida uma grande capacidade computacional, principalmente de CPU e memória, o que dificultou os testes com a topologia completa da RNP. Neste caso, durante a geração do tráfego entre os hosts, o sistema virtualizado não suportou o tráfego e os controladores não conseguiram processar todo o tráfego demandado durante um período de tempo necessário para execução dos testes com o *fping*, logo, o tráfego foi interrompido, e não foi possível executar os testes de latências. Dessa forma, como tentativa de uma solução, todo o fluxo de pacotes foi escalonado com a diminuição de largura de banda e quantidade de tráfego gerado, porém, mesmo com essas alterações, ainda não foi possível executar os testes com um mínimo de eficácia, considerando o tráfego simultâneo de entrada e saída nos 28 nós da rede.

Diante disso, foram pesquisados alguns trabalhos mais recentes com possíveis soluções para o problema. Entre eles, o trabalho apresentado por Ahmadi [27] propõe um algoritmo heurístico denominado *Multi-Start Hybrid Non-Dominated Sorting Algorithm* (ou MHNSGA) para resolver o problema de posicionamento de controladores de forma eficaz. Porém, no contexto da otimização multiobjetivo, na maioria dos casos, pode não existir uma solução única que otimize todos os objetivos considerados.

O autor deste trabalho defende que os resultados de várias avaliações mostraram que o algoritmo proposto é capaz de explorar grande parte do espaço de busca e obter uma estimativa da fronteira ótima de Pareto com alto grau de precisão. Entretanto, segundo o próprio autor, em comparação com a busca exaustiva da ferramenta POCO, esse método pode ser menos preciso, mas requer menos tempo computacional e memória.

Assim, encontrar uma solução ótima para o posicionamento do controlador, considerando características de fluxos de tráfego de uma rede real aplicadas em um ambiente de pesquisa com capacidade computacional viável, ainda requer mais investigação sobre o problema.

Embora a análise do fluxo de tráfego na rede seja um fator crucial para a tomada de decisão sobre o posicionamento de controladores SDN, alguns outros fatores importantes também devem ser considerados. Esses fatores incluem as capacidades dos controladores e as cargas nos switches.

Então, ainda considerando as características de uma rede real em um cenário com tráfego, na próxima seção também serão comparadas algumas métricas críticas, como transferência de dados, *jitter* e perda de pacotes, que podem influenciar na escolha do posicionamento e no comportamento do controlador em uma SDN.

5.2.3 Métricas críticas de desempenho

Nesta seção, serão apresentados outros testes, considerando algumas métricas críticas que podem impactar no posicionamento e também no funcionamento do controlador. Nesse sentido, serão apresentados os testes de transferência de pacotes na **seção 5.2.4**, os testes com a métrica *jitter* na **seção 5.2.5** e o resultado dos testes de perda de pacotes em **5.2.6**. Por fim, essas três métricas serão comparadas na **seção 5.2.7**, observando o desempenho dos controladores ONOS e Floodlight inseridos nas topologias testadas.

Os resultados exibidos foram obtidos por meio da geração de tráfego na rede utilizando a ferramenta *iPerf*, da mesma forma, como executado nos testes com o **cenário 2**, descrito em **5.2.2**. Porém, esse tráfego foi reduzido e gerado durante o tempo total de 60 segundos. Dessa forma, esse período de fluxo foi suficiente para a avaliação dessas métricas sem que ocorressem os problemas de esgotamento dos recursos computacionais. Então, as informações geradas foram analisadas e comparadas, conforme será apresentado nas próximas Seções.

5.2.4 Quantidade de dados transferidos

Cada link diretamente conectado, foi ocupado com os seus respectivos tráfegos máximos de entrada e saída, de acordo com as informações do panorama

de tráfego da RNP verificado no período analisado. Então, após a ocupação dos links e geração do tráfego durante o período de tempo total analisado (60 seg), foram coletados os dados de transferência média em cada um dos links, ou seja, a quantidade média de dados transferidos (Mbps) neste período de teste.

O gráfico na **figura 9** mostra a comparação dos controladores ONOS e Floodlight sobre as quantidades de dados transferidos (Mbps). Nesses gráficos pode-se verificar de forma geral que os controladores tiveram o mesmo tratamento no processamento do tráfego entre os nós, pois, foi transferida a mesma quantidade de tráfego, com os dois controladores e não foram evidenciadas diferenças significativas.

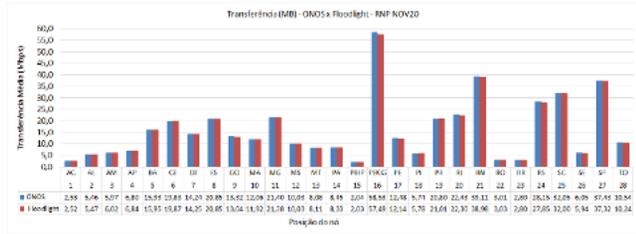


Fig. 9 – ONOS x Floodlight - Comparação da métrica transferência.

5.2.5 Jitter

De forma simples e resumida, *jitter* é a variação da latência, ou seja, essa métrica pode ser definida como a medida de variação ou “flutuação” do tempo que um pacote de dados leva para ir a um destino e voltar.

Os gráficos da **Fig. 10** mostram que o controlador ONOS teve leve desvantagem na comparação isolada da métrica *jitter* com controlador Floodlight, pois apresentou maior variação do atraso total, considerando todas as posições testadas. Por isso, outros fatores também devem ser analisados.

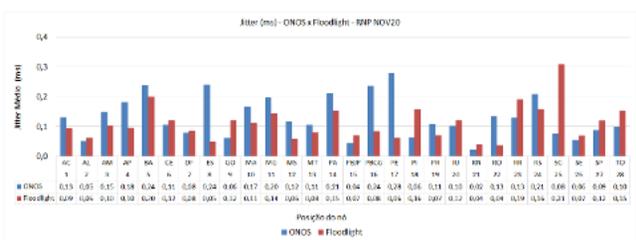


Fig. 10 – ONOS x Floodlight - Comparação da métrica jitter.

5.2.6 Perda de pacotes

Estes experimentos foram executados nos cenários livres de falhas, portanto, foram consideradas as perdas somente nas transferências de pacotes, não foram simuladas falhas nos enlaces ou dispositivos.

Os gráficos da **figura 11** mostram a comparação entre controladores sobre a perda de pacotes medida. Os resultados mostram que o controlador ONOS teve perda média de pacotes menor que a perda testada com o controlador Floodlight.

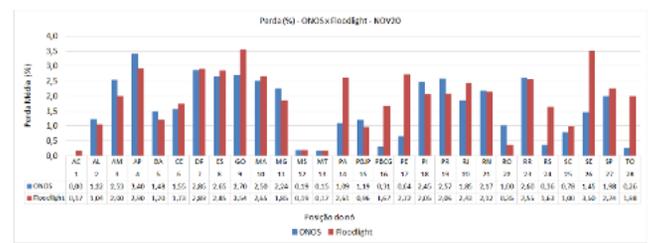


Fig. 11 – ONOS x Floodlight - Comparação da métrica perda.

5.2.7 Comparação das métricas críticas

Em resumo, na comparação entre os dois controladores, foi transferida a mesma quantidade de pacotes, com uma pequena vantagem para o controlador ONOS, porém, não muito significativa. O controlador Floodlight apresentou menor variação do atraso (*jitter*), entretanto, houve maior perda de pacotes. A Tabela 3 mostra o resumo descrito, onde estão representadas a maior transferência, o menor *jitter* e a menor perda de pacotes.

Tabela. 3 - Comparação das métricas críticas de desempenho.

Comparação das métricas críticas			
	Transf	Jitter	Perda
ONOS	✓		✓
Floodlight		✓	

Portanto, com a mesma quantidade de pacotes transferidos nos dois testes com os controladores, o fato de o ONOS ter apresentado menor perda de pacotes, pode ser justificado pelo melhor tratamento dado ao fluxo recebido no controlador, o que está

ligado à sua capacidade de armazenamento. Por outro lado, o Floodlight apresentou menor *jitter*, o que evidencia o baixo armazenamento dos dados na memória RAM, no *buffer* e na *cache*, porém, isso pode causar maior perda de dados e pode reduzir a capacidade de resposta do controlador.

Assim, constata-se também que, além das métricas de distâncias, de latências e de considerações do tráfego na rede, também é importante identificar as prioridades de tráfego e decidir qual será o controlador mais apropriado para tal finalidade.

6. Conclusão e trabalhos futuros

Neste trabalho, foi apresentado um estudo sobre o problema de posicionamento de controladores em SDN. Nesse contexto, foi descrita a implementação realizada no *framework* POCO que, de forma geral, apresentou bons resultados considerando as medidas estáticas de distância e latência, porém, com os resultados

apresentados nos testes incluindo os controladores na rede, é possível destacar que as características reais da rede podem impactar de forma expressiva na escolha do posicionamento de um controlador SDN. Também, é possível observar que a escolha de um determinado tipo de controlador deve ser pensada de acordo com a finalidade desejada, pois suas características também podem impactar no posicionamento.

Por fim, cabe enfatizar que o posicionamento eficiente do controlador tenta melhorar o desempenho de métricas, como latência, prioridades de tráfego, perda e assim por diante. Contudo, o estudo para o problema do posicionamento do controlador ainda pode compreender diversas soluções diferentes. Portanto, espera-se ampliar o estudo para o desenvolvimento e aprimoramento de ferramentas para o posicionamento dos controladores, com base em medidas de desempenho de redes reais, e que essas ferramentas possam analisar o problema de posicionamento de controladores na presença de tráfego de dados.

Referências

- [1] B. A. A. Nunes et al. “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks”. In: IEEE Communications Surveys Tutorials 16.3 (Third 2014), pp. 1617–1634.
- [2] NICK MCKEOWN ET AL. “OpenFlow: enabling innovation in campus networks”. In: ACM SIG-COMM Computer Communication Review 38.2 (2008), pp. 69–74.
- [3] DIEGO KREUTZ ET AL. “Software-defined networking: A comprehensive survey”. In: Proceedings of the IEEE 103.1 (2015), pp. 14–76.
- [4] STANISLAV LANGE ET AL. “Heuristic approaches to the controller placement problem in large scale SDN networks”. In: IEEE Transactions on Network and Service Management 12.1 (2015), pp. 4–17.
- [5] BRANDON HELLER, ROB SHERWOOD, AND NICK MCKEOWN. “The controller placement problem”. In: ACM SIGCOMM Computer Communication Review 42.4 (2012), pp. 473–478.
- [6] DAVID HOCK ET AL. “POCO-framework for Pareto-optimal resilient controller placement in SDN-based core networks”. In: 2014 IEEE Network Operations and Management Symposium (NOMS). IEEE. 2014, pp. 1–2.
- [7] GUODONG WANG ET AL. “The controller placement problem in software defined networking: A survey”. In: IEEE Network 31.5 (2017), pp. 21–27.
- [8] TAMAL DAS, VIGNESH SRIDHARAN, AND MOHAN GURUSAMY. “A survey on controller placement in SDN”. In: IEEE communications surveys & tutorials 22.1 (2019), pp. 472–503.
- [9] PIOTR CZYZZAK AND ADREZEJ JĄSZKIEWICZ. “Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization”. In: Journal of Multi-Criteria Decision Analysis 7.1 (1998), pp. 34–47.
- [10] ABHISHEK RASTOGI AND ABDUL BAIS. “Comparative analysis of software defined networking (SDN) controllers—In terms of traffic handling capabilities”. In: 2016 19th International Multi-Topic Conference (INMIC). IEEE. 2016, pp. 1–6.
- [11] AMIN TOOTOONCHIAN ET AL. “On Controller Performance in Software-Defined Networks”. In: 2nd {USENIX} Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE 12). 2012, pp. 1–6.

- [12] ANDREI BONDKOVSKII ET AL. “Qualitative comparison of open-source sdn controllers”. In: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium. IEEE. 2016, pp. 889–894.
- [13] LUSANI MAMUSHIANE, ALBERT LYSKO, AND SABELO DLAMINI. “A comparative evaluation of the performance of popular SDN controllers”. In: 2018 Wireless Days (WD). IEEE. 2018, pp. 54–59.
- [14] Ola Salman et al. “SDN controllers: A comparative study”. In: 2016 18th Mediterranean Electrotechnical Conference (MELECON). IEEE. 2016, pp.1–6.
- [15] Md Tanvir Ishtaique ul Huque et al. “Large-scale dynamic controller placement”. In: IEEE Transactions on Network and Service Management 14.1 (2017), pp. 63–76.
- [16] Kshira Sagar Sahoo et al. “Solving multi-controller placement problem in software defined network”. In: 2016 International Conference on Information Technology (ICIT). IEEE. 2016, pp. 188–192.
- [17] H SOFIA NANING, RENDY MUNADI, AND MUHAMMAD ZEN EFFENDY. “SDN controller placement design: For large scale production network”. In: 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob). IEEE. 2016, pp. 74–79.
- [18] PETRA VIZARRETA, CARMEN MAS MACHUCA, AND WOLFGANG KELLERER. “Controller placement strategies for a resilient SDN control plane”. In: 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM). IEEE. 2016, pp. 253–259.
- [19] DIOGO MF MATTOS, OTTO CARLOS MB DUARTE, AND GUY PUJOLLE. “A resilient distributed controller for software defined networking”. In: 2016 IEEE International Conference on Communications (ICC). IEEE. 2016, pp. 1–6.
- [20] AHMAD JALILI, MANIJEH KESHTGARI, AND REZA AKBARI. “Optimal controller placement in large scale software defined networks based on modified NSGA-II”. In: Applied Intelligence 48.9 (2018), pp. 2809–2823.
- [21] DAVID HOCK et al. “POCO-PLC: Enabling dynamic pareto-optimal resilient controller placement in SDN networks”. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE. 2014, pp. 115–116.
- [22] STANISLAV LANGE et al. “Specialized heuristics for the controller placement problem in large scale SDN networks”. In: 2015 27th International Teletraffic Congress. IEEE. 2015, pp. 210–218.
- [23] HU BO ET AL. “The controller placement problem for software-defined networks”. In: 2016 2nd IEEE International Conference on Computer and Communications (ICCC). IEEE. 2016, pp. 2435–2439.
- [24] PENG XIAO et al. “The SDN controller placement problem for WAN”. In: 2014 IEEE/CIC International Conference on Communications in China (ICCC). IEEE. 2014, pp. 220–224.
- [25] BOB LANTZ, BRANDON HELLER, AND NICK MCKEOWN. “A network in a laptop: rapid prototyping for software-defined networks”. In: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. 2010, pp. 1–6.
- [26] LIEHUANG ZHU et al. “SDN controllers: Benchmarking & performance evaluation”. In: arXiv preprint arXiv:1902.04491 (2019).
- [27] VAHID AHMADI AND MOSTAFA KHORRAMIZADEH. “An adaptive heuristic for multi-objective controller placement in software-defined networks”. In: Computers & Electrical Engineering 66 (2018), pp. 204–228.