

Uma abordagem multiobjetivo para definição de rotas em Redes Definidas por Software (SDN)

Servílio Souza de Assis e Luciano de Errico

Resumo— A popularização de serviços e aplicações na nuvem tem mudado drasticamente os perfis de tráfego na Internet. Tais mudanças têm motivado a evolução do paradigma SDN (Software Defined Networks), onde os planos de dados e controle dos switches são separados, possibilitando uma visão centralizada de controle e uma série de novas aplicações. Dois importantes problemas em engenharia de tráfego são o balanceamento de carga e a eficiência energética no uso de recursos na rede. Neste cenário, o presente trabalho propõe um modelo de otimização multiobjetivo para roteamento. Testes realizados indicaram melhora em performance e eficiência energética, conforme as preferências colocadas.

Palavras-Chave— *Software Defined Networks, Algoritmos Genéticos, Otimização.*

I. INTRODUÇÃO

O crescente ganho de popularidade de serviços e aplicações na Internet tem gerado grandes demandas por infraestrutura e processamento em servidores dispostos em *data centers*, mudando drasticamente os padrões de tráfego de usuários. Redes corporativas têm mudado significativamente, diante dos novos cenários. Entre outras, essas questões motivaram a construção e evolução do paradigma de Redes Definidas por Software (SDN, do inglês *Software Defined Networks*) [15], que separa o controle existente nos dispositivos de rede convencionais para formar um plano de controle logicamente centralizado, possibilitando uma série de novas aplicações.

No contexto de redes SDN, um dos grandes desafios consiste no processo de roteamento. Dada uma requisição de tráfego, o controlador precisa tomar a decisão de qual a rota mais adequada para o fluxo percorrer na rede até chegar em seu destino, considerando requisitos como qualidade de serviço, balanceamento de carga e utilização eficiente de recursos.

Questões como eficiência energética e performance têm sido tópicos amplamente pesquisados pela comunidade científica. Vários trabalhos propõem soluções para aumentar a banda e diminuir a latência das redes, bem como realizar uso eficiente de recursos [17], [3]. Em um sistema de engenharia de tráfego, balanceamento de carga e eficiência energética se caracterizam como objetivos conflitantes. Melhor eficiência energética pode ser alcançada em detrimento de performance, em termos de parâmetros de qualidade [8].

Visando uma solução para otimização de roteamento em redes SDN, o presente trabalho propõe e avalia, por meio de emulação, um algoritmo de otimização multiobjetivo, com

objetivo otimizar rotas para obter melhoria de performance e realizar uso eficiente de recursos.

O texto está estruturado da seguinte forma: na seção II, é apresentada uma revisão de literatura sobre trabalhos relacionados, suas semelhanças e diferenças com a presente proposta. Na seção III, é apresentada a definição matemática do problema a ser resolvido. Na seção IV, é apresentada a modelagem do algoritmo de otimização empregado. Na seção V, é apresentada a arquitetura desenvolvida para o sistema de emulação. A seção VI traz os resultados e discussões acerca dos experimentos realizados e, por fim, a seção VII traz as conclusões obtidas com o trabalho.

II. TRABALHOS RELACIONADOS

No contexto de balanceamento de carga, diversas abordagens têm sido propostas. Em [16], foi proposto um modelo de predição baseado em redes neurais para estimar e escolher caminhos com menor carga para fluxos admitidos na rede. Em [2], foi proposta uma abordagem para distribuir carga entre servidores de um data center, utilizando o algoritmo de Dijkstra para selecionar caminhos menos sobrecarregados, evitando problemas como congestionamento e subutilização de servidores. Em [7], foi desenvolvida uma solução em SDN utilizando ECMP (*Equal Cost Multipath Routing*), com análise de largura de banda em cada enlace antes da atribuição de caminhos para os fluxos, objetivando minimizar congestionamento e atingir melhor performance na rede. Em [5], foi proposto um modelo de busca entre os melhores caminhos entre origem e destino utilizando o algoritmo de Floyd, utilizando como base informações de performance coletadas.

Além de performance, trabalhos na literatura têm despendido esforços também no contexto de eficiência energética. Em [12] foi proposta uma solução de economia de energia para habilitar o mínimo de recursos possíveis em uma rede SDN, baseada nas estratégias SPF (*Shortest Path First*), LPF (*Longest Shortest Path First*), SDF (*Smallest Demand First*) e HDF (*Highest Demand First*), onde foi possível demonstrar que em momentos de pouca de demanda, como à noite, o consumo de energia pode ser reduzido de maneira significativa. Em [1] foi proposta uma heurística para monitorar a rede com fim de alocar recursos de forma eficiente, garantindo ativação somente de elementos necessários para atender o estado da rede. Em [4] foi proposta uma métrica de eficiência energética aplicável ao contexto de SDN e um método de programação inteira, buscando maximizar a eficiência energética da rede.

Nos trabalhos citados anteriormente, foram desenvolvidas soluções com um único objetivo, seja de performance ou de eficiência energética.

Servílio Souza de Assis, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: serviliosouza@ufmg.br; Luciano de Errico, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: l.errico@cpdee.ufmg.br. Este trabalho foi parcialmente financiado pela CAPES.

No que concerne à tomada de decisão no problema de múltiplos objetivos, trabalhos têm sido desenvolvidos para alcançar soluções com bom desempenho e energeticamente eficientes. Em [8], foi apresentada uma estratégia de roteamento baseada no SPEA2, com fim de otimizar a eficiência energética da rede considerando também as restrições de QoS. Em [11], foi proposto um método de otimização de múltiplos critérios com foco em escalabilidade, utilizando decomposição de Lagrange aprimorado por sequências ergódicas, onde soluções são derivadas com base em pesos atribuídos para os critérios.

Diferentemente, no presente trabalho é proposta uma solução para contextos sem SLA (*Service Level Agreement*), cuja metodologia permita derivar um conjunto pareto aproximado de soluções para que regras de tomada de decisão possam escolher ou permutar soluções com base em critérios previamente definidos.

III. DEFINIÇÃO DO PROBLEMA

Dada uma topologia de rede, com m comutadores, n hosts e k demandas de fluxos a serem admitidos, define-se os conjuntos S de comutadores e H de hosts (1).

$$S = \{s_i \mid 1 < i < m\}, \quad H = \{h_i \mid 1 < i < n\} \quad (1)$$

A representação dos enlaces entre comutadores na topologia é apresentada pela matriz T (2), onde t_{ij} é definido como 1 se s_i conecta s_j ou i é igual a j , e -1 caso contrário.

$$T [t_{ij}]_{m \times m} \quad (2)$$

Cada host $h_i \in H$ está conectado em um comutador $s_j \in S$, ao qual é obtido através da relação c (3), que é sustentada por uma matriz constante $m \times n$.

$$h \rightarrow c(h), \quad c: H \rightarrow S \quad (3)$$

Dada uma demanda por tráfego entre dois hosts h_i e h_j , faz-se necessário determinar um caminho entre os comutadores $c(h_i)$ e $c(h_j)$. Assim, define-se um conjunto F de pares de comutadores de origem e destino determinados pelas demandas a serem admitidas na rede, conforme (4).

$$F [f_i]_k, \quad f_i \in S^2 \quad (4)$$

Para uma instância do problema, uma solução Z consiste em um conjunto de listas de nós intermediários para cada fluxo, sendo estas um subconjunto do complemento de nós comutadores em relação aos nós de origem e destino, conforme (5).

$$Z [z_i]_k, \quad z_i \subseteq S - f_i \quad (5)$$

Dada uma solução Z , a matriz U (6) determina a quantidade de fluxos que passam em cada enlace entre comutadores na rede, onde os índices representam os nós de origem e destino.

$$U = [u_{ij}]_{m \times m} \quad (6)$$

As funções objetivo do problema têm como base de cálculo a matriz U . O primeiro objetivo consiste no balanceamento de carga, cuja medida consiste na dispersão dos valores de

utilização de cada enlace por cada fluxo de comunicação admitido na rede, que pode ser obtida pela variância da matriz. Este valor é igual a zero quando a rede tem a um estado de equidade de utilização de enlaces. O segundo objetivo foi definido como eficiência energética através de uso de poucos enlaces, cuja medida se dá pelo percentual de utilização de enlaces na rede. Nesse sentido, os objetivos do processo de otimização são apresentados em (7) e (8).

$$\min \text{VAR}(U) \quad (7)$$

$$\min \sum_{i=1}^m \sum_{j=i+1}^m \begin{cases} 1 & u_{ij} > 0 \\ 0 & \text{outro caso} \end{cases} \quad (8)$$

IV. ALGORITMO DE OTIMIZAÇÃO

Para implementação do algoritmo de otimização para o problema, foi escolhido como base o NSGA-II (*Nondominated Sorting Genetic Algorithm II*) [6], cujo principal objetivo é encontrar um conjunto pareto de soluções em uma única execução. Esse algoritmo é amplamente conhecido por apresentar uma abordagem de rápida ordenação por não-dominância, mantendo diversidade entre as soluções e utilizando elitismo para preservar as melhores soluções ao longo das gerações.

A definição de uma instância do problema consiste em uma matriz de topologia e um conjunto de pares de demandas. A Figura 1 apresenta um exemplo de instância do problema.

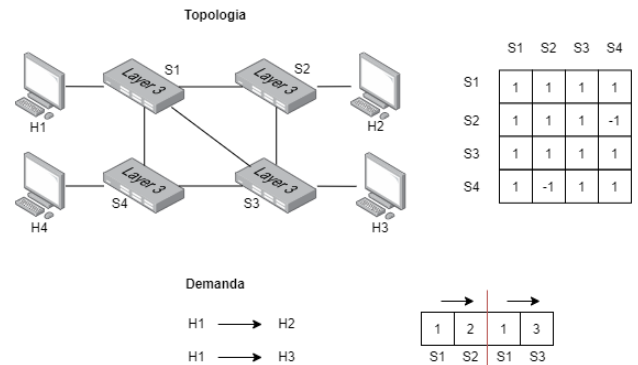


Fig. 1. Representação de uma instância do problema.

Para representação de uma solução para o problema, é criada uma lista de tamanho k , com vetores de tamanho $m - 2$ para compor os possíveis nós intermediários entre a origem e o destino. Como os vetores são definidos com tamanho estático, o valor -1 é utilizado para representar ausência de nó intermediário. Assim, a solução para uma demanda deve ser um subconjunto de todos os outros switches e valores unitários negativos. A Figura 2 apresenta um exemplo de representação.

A representação não aloca variáveis para demandas em que os hosts de origem e destino se encontram conectados em um mesmo switch. No processo de inicialização da população, as soluções são criadas selecionando um subconjunto aleatório do conjunto de valores possíveis para cada uma das demandas.

Para o operador de cruzamento, foi utilizado um ponto de corte para cada caminho presente na solução: o filho recebe os primeiros valores de um pai até tal ponto e, em seguida,

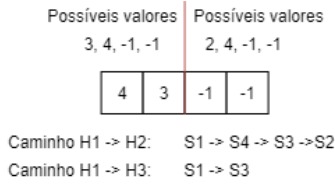


Fig. 2. Representação de uma solução para o problema.

recebe os valores do outro pai, ignorando valores se forem nós já adicionados. Ao final, valores negativos são adicionados se necessário.

Para operação de mutação, o processo seleciona um caminho dentro de uma solução e em seguida seleciona uma posição aleatória. A diferença é calculada entre os valores possíveis para aquele caminho e os valores já presentes nele. Assim, é feita uma troca de valores entre uma posição aleatória do caminho e um valor aleatório do conjunto diferença. A Figura 3 apresenta um exemplo visual de operações de cruzamento e mutação.

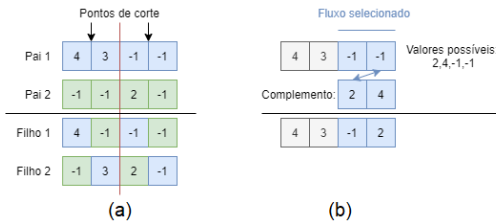


Fig. 3. Operações genéticas. Em (a) um cruzamento e em (b) uma mutação.

Ao aplicar os operadores genéticos, soluções ineficazes podem ser geradas, sugerindo caminhos por enlaces que não existem, o que seria o caso, no exemplo da Figura 2, se a primeira solução estivesse com os valores 3 e 4, sugerindo que a demanda de $H1$ para $H2$ passe pelos switches $S1$, $S3$, $S4$ e $S2$ em sequência, sendo que o salto de $S4$ para $S2$ não é topologicamente possível. Esses casos são tratados com penalizações nos valores das funções de avaliação, fazendo-se com que a própria pressão de seleção elimine, ao longo das gerações do algoritmo, tais soluções.

O cálculo dos valores das funções objetivo é obtido através de uma heurística que processa a solução em função da topologia, gerando a matriz U e computando os valores de dispersão e utilização de enlaces.

O algoritmo genético tem os parâmetros de taxas de cruzamento e mutação, tamanho de população e número de gerações, e a cada geração realiza seleção de pais por torneio, operações genéticas, e substituição utilizando o *fast non-dominated sort* e ordenação por *crowd distance* quando necessário. O critério de parada foi definido como o número máximo de avaliações de solução, que consiste no produto entre número de gerações e tamanho da população.

V. ARQUITETURA DO SISTEMA

A topologia de testes utilizada para realização dos experimentos foi definida como uma topologia mesh, similar à apresentada na Figura 1, se diferenciando pela presença de enlaces

interligando todos os switches e havendo três hosts conectados em cada um dos mesmos. Uma vez que o algoritmo proposto tem a capacidade de explorar diversos caminhos buscando soluções para as demandas, a topologia totalmente conectada foi adotada por proporcionar maior espaço de busca e recursos a serem explorados.

A emulação foi feita através do Mininet [9], definindo uma rotina de experimentos através de um script específico em Python. O ambiente foi criado para executar um cenário de rede por um período de 120 segundos, adicionando uma nova demanda por fluxo de dados TCP a cada 20 segundos, gerado com IPERF¹. O script de experimentos foi programado para coletar dados de utilização de enlace, banda e RTT (*Round Trip Time*).

O experimento foi realizado em três diferentes cenários utilizando *openvswitches* para o processo de *forwarding*. No primeiro cenário, foi utilizado o modo *standalone* com protocolo *spanning-tree*. Nos outros dois cenários, foi utilizado o modo OpenFlow [13], onde o controlador é consultado pelos switches para atribuição de regras de encaminhamento, utilizando o algoritmo genético. Em cada caso, para seleção da solução final dentro dos paretos aproximados, um dos objetivos (balanceamento ou minimização de enlaces) foi priorizado, atribuindo-se a este o peso 0,9 e ao outro o peso 0,1.

A literatura apresenta vários controladores SDN. O Ryu [14] é um controlador desenvolvido em Python, com arquitetura centralizada, a qual é mais utilizada em redes de baixa escala. O controlador apresenta suporte para as versões do OpenFlow 1.0 a 1.5, sendo uma excelente alternativa indicada para aplicações de pesquisa [10]. Trabalhos comparativos têm mostrado que o Ryu apresenta boas avaliações em características como suporte a switches virtuais, modularidade, documentação, compatibilidade de linguagem de programação e disponibilidade de interface de usuário. Além disso, em termos de performance, o controlador apresenta comportamento consistente [18]. Desta forma, para o desenvolvimento do aplicativo de otimização de rotas foi utilizado o controlador Ryu.

Na aplicação desenvolvida, o controlador realiza revisão periódica do estado topológico da rede utilizando protocolo LLDP e define operações com base em dois eventos principais: conexão de switch e entrada de pacotes. Na conexão, após o *handshaking openflow*, é instalada uma regra de encaminhamento de pacote para o controlador, com prioridade mínima. Dessa maneira, um pacote é encaminhado ao controlador quando não há match das informações do pacote com alguma regra de encaminhamento. Quando o controlador recebe um pacote, este então adiciona uma regra definida para o fluxo em questão em cada um dos switches no caminho e retorna o pacote. A Figura 4 apresenta os fluxos de processamento de levantamento de topologia e reação aos eventos trabalhados.

As regras de encaminhamento fazem match com endereços MAC de origem e destino dos pacotes e são adicionadas ao switch com validade de quatro segundos. Assim, periodicamente, a tabela é atualizada. Ao receber um pacote de entrada,

¹<https://iperf.fr/>

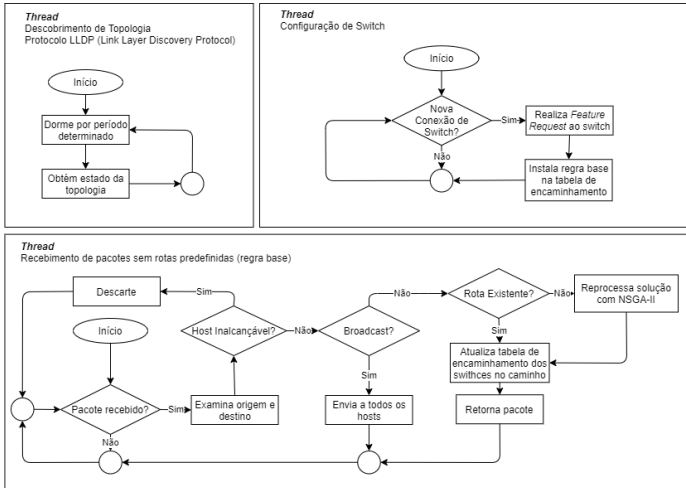


Fig. 4. Arquitetura do aplicativo do controlador.

o controlador verifica se existe uma rota determinada para o fluxo entre os hosts envolvidos. Caso não exista, um novo processo de otimização é executado para definir um conjunto de rotas para o novo fluxo, alterando as rotas dos fluxos correntes se necessário, e atualizar as regras para o fluxo relacionado ao pacote recebido.

VI. RESULTADOS

No processo de otimização com múltiplos critérios, inicialmente várias instâncias do problema foram analisadas, variando a topologia da rede e a quantidade de fluxos a serem adotados. A Figura 5 apresenta o conjunto inicial e a aproximação do pareto para uma instância determinada por topologia de 15 switches totalmente ligados entre si e 44 fluxos a serem alocados. Para esta instância, foram escolhidos parâmetros com melhores resultados de convergência e derivação da fronteira pareto aproximada de forma empírica, ficando o tamanho de população de 80, taxa de mutação de 0,25, taxa de cruzamento de 0,75 e um total de 250 gerações.

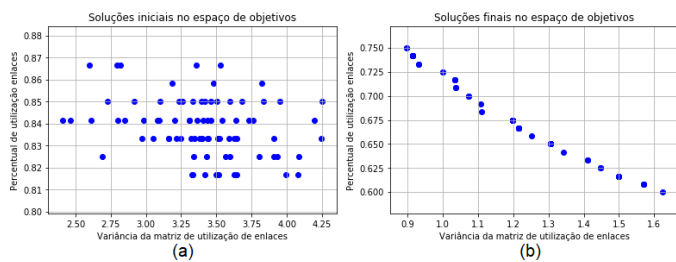


Fig. 5. Resultados obtidos para uma instância do problema de atribuição de rotas. Em (a) têm-se uma população inicial de soluções no espaço de objetivos, e em (b) a fronteira pareto aproximada.

Os testes sugeriram fronteiras pareto convexas para o problema, sendo o pareto, em instâncias para topologias com ausência de enlaces, definido como subconjunto do pareto de soluções para topologia totalmente conectada. O espaço de busca cresce fatorialmente com o aumento do número de vértices comutadores e da quantidade de demandas a serem atendidas, fazendo a convergência ficar mais lenta.

Através da atribuição de pesos aos objetivos, as soluções candidatas são reavaliadas por agregação dos valores e então ordenadas para escolha de uma solução final. Dispondo de um conjunto de soluções, é possível intercambiar as soluções sem nova realização de otimização.

Para emulação da rede, os parâmetros do algoritmo foram também definidos de forma empírica, ficando 10 para tamanho de população, 25 gerações e taxas de cruzamento e mutação de 0,85 e 0,15, respectivamente. Foi executada uma emulação para cada uma das três configurações trabalhadas. As Figuras 6 e 7 apresentam respectivamente as médias de vazões e de valores de RTT experimentados pelos fluxos presentes em uma execução de cada um dos três cenários.

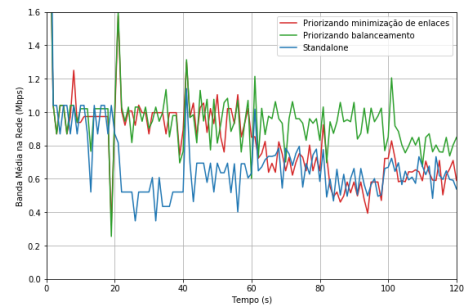


Fig. 6. Largura de banda média alcançada pelos fluxos admitidos na rede em função do tempo em uma execução experimental de cada cenário.

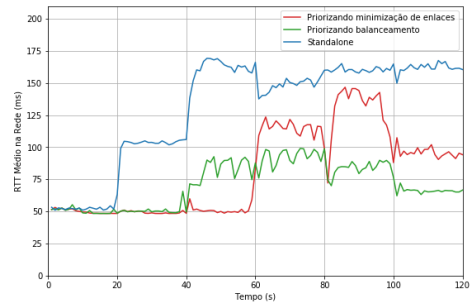


Fig. 7. RTT médio experimentado pelos fluxos da rede em função do tempo em cada um dos cenários reproduzidos.

Nos momentos iniciais da emulação, a diferença é pequena, uma vez que o espaço de busca é pequeno e a fronteira pareto aproximada é concentrada em um padrão de rotas. Além disso, há apenas uma demanda e ainda não há disputa por recursos. Com o aumento do número de demandas na rede, as soluções começam a divergir, se observando maiores valores de vazão e menores valores de RTT para soluções que priorizam balanceamento, onde a utilização de recursos é maior, diminuindo os efeitos da concorrência pelos mesmos. Além da vazão, o número de enlaces entre comutadores utilizados na rede também foi capturado para avaliação, conforme apresentado na Figura 8, onde é possível verificar que soluções mais balanceadas proporcionam maior vazão, ao custo de alocar mais recursos para atender o conjunto de demandas na rede, à medida que a capacidade demandada aumenta. Os valores médios para as variáveis analisadas são apresentados na Tabela I.

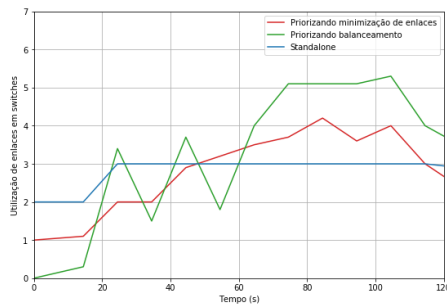


Fig. 8. Número de enlaces utilizados na rede ao longo do tempo em cada um dos cenários reproduzidos.

TABELA I
RESULTADOS MÉDIOS OBTIDOS.

Cenário	Vazão	RTT	Utilização de enlaces
Balanceamento	0,94 Mbps	71,09 ms	3,28
Eficiência Energética	0,81 Mbps	81,93 ms	2,80
Standalone	0,68 Mbps	131,62 ms	2,83

Os resultados obtidos para uma rede em modo *standalone* tendem a ser próximos de soluções no algoritmo genético que priorizam mais o objetivo de minimização de utilização de enlaces, à medida em que mais fluxos são admitidos na rede.

Com a minimização de utilização de enlaces, menores caminhos são adotados para atender as demandas, priorizando também a utilização de enlaces e switches já utilizados. Assim, o uso mínimo de enlaces e switches diminui a energia consumida pelos equipamentos, que é dado pela soma dos produtos entre as potências dos equipamentos em modo de operação e o tempo em que permanecem ativados.

No controlador, a periodicidade de 4 segundos para atualização da tabela de fluxos pode causar uma elevação indesejada na carga de trabalho, em cenários de alto número de demandas, prejudicando a escalabilidade. Um alternativa para proporcionar escalabilidade seria retirar o tempo de duração da regra, e enviar comandos de exclusão de entradas nas tabelas de encaminhamento dos switches ao detectar mudança de rotas para demandas na rede, antes de atribuir novas regras.

O processo de otimização realizado experimentalmente pôde priorizar performance e eficiência energética de acordo com pesos de importância, os quais podem variar de acordo com caracterização da demanda na rede e por mudanças de bandeira tarifárias em horários de pico de consumo energético.

VII. CONCLUSÕES

No âmbito de redes SDN, balanceamento de carga e eficiência energética são questões relevantes. O presente trabalho propôs um modelo de otimização, baseado no NSGA-II, para gerar soluções de rotas visando esses objetivos.

Experimentos realizados mostraram que a técnica proposta pode melhorar performance na rede, bem como minimizar utilização de recursos, de acordo com preferências colocadas.

Como trabalhos futuros, podem ser adicionados na modelagem mais objetivos e restrições, como por exemplo garantia

de QoS e priorização de tráfego. Além disso, outra questão importante é aprimorar a escalabilidade da arquitetura do sistema para cenários de topologias grandes e altas demandas.

AGRADECIMENTOS

O presente trabalho foi integralmente realizado no Laboratório de Redes, Tecnologias e Serviços de Comunicação a Distância (LabCOM) da Universidade Federal de Minas Gerais e teve o apoio financeiro da CAPES, a quem os autores agradecem.

REFERÊNCIAS

- [1] Al-musawi, Ranya; Al Khatib, Obada. "Optimized energy aware resource allocation algorithm using software defined network technology". Journal of Telecommunications and Information Technology, n. 2, p. 83-91. 2019.
- [2] Ali, Tariq Emad; Morad, Ameer Hussein; Abdala, Mohammed A. "Load balance in data center SDN networks". International Journal of Electrical and Computer Engineering, v. 8, n. 5, p. 3086, 2018.
- [3] Assefa, Beakal Gizachew; Özkasap, Öznur. "A survey of energy efficiency in SDN: Software-based methods and optimization models". Journal of Network and Computer Applications, v. 137, p. 127-143, 2019.
- [4] Assefa, Beakal Gizachew; Özkasap, Öznur. "A novel utility based metric and routing for energy efficiency in software defined networking". In: 2019 International Symposium on Networks, Computers and Communications (ISNCC). IEEE, 2019. p. 1-4.
- [5] Chakravarthy, V. Deeban; Amutha, B. "Path based load balancing for data center networks using SDN". International Journal of Electrical and Computer Engineering (IJECE), v. 9, n.4, p. 3279-3285, 2019.
- [6] Deb, Kalyanmoy et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II". IEEE transactions on evolutionary computation, v. 6, n. 2, p. 182-197, 2002.
- [7] Dewanto, Ramadhika; Munadi, Rendy; Negara, Ridha Muldina. "Improved Load Balancing on Software Defined Network-based Equal Cost Multipath Routing in Data Center Network". JURNAL INFOTEL, v. 10, n. 3, p. 157-162, 2018.
- [8] Fernández-Fernández, Adriana; Cervelló-Pastor, Cristina; Ochoa-Aday, Leonardo. "A multi-objective routing strategy for QoS and energy awareness in software-defined networks". IEEE Communications Letters, v. 21, n. 11, p. 2416-2419, 2017.
- [9] Handigol, Nikhil et al. "Reproducible network experiments using container-based emulation". In: Proceedings of the 8th international conference on Emerging networking experiments and technologies. 2012. p. 253-264.
- [10] Islam, Md Tariqul; Islam, Nazrul; Al Refat, Md. "Node to Node Performance Evaluation through RYU SDN Controller". Wireless Personal Communications, v. 112, p. 555-570, 2020.
- [11] Jaglarz, Piotr et al. "Enhanced Lagrange Decomposition for multi-objective scalable TE in SDN". Computer Networks, v. 167, p. 106992, 2020.
- [12] Markiewicz, Adam; Tran, Phuong Nga; Timm-Giel, Andreas. "Energy consumption optimization for software defined networks considering dynamic traffic". In: 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet). IEEE, 2014. p. 155-160.
- [13] Mckeown, Nick et al. "OpenFlow: enabling innovation in campus networks". ACM SIGCOMM Computer Communication Review, v. 38, n. 2, p. 69-74, 2008.
- [14] Nippon Telegraph and Telephone Corporation. "RYU network operating system", 2012. [Online], <http://osrg.github.com/ryu>
- [15] Nunes, Bruno Astuto A. et al. "A survey of software-defined networking: Past, present, and future of programmable networks". IEEE Communications surveys tutorials, v. 16, n. 3, p. 1617-1634, 2014.
- [16] Rupani, Kunal et al. "Dynamic Load Balancing in Software-Defined Networks Using Machine Learning". In: Proceeding of International Conference on Computational Science and Applications. Springer, Singapore, 2020. p. 283-292.
- [17] Shirmarz, Alireza; Ghaffari, Ali. "Performance issues and solutions in SDN-based data center: A survey". The Journal of Supercomputing, v. 76, p. 7545-7593, 2020.
- [18] Zhu, Liehuang et al. "SDN controllers: Benchmarking & performance evaluation". CoRR abs/1902.04491, 2019. <http://arxiv.org/abs/1902.04491>