

MGTD: Um Mecanismo de Gerenciamento de Tráfego Dinâmico Ciente do Consumo de Energia

Wallace Nascimento Paraízo, Luciano Jerez Chaves, Edelberto Franco Silva,
Eduardo Pagani Julio e Alex Borges Vieira

Resumo— Computação verde é tema central em diversas áreas da computação, incluindo redes de computadores. Soluções dinâmicas capazes de ajustar adequadamente os recursos de uma rede podem evitar o superdimensionamento da infraestrutura e reduzir o consumo energético em momentos de baixa demanda. Neste trabalho, apresentamos o MGTD: um Mecanismo para Gerenciamento de Tráfego Dinâmico ciente do consumo de energia, construído sobre o paradigma de SDN. O MGTD monitora continuamente o uso dos enlaces para concentrar os tráfegos e desligar equipamentos ociosos, sem degradar a qualidade do serviço oferecido. Nossas simulações evidenciam economia média de energia de até 46% em cenário de baixa demanda. Comparando com uma solução típica da literatura, o MGTD é 4% a 7% melhor na média, dependendo do cenário.

Palavras-Chave— Computação verde, Consumo de energia, Redes Definidas por Software, Gerenciamento de tráfego.

Abstract— Green computing is a central subject in many computer science areas, including computer networks. Dynamic solutions that can properly adjust network resources can prevent the infrastructure overprovision and mitigate power consumption during low demand periods. In this work, we propose the MGTD, an energy-aware dynamic mechanism for traffic management, built upon the SDN paradigm. The MGTD continuously monitors the use of network link to concentrate traffic and disconnect idle equipment without degrading the offered quality of service. Our simulations show that the mechanism can save up to 46% of energy on average. When compared to an existing solution, MGTD is, on average, 4% to 7% better, depending on the evaluated scenario.

Keywords— Green computing, Power consumption, Software Defined Networking, Traffic management.

I. INTRODUÇÃO

Computação ciente do consumo de energia é tema central de pesquisa hoje em dia, inclusive em redes de computadores. De fato, muitas vezes os recursos de rede são superdimensionados para lidar com sobrecargas de tráfego, resultando em desperdício de recursos e alto consumo de energia. Parte disso se deve ao engessamento das redes legadas, onde a estabilidade é crucial para o sucesso dos negócios. Como o controle e o encaminhamento são indissociáveis e embutidas nos elementos de rede, fica praticamente inviável a adoção de novas tecnologias e protocolos nos ambientes em produção [1].

Por outro lado, o paradigma de Redes Definidas por Software (SDN) faz uma clara separação entre os planos de dados e controle. Sua visão centralizada e fácil programabilidade

permitem um monitoramento acurado da rede e o rápido roteamento dos fluxos. Nessa linha, SDN pode auxiliar no desligamento (ou hibernação) de dispositivos de redes como forma de economizar energia. Comutadores SDN convergem para um estado estacionário, onde as rotas de entrada e saída são bem conhecidas e não mudam, mais rapidamente que dispositivos L2/L3 clássicos [2]. Fluxos de dados podem ser redirecionados para que dispositivos de rede ociosos possam ser desligados. Tão logo se percebe picos de carga na rede, basta religar os enlaces/dispositivos inativos.

Note que há diversas maneiras de se economizar energia em redes como, por exemplo, escalar a frequência dos processadores dos dispositivos ou ajustar a velocidade dos enlaces [3]. Entretanto, o suporte a esses mecanismos não é comumente ofertado por dispositivos comerciais. Até mesmo iniciativas com o IEEE 802.3az excluíram de sua especificação final funcionalidades para configuração de diferentes velocidades em enlaces *Ethernet* por conta de desafios de implementação [4]. Nesse sentido, a maioria das soluções existentes concentram esforços em otimizar o roteamento do tráfego pela rede.

Neste trabalho apresentamos o MGTD: um Mecanismo de Gerenciamento de Tráfego Dinâmico ciente do consumo de energia (seção IV). O MGTD emprega SDN para melhorar o roteamento de tráfego entre os comutadores. De forma breve, o mecanismo monitora os fluxos de rede e propõe o uso do menor número possível de rotas para atender tais fluxos, viabilizando o desligamento de enlaces e dispositivos ociosos. Mais ainda, para evitar que desligamentos ocorram recorrentemente, diminuindo a vida útil dos equipamentos, o mecanismo utiliza uma política de agregação de tráfego. Entretanto, essa política só agrega fluxos, permitindo o desligamento de enlaces, caso o nível de utilização do enlace não indique a necessidade de religamento em um futuro muito próximo.

Ressaltamos que os trabalhos da literatura focam na otimização da topologia e na minimização do número de enlaces e comutadores, mas não consideram a dinamicidade dos fluxos [2], [5]–[7]. De fato, apenas trabalhos como [8] e [2] exploram a sazonalidade do tráfego no tempo. Mesmo assim, suas soluções são por vezes computacionalmente custosas.

Avaliamos o MGTD em um ambiente SDN de uma topologia típica de campus universitário (seção V). Consideramos diferentes níveis de tráfego, de redes com baixa demanda às sobrecarregadas. Os resultados mostram uma economia de energia média de 17% a 46%. Comparando com um trabalho do estado da arte, nossos resultados são, em média, 7% e 4% melhores, para os cenários de média e alta demanda, respectivamente.

Wallace Nascimento Paraízo, Luciano Jerez Chaves, Edelberto Franco Silva, Eduardo Pagani Julio e Alex Borges Vieira. Departamento de Ciência da Computação, Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil. E-mails: wallace.paraizo@ufjf.edu.br, luciano.chaves@ice.ufjf.br, edelberto@ice.ufjf.br, eduardo.pagani@ice.ufjf.br e alex.borges@ufjf.edu.br.

II. TRABALHOS RELACIONADOS

Tradicionalmente, trabalhos para economia de energia em redes de computadores focam em modelos matemáticos da topologia e na minimização de atributos (*e.g.* do número de enlaces, comutadores ou frequência da CPUs) [2], [5]–[7]. Os autores de [9] formulam um modelo de otimização para minimizar enlaces e ativos de rede e posicionar adequadamente o controlador. Segundo eles, essas são questões chave para a economia de energia. Muitas dessas formulações envolvem heurísticas, pois a otimização de recursos nesse ambiente é um problema NP-Completo. Os autores de [6] e [7] propõem heurísticas para o desligamento de placas de redes de enlaces agregados que podem ser desativadas de forma independente. Os autores não desligam equipamentos nem enlaces inteiros pois acreditam que isso reduz a conectividade da rede. Em [10], os autores consideram a largura de banda, carga de enlaces e matrizes de tráfego para hibernar comutadores, reorganizando a topologia da rede sempre que preciso.

Ao invés de reorganizar a topologia de rede e desligar ativos, os autores de [5] trabalham ajustando a velocidade de enlaces subutilizados. Tráfegos com menor prioridade são redirecionados e os tráfegos de tempo real são mantidos no caminho mínimo para satisfazer a QoS desejada. Note que a redução no consumo de energia é alcançada somente com a redução das velocidades dos enlaces, resultando apenas em um ganho marginal. Além disso, a redução na velocidade de enlaces não é comumente suportada em dispositivos comerciais.

Poucos trabalhos consideram a dinamicidade da rede. Os autores de [8] propõem o *ElasticTree*, um gerenciador de energia de rede para *datacenter* que monitora o tráfego e escolhe o conjunto de elementos de rede que devem permanecer ativos para atender às metas de desempenho e tolerância a falhas. O *ElasticTree* desliga ao máximo enlaces e comutadores desnecessários. Para isso, são utilizados um modelo formal, um algoritmo guloso, uma heurística sensível à topologia e métodos de previsão. Na mesma linha, os autores de [2] apresentam um modelo e uma heurística gulosa para encontrar um caminho mínimo em relação ao número de saltos e acréscimo no consumo de energia. Assim, os autores consideram que os demais ativos de rede podem ser desligados.

Diferente do MGTD, a maioria dos trabalhos existentes não considera a dinamicidade das redes. Esses trabalhos apresentam soluções custosas que dependem de mecanismos especializados. Mais ainda, apenas reavaliam a otimização/heurística proposta frente às mudanças nos fluxos, mas não consideram uma suavização na interferência nos ativos de rede e nem possíveis agregações de tráfego. Assim, só desligam enlaces e comutadores quando se tornam naturalmente ociosos.

III. CENÁRIO CONSIDERADO

Neste trabalho, consideramos redes com topologia arbitrária. Na borda da rede, os nós hospedeiros representam os elementos responsáveis pela geração e consumo dos fluxos de dados. Os hospedeiros estão ligados à comutadores de acesso, que por sua vez estão conectados aos comutadores de encaminhamento. Todos os comutadores são compatíveis com o protocolo OpenFlow, que é a plataforma SDN mais utilizada

atualmente, tanto em desenvolvimentos como em pesquisa [1]. Os comutadores são responsáveis pelo encaminhamento de pacotes entre os enlaces, de acordo com as regras de fluxo previamente configuradas por um ou mais controladores, responsáveis pela lógica centralizada de controle da rede.

Essa topologia é modelada como um grafo $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, sendo $\mathcal{N} = \mathcal{N}_h \cup \mathcal{N}_c$ o conjunto de nós e \mathcal{L} o conjunto de ligações (enlaces). O subconjunto \mathcal{N}_h corresponde aos nós hospedeiros, enquanto o subconjunto \mathcal{N}_c corresponde aos nós comutadores. Especificamente, $\mathcal{N}_c = \mathcal{N}_a \cup \mathcal{N}_e$, sendo \mathcal{N}_a os nós comutadores de acesso, para os quais $\exists l_{ci} \in \mathcal{L}$ conectando o comutador c diretamente à um hospedeiro $i \in \mathcal{N}_h$; e \mathcal{N}_e os nós comutadores de encaminhamento, onde $\forall l_{cj} \in \mathcal{L}$ conecta o comutador c diretamente à outro comutador $j \in \mathcal{N}_c$.

O MGTD reduz o consumo energético através do desligamento seletivo de portas em comutadores que estejam com seus enlaces ociosos. Se todas as portas de um comutador forem desligadas, o comutador também pode ser desligado. Para isso, os algoritmos de gerenciamento de tráfego precisam concentrar os fluxos em uma quantidade mínima de enlaces, mas sem degradar a qualidade do serviço oferecido. Essa estratégia é denominada de consolidação de recursos, segundo [11].

Para todo comutador $c \in \mathcal{N}_c$, denotamos como $E_t(c)$ o consumo energético de c no instante t segundo a equação 1 [5]:

$$E_t(c) = Eb_t(c) + Ep_g(c) \times Np_{gt}(c) + Ep_f(c) \times Np_{ft}(c) \quad (1)$$

onde $Eb_t(c)$ indica o consumo base de c no tempo t , necessário para manter o comutador em funcionamento (processador, refrigeração, etc.); $Ep_g(c)$ e $Ep_f(c)$ representam o consumo de cada porta 1 Gbps e 100 Mbps em c , respectivamente; $Np_{gt}(c)$ e $Np_{ft}(c)$ representam o número de portas 1 Gbps e 100 Mbps ativas em c no instante t , respectivamente. Note que o consumo energético da porta varia de acordo com sua velocidade [2]. Assim, pode-se estimar o consumo energético total da rede $E_t(\mathcal{G})$ num instante t através da equação 2:

$$E_t(\mathcal{G}) = \sum_{c \in \mathcal{N}_c} E_t(c) \quad (2)$$

IV. GERENCIAMENTO DE TRÁFEGO DINÂMICO

O MGTD (Mecanismo de Gerenciamento de Tráfego Dinâmico) é ciente do consumo de energia e possui três componentes: o *monitoramento ativo da rede*, que mantém um modelo atualizado do uso dos recursos na rede; o algoritmo para *instalação de novos fluxos*, que identifica reativamente novos tráfegos e aloca rotas apropriadas; e o algoritmo para *redirecionamento de fluxos ativos*, que agrega fluxos na menor quantidade de enlaces, de modo a desligar os recursos ociosos, mas sem causar sobrecarga nos que permanecerem ligados.

A. Monitoramento ativo da rede

É através do monitoramento ativo que o controlador do MGTD mantém uma visão detalhada e atualizada da rede, incluindo informações sobre o uso dos recursos e o consumo energético dos elementos ativos. Como o MGTD pode desligar e religar enlaces e comutadores ao passar do tempo, denotamos como $\mathcal{G}'_t \subseteq \mathcal{G}$ o conjunto de nós e enlaces que estejam ligados e disponíveis para uso em um instante t . Para manter

o modelo atualizado, o controlador envia periodicamente aos comutadores mensagens OpenFlow requisitando por informações e estatísticas. O intervalo Δt entre as requisições pode ser ajustado para um bom compromisso entre a atualização do modelo e a sobrecarga na rede. Com as respostas é possível atualizar \mathcal{G}'_t , a estimativa de consumo energético $E_t(\mathcal{G})$, a vazão instantânea dos fluxos de dados e também a taxa de utilização dos enlaces da rede. Essas informações são posteriormente utilizadas pelos algoritmos de instalação de novos fluxos e de redirecionamento de fluxos ativos.

Seja $U_t(l)$ a taxa de utilização de um enlace $l \in \mathcal{L}$, calculado pela razão da banda passante no instante t pela capacidade de transmissão de l . Esse indicador é usado para classificar o enlace em um estado $S_t(l)$, de acordo com os limiares $T_{0 \rightarrow 1}$, $T_{1 \rightarrow 2}$ e $T_{2 \rightarrow 3}$ indicados na tabela I. Essa categorização é usada pelos algoritmos para impedir que enlaces sobrecarregados recebam novos fluxos, evitando perda de pacotes. Além disso, enlaces com baixa carga mostram-se candidatos favoráveis para o desligamento de portas nos comutadores. Assim, o mecanismo visa realocar todos os fluxos destes enlaces para rotas alternativas, tornando-os ociosos e economizando energia.

Com o modelo da rede atualizado, o controlador constrói o conjunto \mathcal{P}_{ij} com os k caminhos mínimos conectando o hospedeiro i ao j , $\forall i, j \in \mathcal{N}_h$. O controlador também calcula, $\forall c \in \mathcal{N}_c$, o indicador $Q(c)$ representando o total de caminhos nos conjuntos \mathcal{P} que passam pelo comutador c . Os caminhos mínimos e o indicador $Q(c)$ fazem sentido quando vistos sob a ótica da economia de energia, pois quanto menor e mais concentrados são os caminhos, menos elementos de rede são utilizados e mais comutadores podem ser desligados.

TABELA I: Estado $S_t(l)$ em função da utilização $U_t(l)$.

$S_t(l)$	Estado do enlace	Taxa de utilização $U_t(l)$
s_0	baixa carga	$0\% \leq U_t(l) < T_{0 \rightarrow 1}$
s_1	normal	$T_{0 \rightarrow 1} \leq U_t(l) < T_{1 \rightarrow 2}$
s_2	alta carga	$T_{1 \rightarrow 2} \leq U_t(l) < T_{2 \rightarrow 3}$
s_3	sobrecarga	$T_{2 \rightarrow 3} \leq U_t(l) \leq 100\%$

B. Instalação de novos fluxos

A instalação de novos fluxos pelo controlador é reativa. Seja f_{ab} um fluxo entre os hospedeiros $a, b \in \mathcal{N}_h$. Quando o primeiro pacote de f_{ab} alcançar o comutador de acesso, o controlador é notificado e inicia a busca pelo melhor caminho $p_{ab}^m \in \mathcal{P}_{ab}$ com todos os enlaces em estado $S_t(l) \leq s_{max}$, conforme algoritmo 1. A busca inicial é restrita aos caminhos que não utilizam enlaces sobrecarregados ($s_{max} = s_2$). O algoritmo prioriza o caminho ideal p_i que possa ser acomodado em \mathcal{G}'_t e não resulte em incremento no consumo de energia. No caso de vários caminhos ideais, se prioriza o de menor comprimento $|p_i|$, tendo como critério de desempate o indicador $Q(p_i) = \sum_{c \in p_i} Q(c)$. Caso não exista $p_i \in \mathcal{G}'_t$, então avalia-se o impacto energético $E_i(p)$, $\forall p \in \mathcal{P}_{ab}$, e escolhe-se o caminho p_e que resultar no menor incremento energético final. Neste caso, será preciso atualizar \mathcal{G}'_t , ativando enlaces ou comutadores para acomodar o novo fluxo.

Eventualmente, pode não existir um caminho p_{ab}^m sem que algum enlace sobrecarregado seja utilizado. Nesse caso,

relaxa-se a restrição sobre o estado máximo dos enlaces ($s_{max} = s_3$), e realiza-se uma nova busca por p_{ab}^m . Uma vez que o caminho esteja definido para f_{ab} , as regras de fluxo são instaladas em todos os comutadores $c \in p_{ab}^m$.

Algoritmo 1: BUSCACAMINHO

Entrada: Caminhos \mathcal{P}_{ab} ; Subgrafo ativo \mathcal{G}'_t ; Estado s_{max} .
Saída: Melhor caminho $p_{ab}^m \in \mathcal{P}_{ab}$ com enlaces $S_t(l) \leq s_{max}$.

- 1 $p_i \leftarrow \infty$; $p_e \leftarrow \infty$;
- 2 **para todo** $p \in \mathcal{P}_{ab}$ **faça**
- 3 **se** $\forall l \in p / S_t(l) \leq s_{max}$ **então**
- 4 **se** $p \in \mathcal{G}'_t$ **então**
- 5 **se** $(|p| < |p_i|)$ **ou** $(|p| = |p_i| \text{ e } Q(p) > Q(p_i))$ **então**
- 6 $p_i \leftarrow p$;
- 7 **se** $E_i(p) < E_i(p_e)$ **então**
- 8 $p_e \leftarrow p$;
- 9 **se** $p_i \neq \infty$ **então retorna** p_i **senão retorna** p_e ;

C. Redirecionamento de fluxos ativos

A instalação de novos fluxos considera a taxa de utilização instantânea dos enlaces. Entretanto, o comportamento do tráfego ao longo do tempo pode gerar utilização indesejada em alguns enlaces. Nesse caso, o algoritmo de redirecionamento de fluxos ativos entra em ação para redistribuir os tráfegos pela rede. Para isso, ele avalia o estado $S_t(l)$ que cada enlace pode assumir, conforme a figura 1. Entende-se que fluxos alocados em um enlace no estado de sobrecarga $S_t(l) = s_3$ poderão ser penalizados com atrasos maiores e perdas de pacotes. O algoritmo redireciona deste para outros enlaces a menor quantidade de fluxos necessária para que a taxa de utilização diminua para algum estado $S_t(l) < s_3$. Essa estratégia evita que os enlaces permaneçam sobrecarregados, reduzindo eventuais prejuízos aos indicadores de QoS da rede.

Por outro lado, entende-se como subutilizado os enlaces no estado de baixa carga $S_t(l) = s_0$. Nesse caso, o algoritmo tenta redirecionar todos os fluxos deste para outros enlaces na rede, com o objetivo de zerar sua taxa de utilização para que o enlace possa ser temporariamente desativado, economizando energia. Quando um enlace está completamente ocioso é possível desligar suas portas nos comutadores adjacentes. Se todas as portas de um comutador estiverem desligadas então o comutador também pode ser desligado por completo. Caso não exista caminho apto a receber algum dos fluxos em redirecionamento, o processo é abortado com o entendimento de que não é possível desocupar o enlace sem causar prejuízo aos indicadores de QoS dos tráfegos ativos.

O redirecionamento de fluxos é executado como um gatilho sempre que o monitoramento ativo da rede classifica algum

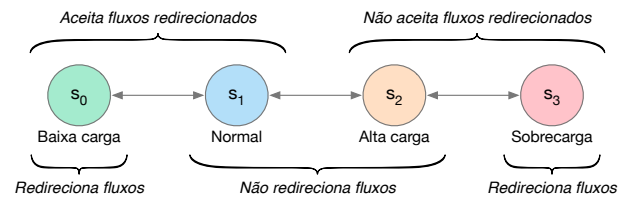


Fig. 1: Estados $S_t(l)$ assumidos por um enlace l .

enlace como baixa carga ou sobrecarga. A busca pelos caminhos alternativos é feita pelo algoritmo 1, com a restrição de $s_{max} = s_1$. Esta restrição obriga que novos caminhos utilizem apenas enlaces nos estados normal ou baixa carga, evitando que enlaces no estado s_2 (alta carga) recebam mais fluxos e sejam classificados como sobrecarregados na sequência, o que iria resultar em novos redirecionamentos e num efeito “ping-pong” entre os estados s_2 e s_3 .

V. AVALIAÇÃO E RESULTADOS

Avaliamos a economia de energia alcançada pelo MGTD e o comparamos com uma abordagem do estado da arte. Descrivemos nas seções a seguir o cenário de avaliação, a metodologia adotada e os resultados alcançados.

A. Cenário Considerado

O MGTD pode ser utilizado em topologias arbitrárias, desde que os nós tenham suporte SDN. Neste trabalho consideramos o cenário apresentado na figura 2. Ele representa a topologia típica da rede de um campus universitário [2]. São, ao todo, 95 enlaces e 45 nós. Dentre os nós, temos comutadores de encaminhamento (1 a 4) e comutadores de acesso (5 a 18). Os demais são nós hospedeiros (19 a 45), formando dois grupos distintos (parte superior e inferior da figura). Os pares de aplicações cliente/servidor para geração e consumo de tráfego estão sempre posicionadas um em cada grupo de hospedeiros.

Nas avaliações, consideramos cenários com enlaces homogêneos e heterogêneos. No cenário homogêneo, de forma equivalente a [2], todos os enlaces possuem a mesma velocidade negociada de 250 Mbps. No cenário heterogêneo, os enlaces interconectando os comutadores de encaminhamento possuem velocidade de 1 Gbps, enquanto os demais enlaces mantêm a velocidade de 250 Mbps.

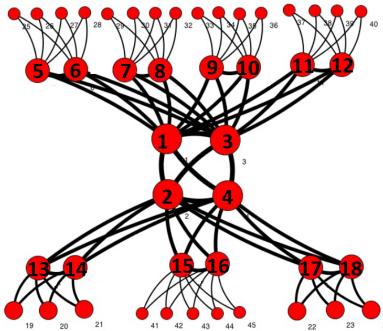


Fig. 2: Topologia da rede de um campus universitário.

B. Metodologia de avaliação

O MGTD foi avaliado por experimentos no emulador de redes *Mininet*¹ (ver. 2.3.0), ferramenta essa que permite a criação de uma rede virtual de hospedeiros, comutadores, enlaces e controladores em um único computador. O controlador OpenFlow utilizado foi o POX (ver. 0.2.0) e os comutadores foram executados em instâncias do *OpenVSwitch*² (ver. 2.5.2). Os experimentos foram realizadas em um computador com

¹<http://mininet.org>

²<https://www.openvswitch.org/>

processador Intel Core i5@3.1 GHz de 7ª geração, 8 GB de memória RAM e sistema operacional Ubuntu 16.04.

Os parâmetros de consumo energético (em Watts) são os mesmos de [5] e [12]. Especificamente: $E_{b_t}(c) = 146$, $E_{p_g}(c) = 0.87$ e $E_{p_f}(c) = 0.18$. A taxa de chegada de novos fluxos segue um processo de *Poisson* com uma expectativa de intervalo $\lambda = 3$ seg. Novos fluxos possuem duração fixa de 15 seg. e podem assumir um dos níveis de carga descritos na tabela II. Esses valores são equivalentes aos utilizados em [2] e consideram o comportamento de tráfego noturno, a média de tráfego diurno e o tráfego de pico anual. Conforme os autores, o tráfego de pico anual é cinco vezes maior que o tráfego noturno, e o tráfego médio diurno é três vezes maior que o tráfego noturno. O tráfego foi gerado pela ferramenta D-ITG (Gerador de Tráfego Internet) [13].

TABELA II: Níveis de carga conforme os padrões de tráfego.

Nível	Demanda tráfego	Descrição	Vazão do fluxo
N_1	Alta	Pico anual]150, 250] Mbps
N_2	Média	Média diurno]50, 150] Mbps
N_3	Baixa	Média noturno	[0, 50] Mbps

Os limiares $T_{0 \rightarrow 1}$, $T_{1 \rightarrow 2}$ e $T_{2 \rightarrow 3}$, usados na classificação dos estados dos enlaces, foram definidos em, respectivamente, 20%, 60% e 80%. Por se tratar de um ambiente emulado, foi definido $\Delta t = 1$ seg. Consideramos a construção dos conjuntos \mathcal{P}_{ij} com os $k = 16$ caminhos mínimos conectando o hospedeiro i ao j , $\forall i, j \in \mathcal{N}_h$. Todos os caminhos mínimos terão cinco saltos entre a origem e o destino nesta topologia.

As avaliações consideram a porcentagem de economia de energia em relação à configuração em que todos os nós estão ligados e a vazão média dos fluxos de dados. Cada experimento foi repetido 50 vezes, com duração individual de 120 seg. Por padrão, as médias e os intervalos de confiança apresentados correspondem ao nível de confiança de 95%.

C. Avaliações

Avaliamos o MGTD nos cenários homogêneo e heterogêneo, considerando os três níveis de demanda de tráfego. As figuras 3(a) e 3(b) apresentam as funções de distribuição acumulada para a economia de energia em cada configuração avaliada. Intuitivamente, quanto maior a demanda, menor a economia alcançada, já que mais enlaces estão em uso e há menos oportunidades para se desligar equipamentos ociosos.

Com baixa demanda de tráfego, os dois cenários apresentam comportamento similar. Nestes casos, o mínimo de enlaces necessários para manter a conectividade da rede está ativo. Por isso, o MGTD atinge uma expressiva economia de energia média superior a 46%. Em média demanda, a economia de energia se diferencia. Em um cenário heterogêneo, a economia média foi de aproximadamente 40%. Em um cenário homogêneo, esse valor é 3,75% menor. Como o cenário heterogêneo possui alguns enlaces com maior capacidade, então ele consegue absorver esta média demanda sem ligar novos enlaces. Já no cenário homogêneo é preciso ligar mais enlaces e, eventualmente mais comutadores, o que aumenta o consumo de energia. Isso fica ainda mais evidente com

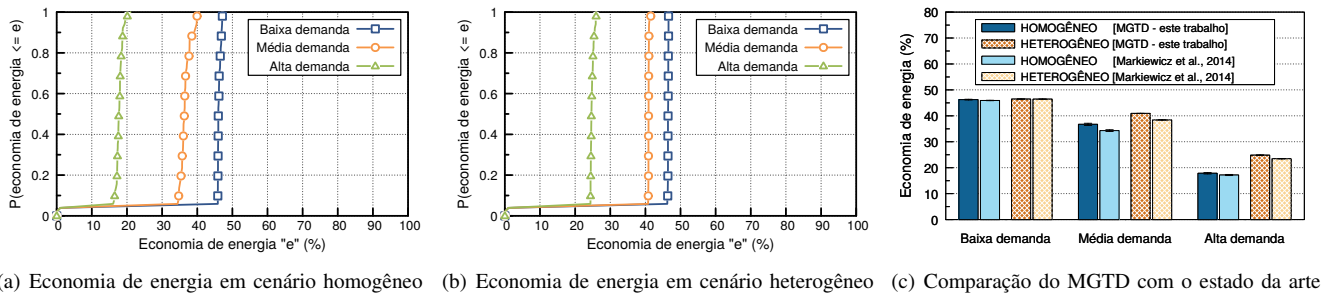


Fig. 3: Economia de energia alcançada nos cenários avaliados e comparação do MGTD com o estado da arte.

uma alta demanda de trabalho, onde um número maior de enlaces e comutadores é necessário para suportar o tráfego total, minimizando a oportunidade de se economizar energia. A economia de energia cai consideravelmente para ambos os casos, porém, é mais evidente no cenário homogêneo. Nessa situação, o cenário homogêneo consumiu 6,58% mais energia do que o heterogêneo, sendo que esse último não alcançou mais do que 25% de economia de energia média total.

A tabela III apresenta a vazão observada na rede durante as simulações. Cenários homogêneos e heterogêneos apresentam comportamento equivalente para baixa e média demandas. Para alta demanda, a vazão observada no cenário heterogêneo é superior. Isso ocorre pois os enlaces do núcleo do cenário homogêneo atingem seu limite de carga e se tornam um gargalo para a rede. Em oposição, os enlaces do núcleo do cenário heterogêneo comportam os fluxos gerados, proporcionando economia de energia e maior vazão entre os comutadores.

TABELA III: Intervalo de confiança para vazão da rede.

Nível	Demanda	Homogêneo	Heterogêneo
N_1	Alta	[115,2; 130,3] Mbps	[135,3; 152,7] Mbps
N_2	Média	[87,5; 98,8] Mbps	[89,9; 101,4] Mbps
N_3	Baixa	[23,2; 26,3] Mbps	[24,2; 27,3] Mbps

Por fim, comparamos o MGTD com uma solução do estado da arte, especificamente [2]. A figura 3(c) mostra que os dois algoritmos obtiveram o mesmo ganho em economia de energia nos cenários com baixa demanda de tráfego. De fato, o número mínimo de enlaces para conectividade da rede é capaz de suportar todos os fluxos dos experimentos de baixa demanda. Assim, nesses cenários, ambas soluções estão com seu nível máximo de economia. Em média demanda de tráfego, observamos uma diferença significativa entre os algoritmos, tanto para enlaces homogêneos quanto para enlaces heterogêneos. O MGTD foi, em média, 7% mais econômico que a proposta de [2] no cenário homogêneo, e 6,5%, em média, mais econômico no cenário heterogêneo. De fato, uma maior demanda de tráfego diminui a margem para se economizar energia, já que, independente da estratégia adotada, a maioria dos caminhos disponíveis terão que ser ligados para acomodar todos os fluxos. Quando a demanda de tráfego é alta, o MGTD foi 3,96% e 5,79% melhor que o método de [2], considerando os cenários homogêneo e heterogêneo, respectivamente. Em suma, mesmo quando as oportunidades de se economizar energia são raras, o MGTD tem vantagem. Os ganhos são expressivos, principalmente se consideramos o impacto final

dado a escala do uso de redes de computadores atualmente.

VI. CONCLUSÕES

Neste trabalho, apresentamos o MGTD: um Mecanismo de Gerenciamento de Tráfego Dinâmico para redes SDN ciente do consumo de energia. O mecanismo foi avaliado através de emulações no *Mininet* considerando uma topologia de rede realista. Os resultados mostraram que o MGTD promove uma economia de energia média de 46,01% em cenários com baixa demanda. Em cenários com média e alta demanda de tráfego, MGTD apresenta uma economia média de 36,72% e 17,86%, respectivamente. Quando comparamos o nosso mecanismo ao proposto por [2], o MGTD obtém uma vantagem média em termos de economia de energia de até 7% no cenário com média demanda de tráfego. Como trabalhos futuros vislumbramos a investigação em relação à escalabilidade do MGTD em outras topologias, considerando cenários com mais caminhos entre hospedeiros de origem e destino.

REFERÊNCIAS

- [1] L. C. Costa *et al.*, "Performance evaluation of OpenFlow data planes," in *IFIP/IEEE IM*, 2017, pp. 470–475.
- [2] A. Markiewicz *et al.*, "Energy consumption optimization for software defined networks considering dynamic traffic," in *IEEE CloudNet*, 2014, pp. 155–160.
- [3] P. Liu *et al.*, "A programmable energy efficient 40Gb/s switch using frequency scaling and OpenFlow," in *IEEE ICC Workshops*, 2017, pp. 405–410.
- [4] B. Addis *et al.*, "Energy management in communication networks: a journey through modeling and optimization glasses," *Computer Communications*, vol. 91–92, pp. 76–94, 2016.
- [5] S. Sasaki *et al.*, "A proposal of QoS-aware power saving scheme for SDN-based networks," in *NBiS*, 2015, pp. 405–410.
- [6] W. Fisher *et al.*, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *ACM SIGCOMM Workshops*, 2010, pp. 29–34.
- [7] G. Lin *et al.*, "Efficient heuristics for energy-aware routing in networks bundled links," *Computer Networks*, vol. 57, no. 8, pp. 1774–1788, 2013.
- [8] B. Heller *et al.*, "Elastictree: Saving energy in data center networks," in *USENIX NSDI*, 2010, pp. 1–17.
- [9] A. Fernández-Fernández *et al.*, "Improved energy-aware routing algorithm in software-defined networks," in *IEEE LCN*, 2016, pp. 196–199.
- [10] K. M. Habibullah *et al.*, "Reducing energy consumption of network infrastructure using spectral approach," in *Technology for Smart Futures*, M. Dastbaz *et al.*, Eds. Springer, Cham, 2018, pp. 235–250.
- [11] A. P. Bianzino *et al.*, "A survey of green networking research," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 3 – 20, 2010.
- [12] P. Mahadevan *et al.*, "Energy aware network operations," in *IEEE INFOCOM*, 2009, pp. 25–30.
- [13] A. Botta *et al.*, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.