Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6629852, 10 pages
https://doi.org/10.1155/2021/6629852

Research Article

DDPG-Based Energy-Efficient Flow Scheduling Algorithm in Software-Defined Data Centers

Zan Yao, Ying Wang, Luoming Meng, Xuesong Qiu, and Peng Yu

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

Correspondence should be addressed to Ying Wang; wangy@bupt.edu.cn

Received 1 March 2021; Accepted 11 June 2021; Published 28 June 2021

Academic Editor: Yan Huang

With the rapid development of data centers, the energy consumption brought by more and more data centers cannot be underestimated. How to intelligently manage software-defined data center networks to reduce network energy consumption and improve network performance is becoming an important research subject. In this paper, for the flows with deadline requirements, we study how to design the rate-variable flow scheduling scheme to realize energy-saving and minimize the mean completion time (MCT) of flows based on meeting the deadline requirement. The flow scheduling optimization problem can be modeled as a Markov decision process (MDP). To cope with a large solution space, we design a DDPG-EEFS algorithm to find the optimal scheduling scheme for flows. The simulation result reveals that the DDPG-EEFS algorithm only trains part of the states and gets a good energy-saving effect and network performance. When the traffic intensity is small, the transmission time performance can be improved by sacrificing a little energy efficiency.

1. Introduction

With the development of 5G technology [1–4], more and more data centers as important carriers of data storage and processing will be established [5–7]. Worldwide data center energy consumption has reached about 8% of global energy consumption. It mainly comes from three aspects: network, server, and refrigeration system. The network energy consumption accounts for about 20% of the total data center energy consumption [8]. With the introduction of DVFS (Dynamic Voltage and Frequency Scaling) and hardware virtualization technology [9, 10], the energy consumption efficiency of servers has been greatly improved, and more and more new heat dissipation technologies have emerged [11]. The proportion of network energy consumption in data center energy consumption will continue to increase, and it cannot be ignored.

The network energy-saving technology can be divided into three types: topology design and transformation [12], device sleeping (DS) [13–18], and adaptive link rate (ALR) [19–21]. The energy-saving topology design and transformation mainly studies how to maximize the ratio of the forwarding capacity provided by the topology to the total energy consumption of the network under constraint conditions, to improve the utilization rate of topology energy consumption. As for DS technology, because in the data center network with SDN technology, the "rich connection" topology has more alternative routes. Under the constraints of network connectivity and network performance, the network flows are aggregated and transmitted in the subset T of the network topology G, and the devices and links in T/G are dormant, to achieve energy saving. The ALR-based energy cost model shows that the link energy consumption is exponentially related to the actual transmission rate on the link. When the energy-saving topology is determined, the ALR-based energy cost model is used to guide further improving the energy utilization efficiency of the network link, which is through adjusting the bandwidth usage of different links at different times. There are few studies on the third one. In this paper, we will study it.

The representative ALR-based energy-saving scheme is a preemptive Most-Critical-First scheduling algorithm for the flow proposed in Ref. [20]. It selects the interval with the largest energy consumption density as the critical interval, and all

the flows in this critical interval will be preferentially scheduled. The transmitting rate of every flow is constant. If the rate is time variable, then the energy saving can be more efficient. Hence, we propose a rate-variable routing scheme based on bandwidth sharing mechanism in Ref. [21]. It sorts the flows according to EDF (earlier deadline first) policy and then calculates the routes in turn. However, the scheduling order based on EDF is too simple and the energy efficiency still has the space to be improved.

In addition to the optimization goal of energy saving, network operators must ensure and optimize the QoS (quality of service). There are some representative flows on the data center network, such as search and social networks that generate many requests and responses, which need to go through the data center to perform the tasks requested by the user. The performance that the user is concerned about is the response speed of the requests, and a tolerable deadline is generally given [22]. If the mean completion time (MCT) of the flows can be decreased based on ensuring the deadline and ensuring the network energy efficiency, then the QoS will be greatly increased [23]. Above all, this paper will assume the route has been determined and try to design a rate variable flow scheduling mechanism to minimize network energy consumption and improve the QoS performance.

Most of the current works model the traffic engineering problem as a mixed-integer linear programming problem and propose heuristic algorithms. The dual optimal problem has a large solution space, and the above heuristic algorithms are close to traversal, and the scalability is limited. Researchers began to study DRL-based flexible traffic control mechanisms to improve the performance of the data center network. The DRL-based traffic engineering algorithms [24–30] are mostly about finding the best QoS route, and they are driven by experience to deal with the overly complex and dynamic network environment. At present, there is no DRL-based flow scheduling scheme with a variable rate as far as we know.

The proposed energy-saving scheduling optimization problem can be modeled as a Markov decision process (MDP) with a state space, action space, and reward function. Although the RL algorithm can learn from the surrounding environment itself, it still needs to design the corresponding features manually for it to be able to converge. In practical applications, the number of states may be large, and in many cases, the features are difficult to be designed manually. The neural network happens to have particularly good processing for massive data. And the flow rate is designed to be constant in the current literatures as we know. We suppose the flow transmitting rate is more flexible and variable, so it is a continuous control problem to design a variable rate energy-saving flow scheduling scheme. The deep deterministic policy gradient (DDPG) algorithm is one based on the actor-critic (AC) framework proposed by Lillicap et al. [31], which is based on the DQN and the deterministic policy gradient (DPG) method, and it is an effective method to solve the continuous control problem. Hence, we adopt the DDPG method to solve it.

In summary, this paper focuses on the energy-saving flow scheduling problem based on the ALR model and applies the DDPG algorithm to solve it. Our main contributions are two folds as follows:

(1) When the network topology and the routes of flows are determined, to further reduce the energy consumption and improve the QoS requirement, based on the ALR energy cost model, the energy-saving QoS flow scheduling problem and the dual optimization objective of minimum energy consumption and mean completion time of flows are proposed. The dual optimal problem is a continual control problem, and it has a large solution space, which can be modeled as a Markov decision process

(2) Based on the advantages of DDPG in solving continuous control problems, and the problem of scalability, the DDPG Energy-Efficient Flow Scheduling (DDPG-EEFS) Algorithm is proposed to obtain the optimal scheduling scheme

(3) Based on the ALR energy cost model, a rate variable flow transmission mechanism is proposed to flexibly scheduling flow and to balance the flow transmission on the link in time and space and improve the energy-saving effectively

The rest of the paper is organized as follows: Section 2 analyzes related works. Then, the dual-objective optimization problem is presented in Section 3. We propose the DDPG-EEFS algorithm to solve the problem in Section 4. The simulation results are presented to verify the feasibility and effectiveness of the proposed approach in Section 5. Finally, the conclusion is given in Section 6.

2. Related Work

In this part, we review the past studies from two aspects: data center network energy-saving technology and DRL-based network traffic control algorithm.

2.1. Data Center Network Energy-Saving Technology. In terms of data center network energy-saving technologies, there is a lot of research, which can be divided into energy-saving topology design [12], and energy-saving routing and flow scheduling schemes [13–21], which are mainly based on the DS energy cost model or the ALR energy cost model.

The first technology is designed from the perspective of topology design and transformation to save energy. Ref. [12] defined the influence parameters of links on topological connectivity and the threshold of network connectivity decline percentage. Under the constraint of the threshold, the links are deleted from the network topology and the topology is updated according to the increasing order of the influence parameters, to achieve the goal of network energy saving. Energy-saving topology is suitable to be applied in the initial stage of network construction.

The DS-based energy-saving schemes are mainly from the view of forwarding and routing. Ref. [13] proposed a method to construct an elastic tree, which can dynamically adjust the set of active nodes and links. While reducing

energy consumption, it can deal with burst traffic and has good fault-tolerant performance. Ref. [14], Ref. [15], and Ref. [16] all proposed heuristic energy-saving routing algorithm to minimize the number of active links, under constraints. Ref. [17] proposed an online switching mechanism of multiple topologies in the data plane, which can sleep some devices and ports to achieve energy saving while meeting the dynamic demand of traffic. Li et al. [18] explored a new energy-aware flow preempting scheduling method in the time dimension and used the policy of EXR (exclusive routing), i.e., each flow preempted the route according to the priority and occupied its own route. This kind of energy-saving routing and scheduling method based on the DS model refers to the state transition of switches and ports. So, this kind of method will take some time and the DS model is not suitable for real-time flow scheduling.

A few literatures are targeted to the ALR-based energy efficiency flow scheduling. In Ref. [20], the flows in critical interval with the largest energy consumption density will be scheduling first. The energy consumption efficiency of the network is further improved by balancing the flow in the link space and time. And the flow scheduling problem mainly involves the transmission rate and transmission time. The flow bandwidth allocation is fixed, i.e., the flow rate is constant in the current flow scheduling literature as we know. In this paper, we will provide a rate variable flow scheduling mechanism to minimize network energy consumption and improve MCT.

2.2. DRL Algorithm for Network Traffic Control. The solution space of the above algorithms in Section 2.1 is large, so the scalability of the above algorithms is limited. With the development of DRL, the recent trend in the field of network technology is to use AI algorithm to control and operate the network traffic. Ref. [24] proposed an adaptive multimedia flow control method based on DDPG to optimize QoE performances. In a complex and dynamic network environment, each multimedia flow is allocated appropriate bandwidth in each path based on experience rather than a mathematical model. Ref. [25] involves scheduling the transmission times and bandwidths of multiple flows. The state is defined as resource allocation, and the action is the route selection. The contribution ratio of multiple resources in reducing delay is quantified, so that the performance requirements of flows are transformed into resource requirements of flows. Then, the DRL agent interacts with the network continuously and obtains a feasible path adaptively according to the network state, which means allocating the optimal network resources for the flow, to improve the network throughput, the completion time of the flow, and the load balance of the link. Most of the above literatures focus on nondata center network, and the optimization objective is network throughput, delay, and other performance requirements. And there is few DRL-based energy-saving flow control study in software-defined DCN.

To optimize the energy-saving effectiveness in DCN, based on the DS energy consumption model, we put forward a DQN-based routing algorithm in Ref. [29]. To further save energy, based on the ALR network energy consumption model, we will use the DDPG algorithm to design a variable flow rate scheduling mechanism to achieve saving energy and improve the QoS.

3. Model of Network System

3.1. Motivation. The data center network is an undirected graph and can be modeled as $G(V, E)$, where V is the set of switches, E is the set of links. The set of traffic that needs to be transmitted is defined as $J = \{j_1, j_2, .j_i, \cdots, j_n\}$, where each flow is defined as $j_i = [p_i, q_i, r_i, d_i, w_i]$, $j_i \in J$. p_i and q_i indicate the source and destination nodes separately; r_i and d_i represent the start time and the deadline of the flow, respectively; and w_i means the data size that needs to be transmitted. Taking Figure 1 as an instance, we assume that one simple network which consists of five switches and six links and currently three flows come, where $j_1 = [1, 3, 0, 4, 8]$, $j_2 = [2, 5, 0, 3, 9]$, and $j_3 = [3, 5, 0, 2, 4]$. We assume each flow uses a single path transmission, and the routing has been determined in advance; thus, active switches and links are also known ahead of time. Let P_i be the sequence of links through which j_i is routed, which is calculated by the shortest path principle. Here, we set $P_1 = [l_1, l_3]$, $P_2 = [l_4]$, and $P_3 = [l_3, l_4]$ .Therefore, the number set of the actual active switches is $\{1, 2, 3, 5\}$, and the number set of the actual active links is $\{1, 3, 4\}$, and both of which are marked with green in Figure 1.

The main problem in flow scheduling is to provide an appropriate flow scheduling scheme to minimize the energy consumption and minimum mean completion time of flows, while guaranteeing the transmission performance of the flow deadline. The scheduling scheme mainly includes flow transmission interval and transmission rate, which have a great impact on the link energy consumption and MCT. Flows can share the same link for transmission, or they can be transmitted by exclusive link mode. The flow can be transmitted at a constant speed or at a variable speed. So, there will be a large number of possible transmission combinations. Different transmission schemes will produce different energy consumption and MCT.

Here, we adopt the ALR power consumption model to calculate the power energy. The power consumption function [19] $f(x_e)$ is given by Formula (1) to uniformly characterize the manner in which the energy is being consumed with respect to the transmission rate x_e of each link $e \in \varepsilon$.

$$f(x_e) = \sigma + \mu x_e^{\alpha}, \tag{1}$$

where σ, μ, and α are constants associated with the link type. Constant σ represents the idle power energy for maintaining link state, and $\alpha > 1$ that means $f(.)$ is superadditive. Here, we define the parameter $\alpha = 2$. Constant C is the maximum link transmission rate.

For the three flows above, the time range is set from the current time to the next five units of time, and each link has five units of bandwidth. We give three representative transmission schemes as shown in Figure 2. The first scheme adopts the transmission strategy of uniform speed and shared bandwidth, and the deadline is guaranteed, and the energy consumption is $15\sigma + 115\mu$, and MCT is 3 units of

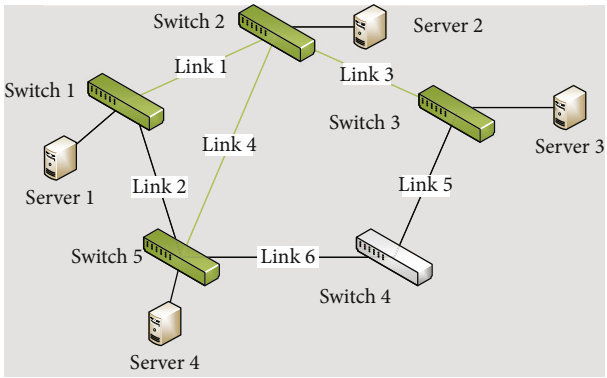

Figure 1: The simple network example.

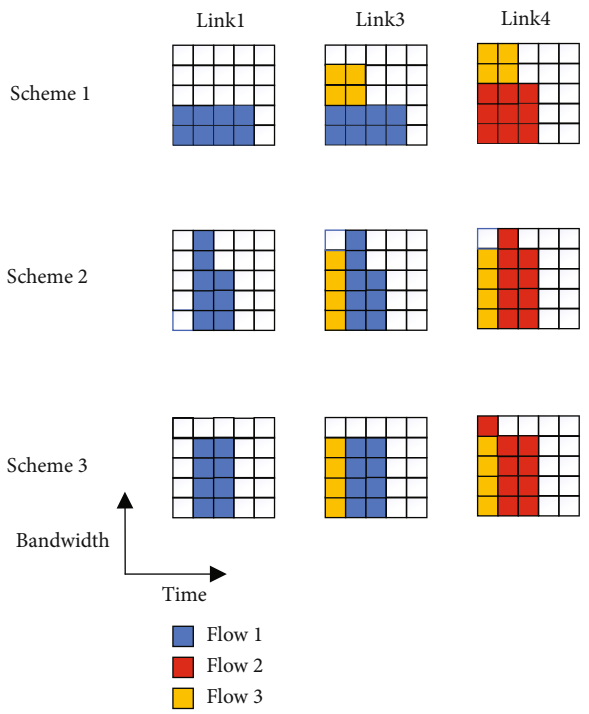

Figure 2: Different flow transmission schemes.

time. In the second scheme, the flow with earlier deadline is transmitted first, and the transmission strategy with the flow occupying the link bandwidth exclusively is adopted. The energy consumption is $15\sigma + 141\mu$, and MCT is 2.33 units of time. Scheme 3 is improved based on scheme 2, that is, the flow is transmitted as evenly as possible in time without changing the transmission interval. And its energy consumption is $15\sigma + 137\mu$, and MCT is 2.33 units of time. The average completion time of scheme 1 is the largest, and its energy consumption is the smallest. The average completion time of scheme 2 and scheme 3 is the smallest one, and the energy consumption of scheme 3 is smaller, because it balances the flow transmission in its transmission interval as much as possible. By adjusting the transmission rate and transmission interval, the energy consumption and the average completion time of the stream can be effectively reduced.

The goal of this paper is to find a flow transmission scheme to minimize the energy consumption and the MCT.

3.2. Problem Formulation. In this paper, the main optimization objective is to minimize the weighted sum of network link energy consumption and mean complete time (MCT) of flows, which is expressed by Formula (2), where the variables ϕ' and MCT$'$ are the normalized ones, which are, respectively, calculated by Formulas (3) and (4). The two constants ρ and $(1-\rho)$ represent the ratio between the energy consumption and MCT, respectively, and s represents one feasible solution of flow scheduling and will be included in the solution set s. Formulas (5)–(7) calculate the energy consumption ϕ and MCT, respectively, where variables r_i' and d_i' represent the start time and ending time of the actual transmission of the flow $j_i \in J$. And constraints are expressed by Formulas (8)–(10):

$$s^* = \arg \min_S \left(\rho\phi' + (1-\rho)\mathrm{MCT}'\right), \tag{2}$$

$$\phi' = \frac{\phi - \min_{1\le j\le n}\{\phi\}}{\max_{1\le j\le n}\{\phi\} - \min_{1\le j\le n}\{\phi\}}, \tag{3}$$

$$\mathrm{MCT}'_{s_i} = \frac{\mathrm{MCT}s_i - \min_{1\le j\le n}\{\mathrm{MCT}s_j\}}{\max_{1\le j\le n}\{\mathrm{MCT}s_j\} - \min_{1\le j\le n}\{\mathrm{MCT}s_j\}}, \tag{4}$$

$$\phi = \int_{t_0}^{t_{\mathrm{end}}} \sum_{e\in\varepsilon_a} (\sigma + \mu(x_e(t))^\alpha)dt, \tag{5}$$

$$x_e(t) = \sum_{e\in P_i} s_i(t), \tag{6}$$

$$\mathrm{MCT} = \frac{\sum_{i=1}^{n}\left(d_i' - r_i'\right)}{n}. \tag{7}$$

With constraints

$$\int_{r_i'}^{d_i'} s_i(t)dt = b_i, \tag{8}$$

$$0 \le x_e(t) \le \beta C, \tag{9}$$

$$\sum_{v\in N(u)} (f_i^{uv} - f_i^{vu}) = \begin{cases} b_i, & \text{if } u = p_i \\ -b_i, & \text{if } u = q_i \\ 0, & \text{else} \end{cases}. \tag{10}$$

Formulas (8)–(10) represent performance constraints. Formula (8) represents each flow must be completed before its latest deadline; Formula (9) represents link resource capacity constraints, i.e., the bandwidth used by network traffic cannot exceed the available bandwidth of the link. To ensure the availability of the link, the available bandwidth of the link is β times the link bandwidth capacity, and $(1-\beta)$ times the link bandwidth needs to be reserved for the emergency. Formula (10) represents the flow conservation

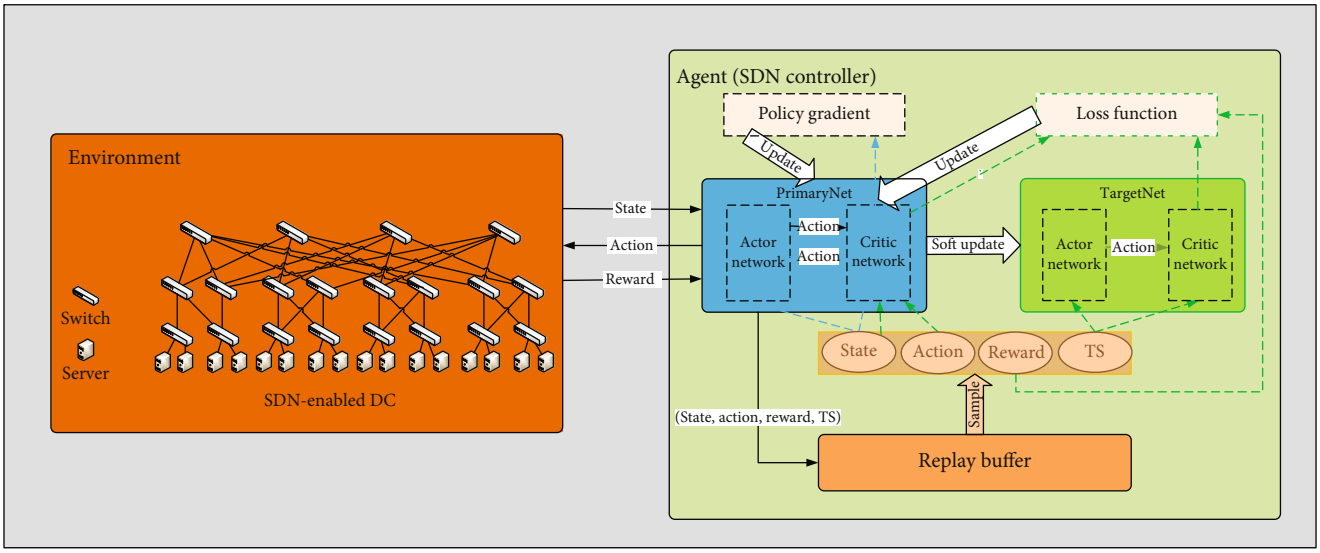

Figure 3: An overview of the DDPG-EEFS algorithm.

limit. The flow from the source switch is equal to the flow into the destination switch, and the flows from all intermediate switches are equal to the flows out. $N(u)$ represents the neighbor switch set of the switch u, and f_i^{uv} represents the flow bandwidth deployed on the link. In the case of flow transmission over a simple path, this value is b_i or 0.

4. DDPG-Based Energy-Efficient Flow Scheduling Algorithm

For the energy-saving scheduling optimization model established above, we model it as a Markov decision process (MDP) with a state space, action space, and reward function. This paper tries to apply the DDPG method to seek the most energy-saving transmission rate for each flow while minimizing the MCT of flows. Firstly, we propose a DDPG-EEFS architecture and describe the components. State, action, and reward are also outlined. Secondly, the process of the DDPG-EEFS algorithm is presented.

4.1. DDPG-EEFS Architecture. The DDPG-EEFS architecture is shown in Figure 3. It mainly includes environment and agent.

4.1.1. Environment. The environment is an SDN-enabled data center network, which consists of switches, links, and servers.

4.1.2. Agent. An agent is used to communicate with the environment. Once it observes the consequences, it learns to change its behavior and action in response to the reward. When DDPG is applied in the system, the SDN controller has a global view to get the state of the network environment, and it can be seen as an agent to make decisions based on observation and carry out a series of actions and take actions to the current state and provide flexible policy deployment.

DDPG combines the DQN method and the DPG method with actor critical framework and uses a neural network to fit policy function and Q function to form a more efficient and stable control model. To improve the convergence and stability of the network, the important idea of experience replay and target network are used in the DDPG algorithm. The purpose of the former is to disrupt the correlation between the data, so that the sequence meets the independence and identical distribution. The latter regularly copies the online network parameters to the target network with the same structure and then uses the target network to update online network parameters.

(1) Primary Network. The primary network is used to determine an action based on the current state with the corresponding critic value. So, its input is the current state, and its output is an action. The primary network consists of an actor network and critic network. The actor network is the online policy network and is responsible for selecting the current action a_i according to the current state s_i and used to interact with the environment to generate the next state s_{i+1} and reward r_i. The critic network is used to approximate the value function $Q(s, a|\theta^Q)$ of the state action pair (i.e., the output of actor network) and to provide gradient information and helps the actor to learn the gradient of the policy.

(2) Target Network. The target network is the same as the primary network model and consists of a target actor network and target critic network. The target actor network is responsible for selecting the optimal next action a_{i+1} according to the next state s_{i+1} sampled in the replay buffer, and its input is the transformed state (TS) (s_i, s_{i+1}), and the network parameter $\theta^{\mu'}$ is periodically copied from θ^{μ}. The target critic network is responsible for calculating the value function $Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$, and the network parameter $\theta^{Q'}$ is copied from θ^Q periodically.

(3) Replay Buffer. The concept of experience replay is used to extract training samples during neural network training. The

observed state transition process is first stored in a replay buffer. After the samples in the replay buffer have accumulated to a certain extent, they will be randomly chosen to update the network. The main reason is that the samples obtained by randomly exploring the surrounding environment by different flows are a sequence associated with time. Due to the temporal correlation, if the data is directly used as a sample for training, the system convergence will be greatly affected, thereby the random sampling method solves the time correlation problem. This random extraction approach disrupts the correlation between experiences and makes neural network updates more efficient. In summary, the replay buffer is an especially important part of the DRL method, which greatly improves the system performance of DRL.

(4) State. The state in DDPG should reflect the situation of the environment. For the problem of flow with hard-deadline flow scheduling, the state of the environment mainly refers to the state of links, so we set $s = [x_{e_1}(t),\cdots,x_{e_i}(t),\cdots,x_{e_k}(t)]$, which is the sum of the transmission rates of the flows on each active link in the DCN, where k is the number of active links.

(5) Action. The agent focuses on mapping the space of state to the space of action and in identifying the optimal policy. The energy-saving flow scheduling is a continuous problem; its action space includes the bandwidth allocation of each flow in the available transmission interval, which is a continuous variable. DDPG is a preponderant method to solve it.

The available transmission time of each flow is divided into $m_i = w_i/T$ time periods, where T is the minimum scheduling time unit. Action a refers to the transmission rate allocation of each flow in different time periods $v_i = [v_{i,1},\cdots,v_{i,j},\cdots,v_{i,m_i}]$, $1 \le j \le m_i$ and $\sum_{j=1}^{m_i} v_{i,j} = 1$. According to the flow transmission rate allocation, the real transmission rates of flows in the transmission interval can be calculated. By adjusting the transmission rate allocation situation of flows during different time periods, different flow scheduling schemes are obtained. Action a is defined as follows:

$$a = \left[\left[v_{1,1},\cdots,v_{1,j},\cdots,v_{1,m_1}\right],\cdots,\left[v_{i,1},\cdots,v_{i,j},\cdots,v_{i,m_i}\right],\cdots,\left[v_{l,1},\cdots,v_{l,j},\cdots,v_{l,m_l}\right]\right]. \tag{11}$$

(6) Reward. The reward r of agent is related to the adopted network operation and maintenance strategy. These control policies can be adjusted by changing the reward settings. The immediate reward is defined by analyzing the objective function. Since the objective function is to find the minimum energy consumption and mean completion time of flows, and the smaller the weighted sum of both items, the larger the reward, so the reciprocal of weighted sum can be immediately used as an immediate reward in Formula (12)

$$r = \begin{cases} \dfrac{1}{\rho\phi' + (1-\rho)\mathrm{MCT}'}, & \text{if meet the constraints (8) – (10),} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

For those states that do not satisfy the bandwidth constraints (8)–(10), the immediate reward is 0.

4.2. DDPG-EEFS Algorithm Process. The general process of DDPG optimizing flow scheduling is as follows: Firstly, the agent obtains an accurate network state. Secondly, the agent determines an action; then, the SDN controller produces rules and distributes them to the switches in the data plane. Finally, the agent obtains the rewards and new network state after the implementation of the new scheduling scheme. The training goal of the DDPG agent is to find the optimal action a according to the input state s to maximize the reward r. The process is shown with black solid lines in Figure 3.

The general DDPG framework cannot clearly define how to explore, and the TE-aware random search method [23] uses the basic TE solution a_{base} as the baseline to guide the exploration process. So here, we use the TE-aware random search method to improve the efficiency of exploration. The DRL agent generates action $a_{\text{base}} + \varepsilon \cdot \mathbb{N}$ with probability ε, and action $a + \varepsilon \cdot \mathbb{N}$ with probability $(1 - \varepsilon)$. a is the output of the actor network. The parameter $\mathbb{N}$ is uniformly distributed random noises. a_{base} can be obtained by different methods. Every flow is supposed to transfer uniformly in the interval between arrival time and its deadline. Although it is not the best solution, it is enough as a fundamental solution. We decide to increase the probability of a random noise to the basic TE solution and the action of actor network instead of directly adopting a_{base} and a. Moreover, ε decreases with the increase of epoch because the more learning, the more action output will be adopted.

Deep neural networks are introduced in these networks, whose parameters are updated according to learning. To make efficient use of hardware optimizations, DDPG explores policy with an off-policy algorithm in minibatches, rather than online. For the flow scheduling problem, it can be considered as a continual control environment.

At each time step, the actor primary network and critic primary network are updated by sampling a minibatch from the replay buffer. Sampling is made up of a series of transitions, which contains the state, the action, the reward, and TS. Their updating is introduced as follows.

The reverse transmission of the policy gradient of the DPG neural network for actor module is shown in Equation (13), the related data flow is also shown in Figure 3 with blue dotted lines.

$$\begin{aligned} \nabla_{\theta^\mu} J &= \text{grad}\,[Q] * \text{grad}\,[\mu] \\ &\approx \frac{1}{N}\sum_i \nabla_a Q\left(s, a\middle|\theta^Q\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu\left(s|\theta^\mu\right)\Big|_{s_i}, \end{aligned} \tag{13}$$

where J is the performance function to measure the performance of the policy μ. The former in the right of equal sign is the action gradient from the critical network, which is used to characterize the direction of movement on which the action gets a higher return; the latter is the parameter gradient for the actor network, which is used to characterize how should the neural network of actor adjust its own parameters in order to make the neural network to select the action with the highest return with higher possibility. The combination of the two items means that the neural network of the actor module moves towards the direction with a higher possibility of getting a higher return to modify its own parameters.

The update process of the DQN network in a critical module is shown in Equation (14), which is also shown in Figure 3 with green dotted lines. TD_Error (TD: temporal difference) L is equal to the mean square error of Q value of target network and the Q value of the online network, where the Q value of target network is shown in Equation (15), and it is based on the Q value of next state s_{i+1} and next action a_{i+1} . The next action here comes from the target network of the actor module.

$$L = \frac{1}{N}\sum_{i}\left(y_i - Q\left(s_i, a_i \middle| \theta^Q\right)\right)^2, \tag{14}$$

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}\middle|\theta^{\mu'}\right)\middle|\theta^{Q'}\right). \tag{15}$$

Therefore, for the network energy-efficient flow scheduling problem, the DDPG-EEFS algorithm only trains some state data to get optimal results.

The simple uniform sampling method ignores the significance of the samples in the pool. Priority experience replay assigns a priority to each sample, and the DRL agent chooses one sample based on the priority to learn experience from transitions more effectively. Priority experience replay is applied to the TE problem [26]. DDPG includes actor and critical networks, so the priority also includes the following two parts. The first part is corresponding to the training of the critical network. The second part is related to the actor network training. Combining the two parts, the priority of samples is shown in Formula (16). When $|\bar{\nabla_a Q}|$ is the mean value of the absolute value of $\nabla_a Q$, parameter φ determines the relative importance between and $\nabla_a Q$. ς is a small constant to avoid the error of edge cases of transition, i.e., if the error is 0, it will need to be revisited. The probability of transition sampling is calculated by Formula (17), where the exponent β_0 describes the priority. When $\beta_0 = 0$, the transition sampling becomes uniform sampling.

$$p_i = \varphi \cdot (L + \varsigma) + (1 - \varphi) \cdot |\bar{\nabla_a Q}|, \tag{16}$$

$$P(i) = \frac{p_i^{\beta_0}}{\sum_j^{|B|} p_j^{\beta_0}}. \tag{17}$$

5. Simulation and Results

To verify the effectiveness of the proposed DDPG-EEFS algorithm, simulation is conducted in the SDN-enabled data center network with Fat-Tree topology.

5.1. Simulation Environment and Setting. Under the Windows 10 system, the Python language is used to program the algorithm. The hardware platform is configured as a 2.4 GHz CPU and 64 GB memory. This work selects the commonly used Fat-Tree data center network topology, which is set to consist of 20 four-point switches, 16 hosts, and 48 links.

We mainly use the network energy-saving percentage P as the evaluation matrix of energy-saving effectiveness, that is, the network energy consumption saved by using the method A accounts for the percentage of the total network energy consumption $\mathrm{NEC}_{\mathrm{full}}$when all the links are full load. The specific definition is as shown in Formula (18):

$$P = 1 - \frac{\mathrm{NEC}_{\mathrm{A}}}{\mathrm{NEC}_{\mathrm{full}}}. \tag{18}$$

5.2. Simulation Results and Analysis. To verify the validity and performance of the proposed DDPG-EEFS algorithm, we mainly design the simulation from two parts.

Firstly, to verify the effectiveness of the algorithm, we design 64 flows, which belong to 4 different pods, and there are half of flows that go through the core switch. To deal with emergencies and failure recovery, the parameter of redundancy δ is taken as 0.8 in our experiment. Our goal is to find the optimal scheduling scheme to make the objective function minimized. Through learning and constantly adjusting various parameters, we finally obtain the actual parameters in the stable convergence algorithm. The parameters in the algorithm are given in Table 1.

After the training of the DDPG-EEFS algorithm is completed, the model is saved and then tested, and the network will find a relatively ideal scheduling scheme. For the test results, the normalized optimization objective is counted every 100 steps, which are shown in Figure 4. It is found that after reaching 800 steps, the algorithm approaches convergence and the objective function tends to be stable.

Secondly, because energy-saving and performance optimization sometimes restrict each other, pure DDPG-EEFS, i.e., DDPG-EEFS without considering the synchronous optimization of network flow completion time performance, is chosen to compare with the base solution a_{base} proposed in Section 4, i.e., every flow is supposed to transfer uniformly in the interval between arrival time and its deadline, and the Most-Critical-First [4] algorithm to evaluate the energy-saving effect and network performance. As shown in Figure 5, we mainly use the network energy-saving percentage P as an evaluation index of the energy-saving effect. We use the energy cost when the network link load is full as the benchmark of comparison in the network energy-saving percentage. Figure 5 shows the network energy-saving percentage of different algorithms under different network loads. The difference between the energy-saving effects of pure DDPG-EEFS and the Most-Critical-First scheduling

Table 1: The parameters of the DDPG-EEFS algorithm.

Parameters	Values
Parameter in the objective function ρ, $(1-\rho)$	0.9, 0.1
Learning rate λ for actor and critic network	0.002, 0.01
Discount factor γ	0.98
Exponent β_0	0.6
Constant ς	0.01
Parameter φ	0.6
Batch size N	64
Buffer size D	10000
Training round e	1200

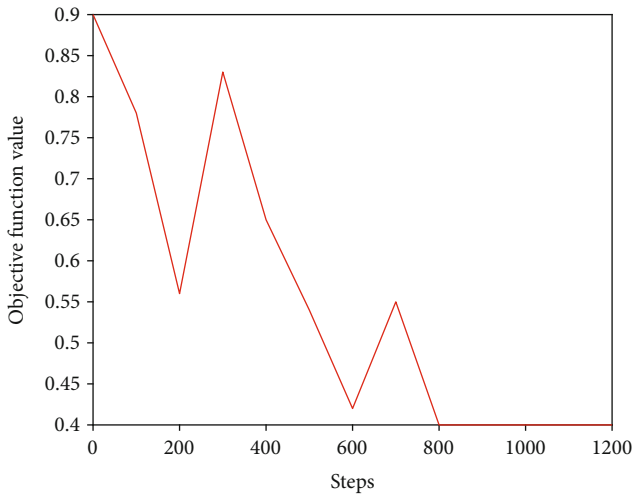

Figure 4: The process of searching the optimal scheduling scheme.

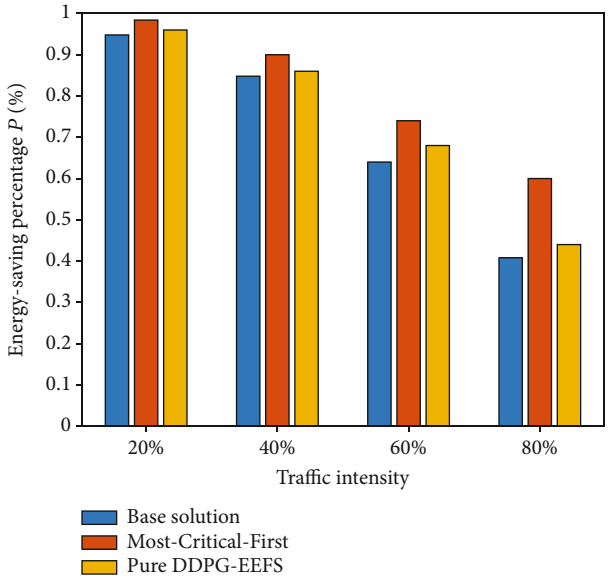

Figure 5: Energy-saving percentage at different traffic intensities.

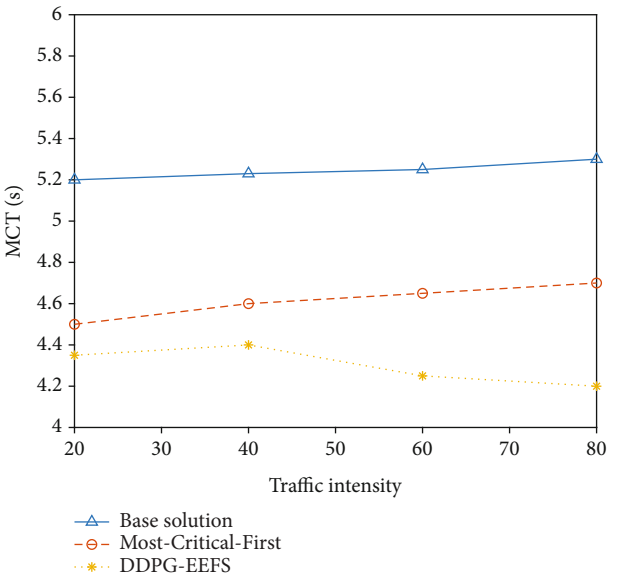

Figure 6: MCT of the three algorithms at different traffic intensities.

mechanism is about 2.4%. The static scheduling mechanism knows the information of all flows in advance, and it can carry out global optimization. Compared with the base solution a_{base}, the network energy-saving percentage of the pure DDPG-EEFS mechanism is increased by 3.2%.

As shown in Figure 6, we can see that under the different load conditions, the MCT by using DDPG-EEFS is lower than that of the baseline solution and Most-Critical-First algorithm. Compared with the base solution, the MCT of flows of DDPG-EEFS is reduced by about 15%.

6. Conclusions and Future Work

SDN is proposed as the promising technology in data center networks, and it can provide central network management and global traffic control. In this paper, we propose the dual optimization goals of the energy-saving and MCT of flows and design the DDPG-EEFS algorithm to solve it. For network operators, the shorter the average completion time is, the better the performance of the data plane is. Compared to other heuristic algorithms, DDPG-EEFS is easy to design the variable transmission rate and achieve good effect of energy saving, and good QoS.

Data Availability

This work selects the commonly used Fat-Tree data center network topology, which is set to consist of 20 four-point switches, 16 hosts, and 48 links. We design 64 flows, which belong to 4 different pods, and there are half of flows that go through the core switch.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (2018YFE0205502).

References

[1] C. Zhang, Y.-L. Ueng, C. Studer, and A. Burg, "Artificial intelligence for 5G and beyond 5G: implementations, algorithms, and optimizations," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 2, pp. 149–163, 2020.

[2] L. Chettri and R. Bera, "A comprehensive survey on internet of things (IoT) toward 5G wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[3] X. Zheng, Z. Cai, J. Li, and H. Gao, "A study on application-aware scheduling in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1787–1801, 2017.

[4] T. Zhu, T. Shi, J. Li, Z. Cai, and X. Zhou, "Task scheduling in deadline-aware mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4854–4866, 2019.

[5] Z. Cai and Q. Chen, "Latency-and-coverage aware data aggregation scheduling for multihop battery-free wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1770–1784, 2021.

[6] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang, and Y. Pan, "Stochastic load balancing for virtual resource management in datacenters," *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 459–472, 2020.

[7] A. Tzanakaki, M. P. Anastasopoulos, and D. Simeonidou, "Converged optical, wireless, and data center network infrastructures for 5G services," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 2, pp. A111–A122, 2019.

[8] W. Yaqi, F. Baochuan, W. Zhengtian, and G. Shuang, "A review on energy-efficient technology in large data center," in *2018 Chinese Control And Decision Conference (CCDC)*, pp. 5109–5114, Shenyang, 2018.

[9] A. Taha, S. Gosali, Z. Gong, N. Sollenberger, and J. Wright, "Method and system for dynamic voltage and frequency scaling (DVFS)," US Patent 12/190,029, 2009.

[10] S. Samanta, R. Beddingfield, I. Wong, and S. Bhattacharya, "Efficient power transfer to data center racks using medium voltage inductive coupling," in *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 1125–1130, Baltimore, MD, USA, 2019.

[11] Y. Berezovskaya, A. Mousavi, V. Vyatkin, and X. Zhang, "Smart distribution of IT load in energy efficient data centers with focus on cooling systems," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4907–4912, Washington, DC, 2018.

[12] F. Cuomo, A. Abbagnale, A. Cianfrani, and M. Polverini, "Keeping the connectivity and saving the energy in the internet," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pp. 319–324, IEEE, 2011.

[13] B. Heller, S. Seetharaman, P. Mahadevan et al., *ElasticTree: saving energy in data center networks//Proc of the 7th USENIX Conference on Networked Systems Design and Implementation*, USENIX, Berkeley, 2010.

[14] M. Wei, J. Zhou, and Y. Gao, "Energy efficient routing algorithm of software defined data center network," in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), IEEE*, pp. 171–176, Guangzhou, China, 2017.

[15] A. Fernandez-Fernandez, C. Cervello-Pastor, and L. Ochoa-Aday, "Achieving energy efficiency: an energy-aware approach in SDN," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, Washington, DC: IEEE Press, 2016.

[16] Z. Wu, X. Ji, Y. Wang, X. Chen, and Y. Cai, "An energy-aware routing for optimizing control and data traffic in SDN," in *2018 26th International Conference on Systems Engineering (ICSEng)*, pp. 1–4, Sydney: IEEE Press, 2018.

[17] J. Ba, Y. Wang, X. Zhong, S. Feng, X. Qiu, and S. Guo, "An SDN energy saving method based on topology switch and rerouting," in *2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–5, Taipei: IEEE Press, 2018.

[18] D. Li, Y. Shang, W. He, and C. Chen, "EXR: greening data center network with software defined exclusive routing," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2534–2544, 2015.

[19] M. Andrews, A. F. Anta, L. Zhang, and W. Zhao, "Routing for power minimization in the speed scaling model," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 285–294, 2012.

[20] G. Xu, B. Dai, B. Huang, J. Yang, and S. Wen, "Bandwidth-aware energy efficient flow scheduling with SDN in data center networks," *Future Generation Computer Systems*, vol. 68, pp. 163–174, 2017.

[21] Y. Zan, W. Ying, Q. Xuesong, and W. Yuqi, "Routing optimization algorithm for SD-DCN based on delay and energy consumption," *Journal of Beijing University of Posts and telecommunication*, vol. 43, no. 2, pp. 46–51, 2020.

[22] K. Fan, Y. Wang, J. Ba, W. Li, and Q. Li, "An approach for energy efficient deadline-constrained flow scheduling and routing," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 469–475, Washington DC, USA, 2019.

[23] D. Zats, T. Das, P. Mohan, D. Borthakur, and R. Katz, "Detail: reducing the flow completion time tail in datacenter networks," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication - SIGCOMM '12*, pp. 139–150, Helsinki, Finland, 2012.

[24] X. Huang, T. Yuan, G. Qiao, and Y. Ren, "Deep reinforcement learning for multimedia traffic control in software defined networking," *IEEE Network*, vol. 32, no. 6, pp. 35–41, 2018.

[25] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials, Fourth quarter*, vol. 21, no. 4, pp. 3133–3174, 2019.

[26] L. A. N. Julong, Y. U. Changhe, H. U. Yuxiang, and L. I. Ziyong, "A SDN routing optimization mechanism based on deep reinforcement learning," *Journal of Electronics and Information Technology*, vol. 41, no. 11, pp. 2669–2674, 2019.

[27] Z. Xu, J. Tang, J. Meng et al., "Experience-driven networking: a deep reinforcement learning based approach," in *IEEE INFOCOM 2018- IEEE Conference on Computer Communications*, Honolulu, HI, USA, 2018.

[28] M. B. Hossain and J. Wei, "Reinforcement learning-driven QoS-aware intelligent routing for software-defined networks," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Ottawa, ON, Canada, 2019.

[29] T. Ma, H. Yuxiang, and Z. Xiaohui, "Data center network coflow scheduling mechanism based on deep reinforcement learning," *Acta Electronica Sinica*, vol. 46, no. 7, pp. 1617–1624, 2018.

[30] Z. Yao, Y. Wang, and X. Qiu, "DQN-based energy-efficient routing algorithm in software-defined data centers," *International Journal of Distributed Sensor Networks*, vol. 16, no. 6, 2020.

[31] P. L. Timothy, J. H. Jonathan, P. Alexander et al., *Continuous Control with Deep Reinforcement Learning*, Proc. ICLR, San Juan, Puerto Rico, 2016.