Scientific Research Publishing

Open Journal of Applied Sciences, 2024, 14, 2223-2235
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Optimising Energy Consumption in SD-DCN Networks (Software Defined-Data Center Network)

Narcisse Tahi[1], Etienne Soro[2], Pacôme Brou[2], Olivier Asseu[1,2]

[1]UMRI STI, INPHB INP—Houphouët Boigny, Yamoussoukro, Côte d'Ivoire
[2]Ecole Supérieure Africaine des Technologies de l'Information et de la Communication (ESATIC), LASTIC Laboratory of ESATIC, Abidjan, Côte d'Ivoire
Email: broupacom@hotmail.fr

How to cite this paper: Tahi, N., Soro, E., Brou, P. and Asseu, O. (2024) Optimising Energy Consumption in SD-DCN Networks (Software Defined-Data Center Network). *Open Journal of Applied Sciences*, **14**, 2223-2235.
https://doi.org/10.4236/ojapps.2024.148149

Received: July 27, 2024
Accepted: August 20, 2024
Published: August 23, 2024

Abstract

Over the last decade, the rapid growth in traffic and the number of network devices has implicitly led to an increase in network energy consumption. In this context, a new paradigm has emerged, Software-Defined Networking (SDN), which is an emerging technique that separates the control plane and the data plane of the deployed network, enabling centralized control of the network, while offering flexibility in data center network management. Some research work is moving in the direction of optimizing the energy consumption of SD-DCN, but still does not guarantee good performance and quality of service for SDN networks. To solve this problem, we propose a new mathematical model based on the principle of combinatorial optimization to dynamically solve the problem of activating and deactivating switches and unused links that consume energy in SDN networks while guaranteeing quality of service (QoS) and ensuring load balancing in the network.

Keywords

DCN, Optimisation, Energy Consumption, QoS, SDN

1. Introduction

The development of intelligent connection equipment because of Industry 4.0 has led to rapid growth in traffic and the number of network devices, resulting in increased energy consumption in networks [1]. As a result, more and more companies are using Software Defined Networking (SDN), an emerging technique that separates the control plane and the data plane of the connected deployment network, enabling centralized control of the network while offering

flexibility in network management, to solve the energy problems of data center networks [2]. Energy consumption is a significant part of the total cost of information and communication technology [3]. Computer network equipment is designed to handle network traffic; however, the level of use of the device is not necessarily proportional to the energy it consumes [4]. For example, DCNs do not always operate at full capacity, but the fact that they support lower loads means minimal energy consumption. It has been shown that the DCN consumes too much energy unnecessarily when it is not fully loaded. Some research work is moving in the direction of optimizing energy consumption, but still does not ensure good performance and quality of service of the DCN network [5]. Therefore, we propose a new mathematical model based on combinatorial optimization principles to dynamically solve the problem of switch activation and deactivation as well as idle links and zero energy consuming activity in the DCN network, while guaranteeing Quality of Service (QoS) constraints. And ensuring load balance in the network. Part 2 presents different mathematical models. The problem statement is presented in Section 3. The proposed model is presented in Section 4. We conclude by summarizing the main contributions and highlighting some future directions in Section 6.

2. Related Works

This section of our paper presents important recent work that has used integer linear programming to solve the problem of optimizing energy consumption in SDN networks. To minimize power consumption in SDNs, Kra *et al.* [6] proposed a mathematical model in the form of an integer linear program. This model involves turning ports on and off at certain intervals to minimize power consumption. This model is flexible and can double the energy savings at port and link level, but does not take into account switch switching or load balancing in the network. A traffic-sensitive mathematical model for making energy consumption proportional to traffic in SDN networks was proposed by Assefa and Ozkasap in [7]. This model presents a multi-objective function that minimizes the total energy consumption of switches and links. The problem to be solved is defined as the allocation of links and switches for each flow. It is true that this model shows a significant reduction in energy consumption, but it does not take into account certain parameters such as timing, the elimination of redundant paths and, above all, load balancing in the network. Wu *et al.* in [8] presented a model similar to the previous one, but this time applied to SDN-based data center networks. The energy-efficient routing problem is modelled as a mixed-integer linear problem, which aims to minimize the network energy consumption according to the flow demand of the data center network. This model considers network redundancy by defining redundancy parameters within a limited framework to manage potential bandwidth emergencies. But it offers average requirements in terms of quality of service and load balancing. Lu *et al.* in [9] also presented the energy optimization problem in SDN-based data centers as an integer linear program that also aims to minimize the number of switches and

links activated to meet the traffic demand. This model takes into account traffic variability and uses multipath routing to satisfy flow allocation QoS in high traffic situations. In addition, the model presented does not take into account load balancing or end-to-end latency. Torkzadeh *et al.* in [10] proposed a model that minimizes energy consumption in SDN data centers and takes into account QoS parameters such as latency and packet loss rate. In addition, the link load balancing model in the network was added to the model, making the proposed model multi-objective. Nsaif *et al.* in [11] proposed an integer linear programming model for traffic-aware routing that minimizes the number of active links in order to minimize energy consumption. The model takes into account the correlation between links and traffic. In addition, this model does not take into account load balancing and the deactivation of under-used switches. Most of the above work uses integer linear programming to solve the energy optimization problem. Our contributions include the consideration of additional parameters such as time, bandwidth and the elimination of redundant paths. To avoid network congestion, we will ensure load balancing in the data center network.

3. Energy Consumption Model Based on Ear (Energy Aware Routing)

To reduce energy consumption in SDN data centre networks, we propose a model for optimising energy consumption based on EAR (Energy Aware Routing), which is a type of routing suitable for reducing the amount of energy. For data centres, a traffic-aware approach can save up to 50% energy during periods of low load [4]. The aim is to direct traffic to the most energy-efficient links while maintaining the required performance and quality of service. The basic idea behind (EAR) is that during periods of low traffic (e.g. at night), traffic requests can be routed via a subset of the network's links and switches while maintaining connectivity and quality of service. In this way, the links and switches excluded by the routing path can be put to sleep to save energy. We propose a general optimization model that determines the energy capacity of SDN from a traffic perspective, where the main energy-saving components considered are links and switches. Our proposed model collectively minimises the number of switches and links used to carry network traffic. For us, this will involve automatically putting idle switches and links into sleep mode as a first step, and then minimising the number of active network devices in order to minimise network energy consumption.

3.1. Graph Modelling of the Problem

We begin by presenting a few notations that will be used in the rest of the document. This notation is used below to construct our optimization model.

C: The switches in the network where i with ($i \in C$) represents switch i.

L: The links in the network where ($(i, j) \in L; i \neq j$), represents the link between switches i and j.

n: Total number of switches in the network.

C_{ij} : The bandwidth or capacity of the link connecting switches i and j with $i \neq j$.

E_i : Power consumption by switch i when it is activated.

E_{ij} : Energy consumption of the link between switch i and j with ($i \neq j$).

F: All network flows $f \in F$.

$f_{ij} = \left(sr, ds, \lambda_f\right)$: flows between switches i and j.

ω_i : Binary variable. If $\omega_i = 1$ then switch i is enabled; otherwise $\omega_i = 0$.

σ_{ij} : Binary variable, if $\sigma_{ij} = 1$ then a link between switches i and j is activated, 0 otherwise.

In this paper, we consider the FAT TREE data centre network as a weighted directed graph $G = (C, L)$ where $C = \{1, \cdots, n\}$ represents a finite set of nodes (switches) in the graph; ($(i, j) \in L; i \neq j$) represents a finite set of arcs (links) between switches, where each link is used for communication in both directions. Figure 1 shows two switches i and j interconnected by an (i, j) link.

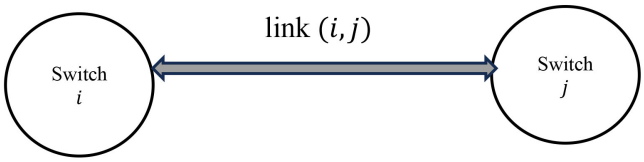

Figure 1. Representation of the link between two switches.

3.2. Network Model

The network topology is modelled using a weighted directed graph $G = (C, L)$ with a set of vertices C, and a set of edges L. Each node is an OpenFlow switch and a switch is represented by the letter i. The role of each switch is to transmit information according to the path chosen by the network controller. Each edge of the graph is a link. The transmission of packets between two ports forms a link. The link between switches i and j is called (i, j) . The links and switches in the network can be in ON or OFF mode.

We define the binary variables ω_i and σ_{ij} : ω_i to specify the current mode of the switch. is 1 if switch i is ON, 0 otherwise. Furthermore, the variable σ_{ij} specifies the mode state of the link between two connected ports. $\sigma_{ij} = 1$ if the link between switches i and j is active and 0 otherwise.

We define another binary variable f_{ij} to identify flows on the DCN topology. A flow represents a group of packets with the same source and destination addresses, using the same link to reach the destination.

The variable f_{ij} takes the value 1 if the flow f crosses the link $(i; j)$ with $(i \neq j)$ and the value 0 otherwise. The link utilisation matrix, U, is constructed by considering the utilisation of each link. DCN traffic is represented by the set of flows $\mathbb{F}$, where each $f \in \mathbb{F}$ is defined as $f_{ij} = \left(sr, ds, \lambda_f\right)$. Source and destination switches are represented respectively by sr and ds. λ_f , transmission rate of f packets measured in bits per second.

In this model, we consider links and switches as the main energy-consuming

components of the network. Our deployment model allows them to be dynamically enabled and disabled to save energy. To support distributed traffic, the model uses the smallest possible set of active links and the fewest possible switches. Our optimisation model considers the following features:

1) For the sake of simplicity, the model parameters are based on the state of the network at a given time.

2) The model is defined as a problem of flows of various commodities.

3) the problem is defined as allocating links and switches to each flow while minimising the total number of active links and switches.

4) Given the model formulation and parameters, the result of the optimizer is a list of active links and switches for each flow $f \in F$.

5) In our model, we assume that all links have the same energy consumption and that this is a fixed constant. Switches also have a similar constant energy consumption. In other words, and are constants in our problem.

3.3. Mathematical Formulation of the Problem

We formulate the SD-DCN energy optimisation problem as an integer linear programme (ILP). To better characterise the problem, we define binary variables for each switch and link in the network.

ω_i : Binary variable indicating whether switch i is active (1) or on standby (0).

$$\omega_i = \begin{cases} 1, \text{ if the Switches } i \text{ is activated} \\ 0, \text{ otherwise} \end{cases}$$

σ_{ij} : Binary variable indicating whether the link between switches i and j is active (1) or not (0).

$$\sigma_{ij} = \begin{cases} 1, \text{ if a link between switches } i \text{ and } j \text{ is activated} \\ 0, \text{ otherwise} \end{cases}$$

f_{ij} : Packet flow over the link between switches i et j.

The optimisation problem described at the beginning of this section can be modelled as follows:

$$\min\left(\sum_{i=1}^{n} \omega_i \cdot E_i + \sum_{i=1}^{n} \sum_{j=1;i\neq j}^{n} \sigma_{ij} \cdot E_{ij}\right) \tag{1}$$

At the same time as relation (1), certain constraints are necessary to meet the limitations of the problem.

$\forall i; j$ j among the switches, with, $i \neq j$,

$$\omega_i + \omega_j - 1 \leq \sigma_{ij} \leq \omega_i; \sigma_{ij} \leq \omega_j \tag{2}$$

$$\sum_{i=1}^{n} \omega_i \leq K \tag{3}$$

$$f_{ij} \leq C_{ij}\sigma_{ij}, \forall i; j \text{ et } i \neq j \tag{4}$$

$$\sum_{i\neq k} f_{ik} = \sum_{j\neq k} k_j, \forall k \notin S \cup T \tag{5}$$

$$\sum_{j:(s,j)\in \text{link}} f_{sj} - \sum_{i:(i,s)\in \text{link}} f_{is} \geq D_{st}, \forall (s,t) \notin S \times T \tag{6}$$

$$\sum_{i:(i,t)\in \text{link}} f_{it} - \sum_{j:(t,j)\in \text{link}} f_{tj} \geq D_{st}, \forall (s,t) \notin S \times T \tag{7}$$

$$\omega_j \leq \sum_{i \neq j} \sigma_{ij} + \sum_{i \neq j} \sigma_{ji}, \forall j \tag{8}$$

$$\omega_j \leq \sum_{i \neq j} \left(f_{ij} + f_{ji} \right), \quad \forall j \quad \text{is neither source nor destination} \tag{9}$$

Relation (1) is made up of two terms; the first term represents the energy consumption of the links traversed by a network flow, and the second term concerns the energy consumption of all the active switches in the network. Constraint (2) concerns network connectivity. Constraint (3) stipulates that the controller cannot simultaneously manage a certain number of switches more than the maximum capacity. The constraint is given in Equation (4), which represents the flow constraint and states that the total packet throughput between two switches cannot exceed the capacity of the link. Constraint (5) represents the flow conservation constraint and states that the number of incoming and outgoing packets for switches that are neither the destination nor the source of the flow must be equal. Constraints (6) and (7) are path constraints for the source and destination and indicate that the flow from the source switch must reach the destination switch. Constraint (8) ensures that no flow passes through an idle switch. Constraint (9) states that if no flows pass through the links connected to a given switch, the switch will be disabled.

Relation (1) is made up of two terms; the first term represents the energy consumption of the links traversed by a network flow, and the second term concerns the energy consumption of all the active switches in the network.

Constraint (2) concerns network connectivity. Constraint (3) states that the controller cannot simultaneously manage a certain number of switches in excess of the maximum capacity. The constraint is given in Equation (4), which represents the flow constraint and states that the total packet throughput between two switches cannot exceed the link capacity. Constraint (5) represents the flow conservation constraint and states that the number of incoming and outgoing packets for switches that are neither the destination nor the source of the flow must be equal. Constraints (6) and (7) are path constraints for the source and destination and indicate that the flow from the source switch must reach the destination switch. Constraint (8) ensures that no flow passes through an idle switch. Constraint (9) states that if no flows pass through the links connected to a given switch, the switch will be disabled.

4. Design of the Algorithm

The optimization problem described in the previous section can be solved using optimization software such as CPLEX or Gurobi [9], but the computation time increases exponentially with the size of the network. In a real network scenario with hundreds or thousands of nodes, it is difficult to compute the optimal solution in a limited time, especially with a large number of traffic re-

quests.

Our problem solving is based on the EAR technique, which is a multi-objective problem and therefore consists of finding the set of paths that minimizes the number of network links and switches. The solution we propose is an algorithm for finding this set of paths. Let's assume that the network contains switches. If we compute all the paths in the network, then the time complexity of the algorithm is $O(n!)$ [12]. Various algorithms and methods such as dynamic programming, interactive methods, evolutionary algorithms, etc. have been deployed and studied to solve this type of problem. We propose a heuristic to solve the problem. This heuristic is divided into two parts: the first stage takes as input, at a given time, the initial topology of the FAT TREE data center, the state of the links and switches and provides as output a subset of active links and switches which is in fact a basic topology responsible for routing. In the first phase, using the SDN controller's monitoring module, all inactive switches and links are deactivated to make way for the sub-topology responsible for the traffic, as shown in the algorithm below.

Algorithm 1. Heuristics for building an energy-efficient sub-topology.

Input: Topology SD-DCN (FAT TREE),
Link capacity: W
Link status: Link_statut
Switch status: Switch_status
Output: Sub-topology (Set of enabled switches and links)
Until *Link_statut*==0 **Do**
1. Deactivate the link
End Until
2. **Until Link_statut**==0 Do
3. Deactivate the link
End Until
4. **For** all links;
5. List activated links
EndFor
6. **For** all links **Do**
7. List links actived
EndFor
8. **Return** a sub-graph of all activated links and switche

The second phase will involve the installation of flows through different conduits. To avoid network congestion based on the initial topology and ensure good quality of service, our algorithm redirects certain flows and balances the network load (**Algorithm 1**).

In **Algorithm 2**, we use the first Find Paths function on line 1 to initialize an empty path list. On lines 2 and 3 for each node in the topology, it checks the neighbors of that node. On lines 4 and 5, if the capacity between the node and its neighbor is less than or equal to the capacity of the link, it calls the Find Paths function. On lines 6 and 7, if a valid (non-empty) path is found, it is added to the

list of paths.

Algorithm 2. Energy-saving traffic management algorithm.

Input: SD-DCN (FAT TREE) sub-topology: newTopology
Link capacity: W
Set of paths: Epath
Set of possible paths: Epossible_path
Traffic matrix: MT
Output: Set of candidate paths for routing flows
1. **Function** findPaths (newTopology, Link capacity):
Paths <- ∅
2. **For** each node in newTopology:
3. **For** each node neighbour:
4. **If** capacity(node neighbour) ≤ Link Capacity:
5. Path <- FindPath(node, neighbour, capacity Link)
6. If Path ≠ ∅:
7. add path to paths
8. **Return** Paths
9. **Function** FindPath(node, neighbour, capacity Link):
10. Path <- [node]
11. current node <- node
12.**Until** current node ≠ neighbout:
13. Find <- False
14.**For** each nextNode in neighbours (Currentnode):
15. **If** capacity(currentNode, nextNode) ≤ Linkcapacity and nextNode isn't in the Path:
16 add nextNode to path
17. CurrentNode <- nextNode
18. Find <- False
19. Getting out of the loop **For**
20. **If** not find:
21. **Return** ∅
22. **Return** path

For the second function from line 9, takes as parameters the node, the neighboring node and the link capacity. It initializes a path with the starting node (line 10). It analyses the neighbors of the current node to find a valid path to the target neighbor (lines 11 to 13). Then, if a valid path is found, it will be returned. Otherwise, the function returns an empty set (indicating no path found) from lines 14 to 22.

5. Simulations and Results

In this section, we describe the evaluation of our energy approach and analyze the results obtained. We used the linear programming solver Gurobi Optimizer [13] to evaluate the performance of the ILP model and the heuristic algorithm developed in Python. All calculations were performed on a computer equipped with an Intel Core i7 at 2.80 GHz and 16 GB of RAM. We ran our simulation using a FAT TREE topology with $k = 4$.

To illustrate the energy savings achieved by the algorithm, we define the percentage of energy savings as an indicator of energy savings. We define an energy savings index E.S(%). It is given by this formula:

$$\text{E.S}(\%) = \left(1 - \left(\text{Energy Obtained}\right) / \left(\text{Initial Energy}\right)\right) \times 100$$

In our experiment, the percentage energy saving is a function of the number of active switches and links in the network for the initial topology and then it is a function of the number of flows in the network, *i.e.* the new topology responsible for the traffic at a given moment. It should be noted that this topology changes at each instant due to the random state of the links and switches. We assume that each link (i, j) has the same energy consumption $E_{ij} = 0.6\,\text{kW}$ and that each switch consumes the same power $E_i = 3\,\text{kW}$ [14]. This table shows the variation in the number of switches and links activated from the initial topology. Consequently, the variation in energy consumption before data transmission. Our algorithm for building a sub-topology proves to be energy-efficient, as can be seen from Table 1.

Table 1. Variation in energy after generation of a sub-topology.

Instances	Initial topology energy	Energy obtained after applying Algorithm 1	Ratio E.S (%)
1	79.2	55.8	29.54
2	79.2	54.6	31.06
3	79.2	64.2	18.93
4	79.2	56.4	28.78
5	79.2	54	31.81
6	79.2	49.8	38.03

Figure 2 shows an example of an initial topology to which we apply our sub-topology algorithm. The aim is to deactivate (put to sleep) these inactive switches and links in red, which are consuming unnecessary energy. Activated switches and links are shown in green in Figure 2.

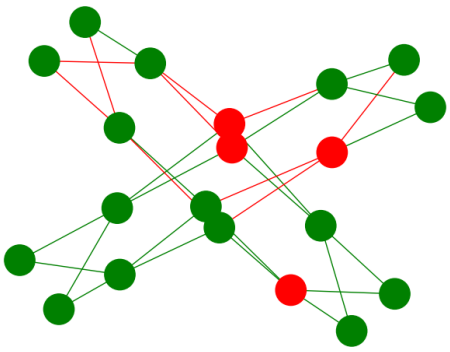

Figure 2. An initial FAT TREE $k = 4$ topology.

The graph below clearly shows a reduction in energy consumption after disabling inactive links and switches at a given time in the FAT TREE network with $k = 4$. This reduction has an average energy saving ratio of around 29.7%. This energy reduction is shown in Figure 3 below.

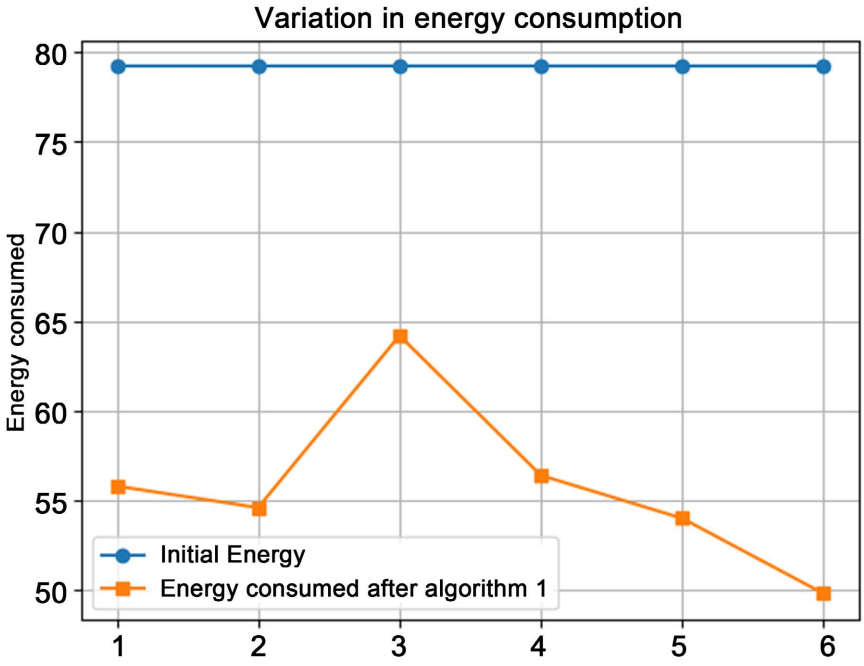

Figure 3. Graph of variation in energy consumption for Algorithm 1.

In the second part of our simulations, we will apply Algorithm 2 to our new traffic management topology. To evaluate the effectiveness of our algorithm, we compare the energy savings ratio as shown in Table 2, the algorithm for minimum energy consumption in the framework of the Random-order Demand Minimum Energy Consumption (RD-MEC) strategy of the author [9] with our own under the same experimental conditions, *i.e.* we use the same FAT TREE topology with $k = 4$ and the same proportions of elephant-mouse flows (1:9) and (2:8) generated randomly.

Table 2. Comparison of the RD-MEC algorithm and the H2ES algorithm in the proportion (1:9).

Instances	Number of Flows	Energy Saving Ratio RD-MEC (%)	H2ES Energy Saving Ratio (%)
1	100	35	38.63
2	200	35	36.1
3	300	34	35.2
4	400	33	33.4
5	500	30	30
6	600	27	28

The graph in Figure 4 below compares the energy-saving ratio of the RD-MEC algorithm and our algorithmic approach in proportions (1:9).

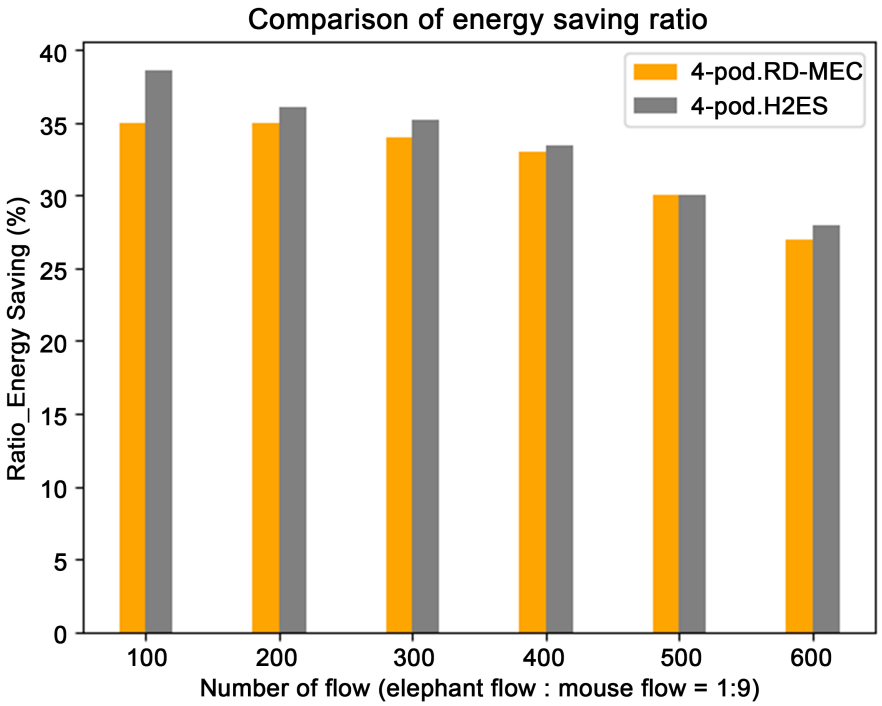

Figure 4. Energy saving ratio as a function of the number of flows (1:9).

The graph above shows that our algorithmic approach has a much higher ratio than that of the author [9]. With regard to the number of flows in the proportions (2:8), we obtain Table 3.

Table 3. Comparison of the RD-MEC algorithm and our H2ES algorithm in the proportion (2:8).

Instances	Number of flows	Energy Saving Ratio RD-MEC	H2ES Energy Saving Ratio
1	100	25	25.63
2	200	35	28.1
3	300	30	32.2
4	400	25	27.8
5	500	15	17.3
6	600	10	11

The graph below in Figure 5 compares the energy saving ratio of the RD-MEC algorithm and our algorithmic approach in the proportions (2:8).

In the end, our algorithm is no worse than the RD-MEC approach. It provides an average energy saving ratio of 23.67% in the case where the number of elephant flows: mouse flows (1:9) is 0.34% higher than that of RD-MEC.

For the proportion of elephant flows: mouse flows of (2:8) we observe an en-

ergy saving of 33.55%, *i.e.* a ratio of 1.22% above the energy saving ratio of the author [9].

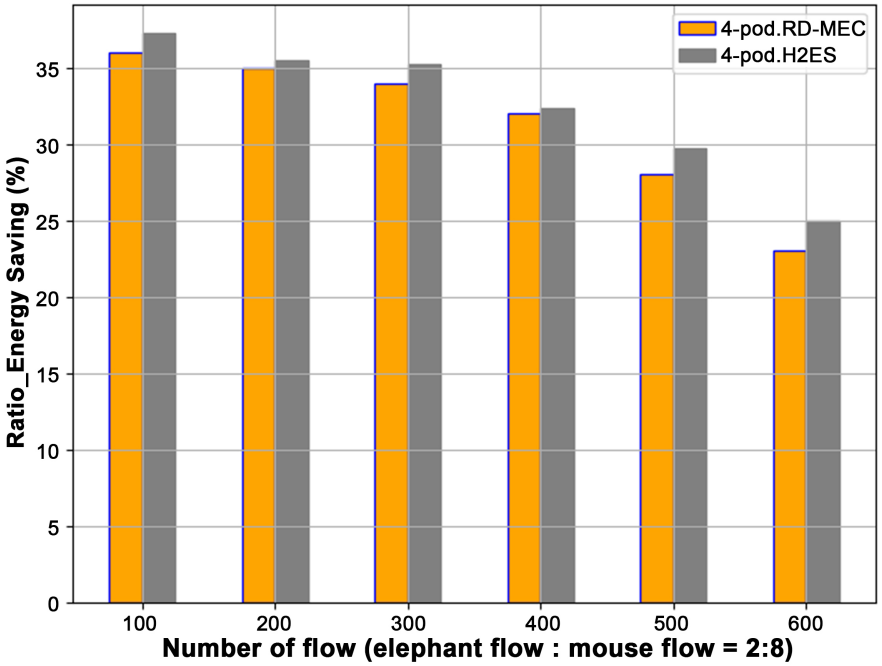

Figure 5. Energy saving ratio as a function of the number of flows (2:8).

6. Conclusion

In this article, we propose a mathematical model based on integer linear programming. This model makes it possible to reduce energy consumption in data centre networks by disabling network links and switches. Our model integrates all QoS parameters. Through our proposed heuristic for solving the model, the set of paths found for dynamic routing of data flows minimizes the energy consumption of FAT TREE data center. However, we intend to work on further improving QoS using machine learning algorithms. Our ultimate objective is to use the results of our simulations to implement the system in a real enterprise environment.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Giroire, F., Moulierac, J., Phan, T.K. and Roudaut, F. (2015) Minimization of Network Power Consumption with Redundancy Elimination. *Computer Communications*, **59**, 98-105. https://doi.org/10.1016/j.comcom.2014.12.002

[2] Tu, R., Wang, X. and Yang, Y. (2014) Energy-Saving Model for SDN Data Centers. *The Journal of Supercomputing*, **70**, 1477-1495. https://doi.org/10.1007/s11227-014-1237-3

[3] Chiaraviglio, L., Mellia, M. and Neri, F. (2012) Minimizing ISP Network Energy Cost: Formulation and Solutions. *IEEE/ACM Transactions on Networking*, **20**, 463-476. https://doi.org/10.1109/tnet.2011.2161487

[4] Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee S. and McKeown, N. (2010) Elastictree: Saving Energy in Data Center Networks. chez *Nsdi*, 2010.

[5] Hammadi, A. and Mhamdi, L. (2014) A Survey on Architectures and Energy Efficiency in Data Center Networks. *Computer Communications*, **40**, 1-21. https://doi.org/10.1016/j.comcom.2013.11.005

[6] Kra, L., Gondo, Y., Gooré, B.T. and Asseu, O. (2018) Contribution to the Optimization of the Energy Consumption in SDN Networks. *Journal of Sensor Technology*, **8**, 59-67. https://doi.org/10.4236/jst.2018.83005

[7] Assefa, B.G. and Özkasap, Ö. (2019) A Survey of Energy Efficiency in SDN: Software-Based Methods and Optimization Models. *Journal of Network and Computer Applications*, **137**, 127-143. https://doi.org/10.1016/j.jnca.2019.04.001

[8] Wu, Z., Ji, X., Wang, Y., Chen, X. and Cai, Y. (2018) An Energy-Aware Routing for Optimizing Control and Data Traffic in SDN. 2018 26*th International Conference on Systems Engineering* (*ICSEng*), Sydney, 18-20 December 2018, 1-4. https://doi.org/10.1109/icseng.2018.8638019

[9] Lu, Z., Lei, J., He, Y., Li, Z., Deng, S. and Gao, X. (2019) Energy Optimization for Software-Defined Data Center Networks Based on Flow Allocation Strategies. *Electronics*, **8**, Article 1014. https://doi.org/10.3390/electronics8091014

[10] Torkzadeh, S., Soltanizadeh, H. and Orouji, A.A. (2021) Energy-Aware Routing Considering Load Balancing for SDN: A Minimum Graph-Based Ant Colony Optimization. *Cluster Computing*, **24**, 2293-2312. https://doi.org/10.1007/s10586-021-03263-x

[11] Nsaif, M., Kovásznai, G., Rácz, A., Malik, A. and de Fréin, R. (2021) An Adaptive Routing Framework for Efficient Power Consumption in Software-Defined Datacenter Networks. *Electronics*, **10**, Article 3027. https://doi.org/10.3390/electronics10233027

[12] Zhu, H., Liao, X., de Laat, C. and Grosso, P. (2016) Joint Flow Routing-Scheduling for Energy Efficient Software Defined Data Center Networks. *Journal of Network and Computer Applications*, **63**, 110-124. https://doi.org/10.1016/j.jnca.2015.10.017

[13] Gurobi optimization (2016) Gurobi Optimizer Reference Manual. http://www.gurobi.com

[14] Wang, R., Gao, S., Yang, W. and Jiang, Z. (2017) Energy Aware Routing with Link Disjoint Backup Paths. *Computer Networks*, **115**, 42-53. https://doi.org/10.1016/j.comnet.2017.01.015