



# Energy and Congestion Awareness Traffic Scheduling in Hybrid Software-Defined Network with Flow Splitting

C. S. Lim, S. C. Tan<sup>(✉)</sup>, A. Imanuel, J. S. L. Teh, and N. Q. Baderulhisham

Faculty of Computing and Informatics (FCI), Multimedia University, Cyberjaya, Malaysia  
sctan1@mmu.edu.my

**Abstract.** Software-Defined Networking (SDN) has received a lot of interest in recent years because of its benefits over network controllability. Nonetheless, the deployment of SDN in legacy networks is likely to take months or years due to funding constraints. Traffic scheduling that involve flow splitting provides the flexibility for traffic flow. It is able to minimize the maximum link capacity of a network and to reduce the traffic congestion in the network. The majority of the studies focus on how to balance the flows coming out of the conventional nodes and how to partition the flows that gather at the SDN nodes so that the maximum link usage of the entire network can be reduced. Energy efficiency of a network are important to save cost and energy. During traffic scheduling, the energy consumption of a traffic flow should be considered. As a result, in hybrid SDN, we offer a heuristic approach for energy and congestion awareness traffic scheduling with flow splitting. We first define the aforementioned issue in an Integer Linear Programming (ILP) model, and then we assess the suggested ILP model and heuristic algorithm in terms of solution quality and processing time. The findings indicate that with polynomial time complexity, our suggested approach retains its overall soundness.

**Keywords:** Energy-awareness · congestion · hybrid Software Defined Networking

## 1 Introduction

SDN promotes potential qualities as the next generation communication network architecture [1, 2], although obstacles occur from SDN adoption in legacy networks. Network operators are often concerned with three key aspects of challenges: economic, organizational, and technological. The cost of converting old infrastructure to SDN-enabled equipment is high, making SDN implementation a significant financial burden for network operators. Due to financial constraints, the implementation of SDN is likely to span numerous months or years, particularly for large networks with thousands or more nodes. As a result, during the transition of SDN deployment, hybrid SDN (hSDN) is produced, in which a network comprises of both traditional and SDN-enabled nodes to act as a whole. Furthermore, network stability remains a high need in the hybrid SDN context.

With the advent of SDN technology, dynamic routing route management for flows becomes feasible. An OpenFlow network [3], the most widely used SDN enabler, is made up of OpenFlow-enabled switches and a central controller. The controller plans routing pathways and subsequently manipulates flow tables in switches using the OpenFlow protocol. The SDN controller may centrally regulate the network state by altering the flow tables that routers maintain [4–8]. Through the use of SDN, network administrators may partition arbitrary flows to outgoing links in a flexible manner. However, SDN has its own set of deployment issues, making comprehensive SDN implementation problematic in the immediate future. Some approaches can be used to route multiple flows to the same destination to separate next hops, resulting in traffic splitting at the network layer. The deployment of SDN in the network provides a straightforward and effective approach to undertake traffic engineering and can significantly increase network performance. The majority of the work is on how to balance the flows coming out of the normal nodes and how to separate the flows that gather at the SDN nodes so that the maximum link usage of the total network may be reduced. Traffic scheduling that involve flow splitting provides the flexibility for traffic flow. It is able to minimize the maximum link capacity of a network and to reduce the traffic congestion in the network. Energy efficiency [9–12] of a network are important to save cost and energy. The energy consumption of a traffic flow should be taken into account during traffic scheduling. Therefore, in this paper, we explore energy and congestion awareness traffic scheduling in hybrid software-defined network with flow splitting. The proposed method provides complete solution to H-SDN in terms of: Traffic congestion by minimizing maximum link utilization with flow splitting and cost optimization by selecting path with low energy consumption.

The remainder of this article is organized as follows. The second portion analyses previous research, and the third section offers the energy and congestion-aware traffic scheduling in hybrid SDN problem formulation in Integer Linear Programming (ILP) model and a heuristic algorithm is provided. Section 4 evaluates and discusses the results of the proposed method with various network sizes. Section 5 brings our paper to a close.

## 2 Related Works

In this part, we show some recent relevant work. Firstly, [13] AI detecting congestion on network service, obtain policy associated with the congested network service and causing bandwidth on demand in the network to mitigate the congestion. However, it does not consider congestion issue in MLU in H-SDN by proposing a routing algorithm. In [14], machine learning method is used for the optimal route selection algorithm to solve the quick dynamic routing of different business stream in SDN. It is applicable to SDN and congestion issue is not considered. In [15], Path Computation Element (PCE) is used as a central controller of the SDN network to transition a traditional network to an SDN enabled network and focusing on SDN migration. In [16], open shortest path first (OSPF) routing is used for H-SDN to determine the network topology and send tunneling instructions between 2 non-SDN portion separated by an SDN portion. It does not consider to minimize the MLU in traffic routing. Instead, it uses OSPF purely for determining the network topology in non-SDN and SDN networks. In [17], a centralized control plane device is implemented to select a candidate path for forwarding

data traffic in SDN. The work in [18] providing tools to manage the hybrid network for conceptualizing the overall security and functionality requirements of a network and plan how these can be satisfied using a hybrid network parts as appropriate. The tools are to find paths in networks satisfying access-control, capacity, bandwidth and routing policy constraints by utilizing simultaneous multi-threading (SMT) solver. However, it is focusing on legacy policy enforcement of SDN instead of routing.

The work in [19] is using a hybrid routing table which is in the form of hash table for network traffic routing to strike a balance between lookup performance and memory. In [20], Path selection update processes that are optimized for a hybrid network in which many hybrid devices may be used for a certain path may be used. Path updating for load balancing may be affected by whether a packet stream is elastic or inelastic. An approach [21] that using hybrid networking devices along with advanced control algorithm to implement path selection, load balancing, stream splitting/aggregation and packet loss minimization. Link capacity and bandwidth is being considered for path selection. This approach is implemented for hybrid network. In [22], Path selection mechanisms may also entail determining end-to-end path capacity for various pathways. End-to-end route capacity may be determined in part by contention groups associated with shared media. A hybrid network device is implementing an automatic path selection system to modify the previously selected path associated with a packet stream [23]. The path must not be exceeding the medium utilization threshold and the network interface is working. This approach is able to select the best network path, maximize bandwidth and avoid packet drops.

However, many of the contributions are only applicable to SDN. The hybrid SDN also need to be take into consideration because many companies are in the migration stage to SDN. Besides that, the early work also having limitation to minimize the maximum link utilization and energy together in a network, at the same time include the flow splitting in traffic route. The summary of related work as shown in Table 1.

**Table 1.** Summary of Energy-aware traffic scheduling and routing

Reference	Objective & Contribution	Issue and Challenges
[13]	Detecting network service congestion, obtaining policy related with the congested network service, and creating bandwidth on demand in the network to alleviate congestion	It does not consider congestion issue in MLU in H-SDN by proposing a routing algorithm
[14]	Machine learning method is used for the optimal route selection algorithm to solve the quick dynamic routing of different business stream in SDN	It is not H-SDN and congestion issue is not considered

*(continued)*

**Table 1.** (continued)

Reference	Objective & Contribution	Issue and Challenges
[15]	Path Computation Element (PCE) is used as a central controller of the SDN network to transition a traditional network to a SDN enabled network	Focusing on SDN migration instead of traffic routing in H-SDN
[16]	A method of open shortest path first (OSPF) routing is used for H-SDN to determine the network topology and send tunneling instructions between 2 non-SDN portion separated by an SDN portion	It does not consider to minimize the MLU in traffic routing. Instead, it uses OSPF purely for determining the network topology in non-SDN and SDN networks
[17]	A centralized control plane device is implemented to select a candidate path for forwarding data traffic	SDN is the focus of the above approach however, H-SDN is excluded
[18]	Providing tools to manage the hybrid network for conceptualizing the overall security and functionality requirements of a network and plan how these can be satisfied using a hybrid network parts as appropriate. The tools are to find paths in networks satisfying access-control, capacity, bandwidth and routing policy constraints by utilizing simultaneous multi-threading (SMT) solver	The above approach is on SDN. However, it is focusing on legacy policy enforcement of SDN instead of routing,
[19]	Using a hybrid routing table which is in the form of hash table for network traffic routing to strike a balance between lookup performance and memory	The approaches are in legacy network instead of H-SDN
[20]	Path selection update processes that are optimized for a hybrid network in which many hybrid devices may be used for a certain path may be used. Path updates for load balancing may be affected by whether a packet stream is elastic or not	

(continued)

**Table 1.** (continued)

Reference	Objective & Contribution	Issue and Challenges
[21]	An approach that using hybrid networking devices along with advanced control algorithm to implement path selection, load balancing, stream splitting/aggregation and packet loss minimization. Link capacity and bandwidth is being considered for path selection. This approach is implemented for hybrid network	
[22]	The path selection method may also compute end-to-end path capacity for potential pathways. End-to-end route capacity might be determined in part by contention groups associated with shared media	
[23]	A hybrid network device is implementing an automatic path selection system to modify the previously selected path associated with a packet stream. The path must not be exceeding the medium utilization threshold and the network interface is working. This approach is able to select the best network path, maximize bandwidth and avoid packet drops	

### 3 Energy and Congestion Awareness Traffic Scheduling in Hybrid Software-Defined Network with Flow Splitting

Taking into account the elements and restrictions mentioned, we provide a hybrid SDN traffic scheduling method in this paper by constructing an energy and congestion awareness traffic scheduling issue in an ILP model. In addition, the developed ILP model intends to improve network performance by reducing energy consumption and congestion during SDN implementation. Several inputs are required, which are mentioned below:

#### 3.1 Model Formulation

All math's formulation/equation: Input notation, Objective functions, constraints.

List of notations:

- $E$ : set of all links
- $R$  within  $V$ : set of objects  $R$  requested by SDN nodes

- $O$ : set of network object (k)
- $S\{O\}$  within  $V$ : set of nodes store object
- $P$ : set of all path
- $PathWithLink \{P\}$  within  $E$ : path in term of link
- $Path\{i, j\}$  within  $P$ : path number
- $C\{E\} \geq 0$ : The link's capacity  $C$  on the link  $E$  must be greater or equal to zero with default value
- $B\{E\} \geq 0$ : Background traffic  $B$  on the link  $E$  must be greater than or equal to zero.
- $d_{\{i,k,s\}}$ : Traffic demand  $i \in R, k \in RR[i], S\{O\}$  must be greater than zero and default 0.
- $energyfactor\{E\}$ : energy factor of link  $E$
- $totaltrafficDemand = \sum_{i,k,j} d(i, k, j)$ : Total traffic demand.  $i \in R, k \in RR[i], j \in s\{O\}$ .

List of variables

- $Weight_{\{V,V\}}$ : Weight of nodes two dimensional
- $L \in (0, 1)$ : maximum link utilization.
- $flowInLink\{E\}$ : flow in link  $E$ .
- $flowForRequest\{i, k, j, Path\}$ : Source destination for the flow in path.  $i \in R, k$  in  $RR[i], j$  in  $S\{k\}, Path[j, i]$
- $energy = \sum_{e \in E} flowInLink\{e\}.energyFactor\{e\}$ : energy for each link  $E$ .

The objective function is to minimize the maximum link utilization by energy and total Traffic demand:

$$\min L + \frac{energy}{totalTrafficDemand} \quad (1)$$

Subjects to the constraints:

$$flowInLink\{e\} + b\{e\} \leq c\{e\} \leq 0 \quad \forall e \in E \quad (2)$$

Constraint (2) Flow in link plus Background traffic on the link must be less than or equal to the link's capacity.

$$\sum_{i,k,j,p[j,i]} flowForRequest[i, k, j, p] = \sum_{j,k,l} d[j, k, l] \quad (3)$$

$$\forall i \in R, k \in RR[i], j \in S[k]$$

Constraint (3) Source destination for the flow in path must be equal to total traffic demand. Traffic demand  $i \in R, k \in RR[i], S\{O\}$  must be greater than zero and default 0.

$$flowForRequest\{i, k, j, p\} \geq 0 \quad \forall i \in R, k \in RR[i], j \in S[k], p \in Path \quad (4)$$

Constraint (4) Source destination for the flow in path must be greater or equal to zero to ensure there is a flow in a path.

$$flowInLink\{e\} = \sum_{i,k,j,p[j,i],e2} flowForRequest\{i, k, j, p\} \quad (5)$$

$\forall e \in E$

Constraint (5) is flow in link E must be equal to source destination for the flow in path.

$$flowInLink\{e\} + b\{e\} \leq L.c\{e\} \quad (6)$$

$\forall e \in E$

Constraint (6) Summation of source destination for the flow in path Background traffic B on the link E less than equal to maximum link utilization multiply the link's capacity C on the link E.

### 3.2 Heuristic Algorithm

The calculation of an MLU optimized energy and congestion aware traffic scheduling in hybrid SDN with respect to restrictions was shown to be an NP-complete issue [6]. The computation of an optimum traffic scheduling in the solution space is time consuming. For big networks, brute forcing every potential combination is wasteful. As a result, taking into account the limitations described in this part, we suggest a heuristic method for the given NP-complete issue. Figure 1 depicts a flowchart/pseudo-code description of the suggested method.

```

1. Start
2. Get user input from data file
3. Find all possible paths for all nodes
4. Find the shortest path for each request
5. Set total energy = 0
6. FOR each request
7.   Set fit = true
8.   FOR each link in shortest path
9.     IF traffic demand > link capacity
10.      fit = false
11.    END for loop
12.  ENDIF
13. END for loop
14. IF fit = true
15.   FOR each link in shortest path
16.     energy = traffic demand * energy factor
17.     total energy = total energy + energy
18.     Link capacity = link capacity - traffic demand
19.   END for loop
20. ELSE
21.   END for loop
22. ENDIF
23. END for loop
24. IF fit = true AND total energy < threshold energy
25.  Print results
26. Else
27.  Print no results
28. ENDIF

```

**Fig. 1.** The proposed Algorithm

The proposed algorithm searching for the shortest path for each request as shown in line 3 and line 4 in the Fig. 1. If the link capacity unable to support the traffic demand on any link of the shortest path, that path will be eliminated for choosing as the route as shown in line 8 to line 12. Then, the proposed algorithm will find the total energy on each chosen route by summation of the energy in each link as line 15 to line 19. Finally, the route that fulfil the link capacity and threshold energy is selected as best route as in line 24 to line 27. The MLU is calculated according to the target node with highest traffic flow. The path chosen to assign traffic flow are path with lowest energy and the MLU of the path is less than MLU value calculated earlier. If the MLU value for a path is less than or equals to MLU value calculated earlier, split the traffic flow to another path. An assign-remove technique is being used if no path is available for the traffic flow. It will restore the related traffic assigned previously to reduce the MLU value of the current path and assign the new traffic flow. The current path will then be locked to avoid restoring. The details of flowchart of the proposed algorithm as shown in Fig. 2.

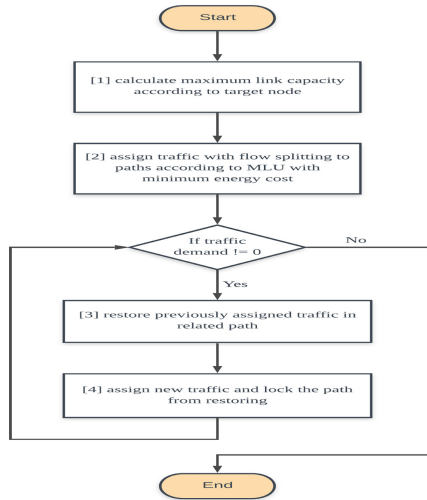
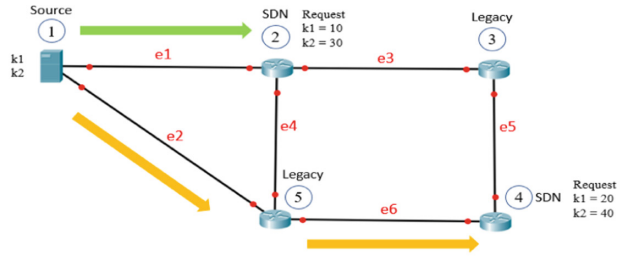


Fig. 2. The flowchart of the proposed algorithm

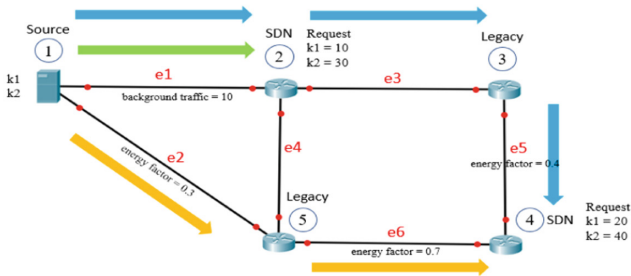
### 4 Result and Discussion

In this part, we will exhibit the simulation results and compare the performance of our suggested heuristic method to that of the ILP formulation model. Simulations are conducted on a Dell Poweredge T330 server with 16 GB RAM installed. For ILP formulation model, simulations are coded in AMPL environment with CPLEX solver and the proposed heuristic algorithm is coded in Java version 1.8. In terms of temporal complexity, our suggested technique outperforms the results of the ILP formulation model while preserving overall accuracy. To test our proposed heuristic approach, we ran tests in two different real-world topologies, a 5-node mesh topology in Fig. 3 and a 10-node mesh

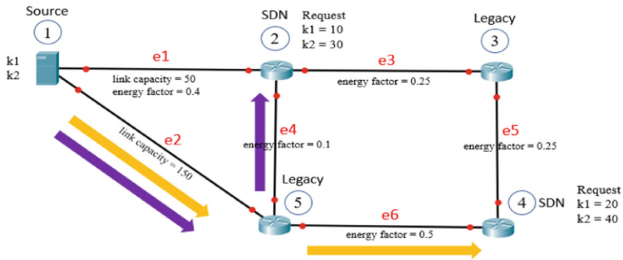




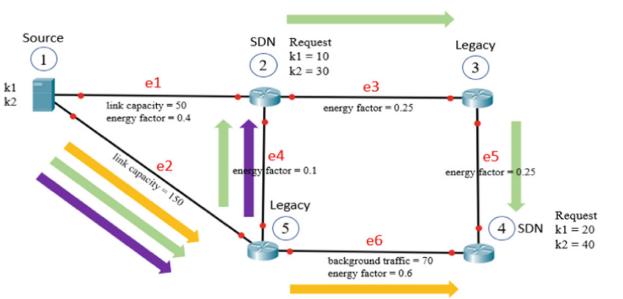
(a) Test Case 1



(b) Test Case 2



(c) Test Case 3



(d) Test Case 4

Fig. 3. Five-nodes Mesh Topology

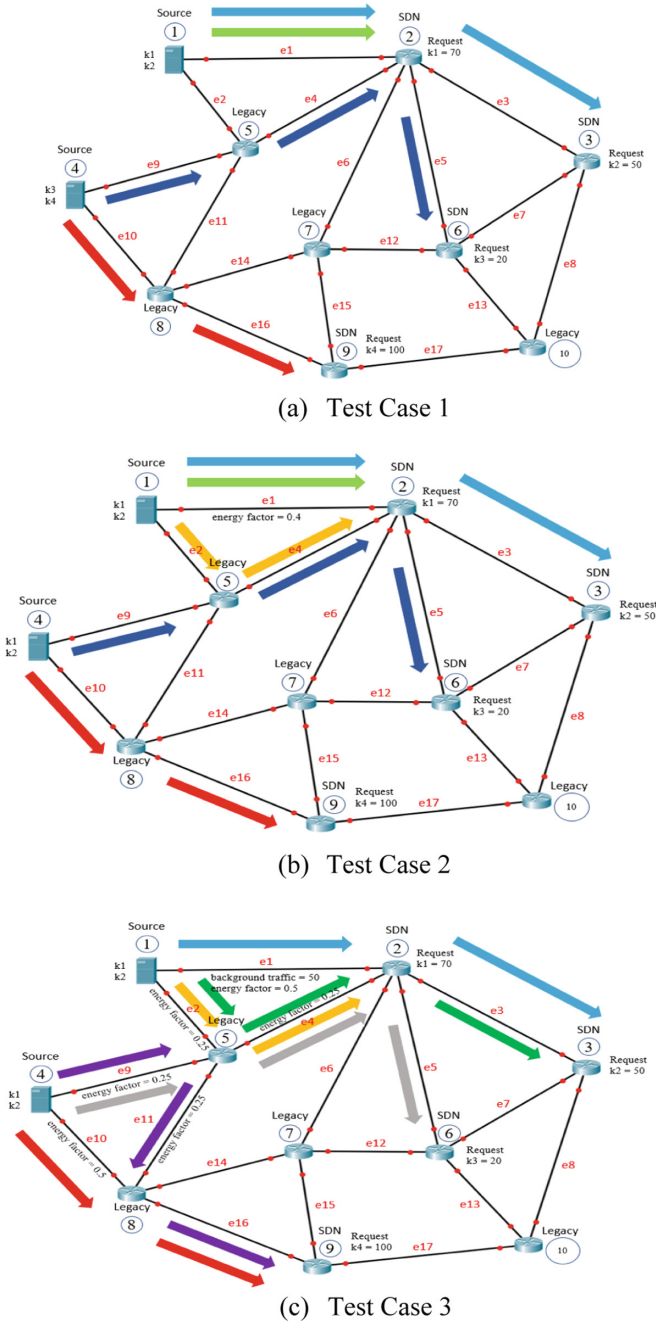


Fig. 4. 10-nodes Mesh Topology

topology in Fig. 4. Figures 3(a), 3(b), 3(c), and 3(d) illustrate the topology specification for each test scenario, as do Figs. 4(a), 4(b), and 4(d) (c).

To begin, we test the accuracy of our suggested algorithm by comparing its output to the ILP formulation. In Table 2, we show the minimum MLU and energy consumption varies to varies test cases 5-nodes mesh network with different traffic conditions. It is noteworthy to highlight that in the majority of circumstances; our suggested method produces the same results as the ILP formulation.

**Table 2.** Simulation Result of 5-Nodes Mesh Topology

	MLU		Energy		Computation Time (s)	
	AMPL	Algorithm	AMPL	Algorithm	AMPL	Algorithm
Test Case 1	0.6	0.61	32	32	0.015625	0.004724120
Test Case 2	1	1	58	58	0.015625	0.006216550
Test Case 3	0.66667	0.66667	54	54	0.0171875	0.005461840
Test Case 4	0.85	0.86	60	60	0.015625	0.005744500

In addition to soundness, our suggested Heuristic method has a major advantage over the ILP formulation in terms of time complexity. Time complexity is an important factor in computing because it influences the application of massive network computations such as an ISP network.

As illustrated in Table 2 and Table 3, we compare the average computational time over 10 simulations in 5-nodes mesh and 10-nodes mesh networks between ILP model and the proposed heuristic algorithm. We note that the suggested heuristic approach is substantially quicker than the ILP model across all test cases in both topologies. This raises the level of uncertainty in the ILP model for SDN traffic scheduling computation in big networks. The suggested heuristic approach, on the other hand, stays stable in all instances.

**Table 3.** Simulation Result of 10-Nodes Mesh Topology

	MLU		Energy		Computation Time (s)	
	AMPL	Algorithm	AMPL	Algorithm	Algorithm	AMPL
Test Case 1	0.6	0.61	86	86	0.0125	0.05344476
Test Case 2	0.5	0.55	110	110	0.0203125	0.05016853
Test Case 3	0.5	0.55	154	154	0.015625	0.05439731

## 5 Conclusion

We proposed an energy and congestion awareness traffic scheduling heuristic algorithm in hybrid SDN. In particular, our method takes into account critical parameters such as energy consumption and MLU while adopting dependable traffic scheduling in hybrid SDN. We also demonstrated the performance of our suggested heuristic algorithm vs the ILP model in terms of solution quality and processing time. With polynomial time complexity, the suggested heuristic method maintains its overall accuracy. This assures that our suggested technique is applicable in big networks such as an ISP network.

## References

1. Kaljic, E., Maric, A., Njemcevic, P., Hadzialic, M.: A survey on data plane flexibility and programmability in software-defined networking. *IEEE Access* **7**, 47804–47840 (2019)
2. Maity, I., Misra, S., Mandal, C.: SCOPE: cost-efficient QoS-aware switch and controller placement in hybrid SDN. *IEEE Syst. J.* **16**(3), 4873–4880 (2022)
3. Alsaedi, M., Mohamad, M.M., Al-Roubaiey, A.A.: Toward adaptive and scalable OpenFlow-SDN flow control: a survey. *IEEE Access* **7**, 107346–107379 (2019)
4. Bhardwaj, S., Girdhar, A.: Software-defined networking: a traffic engineering approach. In: 2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), pp. 1–5 (2021)
5. Josbert, N.N., Ping, W., Wei, M., Muthanna, M.S.A., Rafiq, A.: A framework for managing dynamic routing in industrial networks driven by software-defined networking technology. *IEEE Access* **9**, 74343–74359 (2021)
6. Guo, Y., Wang, Z., Yin, X., Shi, X., Wu, J.: Traffic engineering in hybrid SDN networks with multiple traffic matrices. *Comput. Netw.* **126**, 187–199 (2017)
7. Lin, C., Wang, K., Deng, G.: A QoS-aware routing in SDN hybrid networks. *Proc. Comput. Sci.* **110**, 242–249 (2017)
8. Tseng, S.H., Tang, A., Choudhury, G.L., Tse, S.: Routing stability in hybrid software-defined networks. *IEEE/ACM Trans. Netw.* **27**(2), 790–804 (2019)
9. Awad, M.K., Rafique, Y., M'Hallah, R.A.: Energy-aware routing for software-defined networks with discrete link rates: a benders decomposition-based heuristic approach. *Sustain. Comput.: Inform. Syst.* **13**, 31–41 (2017)

10. Fernandez-Fernandez, A., Cervello-Pastor, C., Ochoa-Aday, L.: Achieving energy efficiency: an energy-aware approach in SDN. In: 2016 IEEE Global Communications Conference (GLOBECOM) (2016)
11. Tao, J.: Research on SDN-based dynamic management of energy consumption of cloud data center. In: 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 1266–1270 (2021)
12. Naeem, F., Tariq, M., Poor, H.V.: SDN-enabled energy-efficient routing optimization framework for industrial internet of things. *IEEE Trans. Industr. Inf.* **17**(8), 5660–5667 (2021)
13. US 20170244606 A1: Systems and Methods for Bandwidth Management in Software Defined Networking controlled multi-layer Networks, 24 August 2017
14. CN109831386A: Optimal route selection algorithm based on machine learning under a kind of SDN, 31 May 2019
15. US 20160359735 A1: Using PCE as SDN controller, 8 Dec 2016
16. US 20170005910 A1: Open shortest path first routing for hybrid networks, 5 Jan 2017
17. US 20160218917 A1: SDN based inter-domain and intra-domain traffic engineering, 28 July 2016
18. US 20170006067 A1: Hybrid SDN/legacy policy enforcement, 5 Jan 2017
19. US 20190140951 A1: Hybrid routing table for routing network traffic, 9 May 2019
20. US 20140269260 A1: Distributed path update in hybrid networks, 18 September 2014
21. US 20130128738 A1: Hybrid networking path selection and load balancing, 23 May 2013
22. 61956164: Distributed path selection in hybrid networks, 4 December 2012
23. US 20120320919 A1: Automatic path selection for hybrid communication networks, 20 December 2012