Received March 14, 2019, accepted May 7, 2019, date of publication May 27, 2019, date of current version June 10, 2019.
Digital Object Identifier 10.1109/ACCESS.2019.2919348

Effective Resource Management in SDN Enabled Data Center Network Based on Traffic Demand

MANISH PALIWAL AND DEEPTI SHRIMANKAR
Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology, Nagpur 440010, India
Corresponding author: Manish Paliwal (paliwalmanish1@gmail.com)

ABSTRACT Software-defined networking (SDN) changes the way forwarding devices works and makes data center network simplified and efficient to work. Energy is the main driving force behind the activity of the data center network. The network elements that are in low load condition can be shifted in switch-OFF condition after migrating their active traffic on some other element. This traffic analysis-based energy-saving approach will increase the overall utilization of the system. This paper presents a switch-ON/OFF procedure along with the re-routing procedure to achieve the desired activity. The modules are tested over some well-known data center topologies, such as Fat-Tree and Bcube. The simulation study is carried out with the classical SDN controller instances, e.g., OpenDayLight, POX, Beacon, and NOX. The results are compared with some similar energy-efficient data center approaches, such as MNLR DCN and WDM Optical Switched DCN. The proposed approach presents an enhancement of 77%, 37%, and 63% in achieved energy saving, average round trip delay, and average bandwidth utilization, respectively.

INDEX TERMS Bandwidth utilization, data center, energy efficient network, round-trip delay, software defined networks.

I. INTRODUCTION

Recently in the networking field, a term Software Defined Networking (SDN) is introduced which is gaining popularity from various research communities and networking industries. The major reason behind this popularity is the simplified device and easy configuration of the network [1]. Earlier in a traditional network, the control plane (the logic behind forwarding functionality) and data plane (actual forwarding elements) are bundled together in a single box in a tightly coupled manner. This complex and tight integration makes the overall design complex. The network architecture is facing a problem with this integration as they are unable to perform necessary changes in the underlying data plane or customize a new functionality in it [2]–[4]. SDN overcomes this limitation of traditional network and provides complete control over the forwarding devices. The control logic is now detached from the forwarding devices and placed at a centralized location called a controller. The underlying forwarding devices are only left with the simplified data plane. Now, the network administrator can perform customization in the control plane of forwarding devices from the controller itself. They only need to place the activity module inside the controller and controller applies the appropriate functionality over underlying data plane devices [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Mubashir Husain Rehmani.

The data center [6] network is the backbone of any organization. It allows the organization to communicate with the outside world. In the data center, requests can arrive at any time. So, it becomes necessary for data center devices to run 24X7 continuously. A significant amount of energy is required to run them continuously. However, it has been seen that the number of requests, which arrives at the data center, are not the same all the time [7], [8]. Certain variation is there in the number of requests which depends upon the users who are active in the system [9]. The energy resources are very limited in quantity that may not be available in future [10], [11]. We can save a significant amount of energy by adjusting the power requirement of the data center

This paper presents an approach where the devices work according to the demand of traffic. During the peak hours, all the devices are in the switch ON condition and consume the highest energy from energy resource. On the other hand, in a non-operational hour, some of the lightly loaded devices are in switch OFF condition that saves a significant amount of energy. The Switch ON/OFF module ensures about this and put the lightly loaded devices in the switch OFF mode according to the traffic demand metrics statistic. During the switch OFF process, the current traffic of the corresponding device is moved to some alternate devices so that the overall functioning of the system is not disturbed. The re-routing module is responsible for this activity and does the necessary task. The module finds the alternate LSPs in the architecture so that the current traffic of the switched OFF node/link can be migrated on that without affecting the functionality of the system. Once the traffic crosses the threshold limit, the switch OFF devices need to bring in the switch ON condition again. This complete transition process is carried out according to the traffic demand matrix which is obtained by the real-time traffic analysis. We tested the proposed hybrid approach along with some similar energy efficient data center approaches such as MNLR (Multinomial Logit Model Routing) DCN, traditional DCN and WDM Optical Switched DCN. The study is carried out on the performance parameter e.g. achieved energy saving, average round trip delay and bandwidth utilization.

This approach is mainly intended for those data center networks where the traffic pattern changes according to operational hours. The individual organizations, institutions, the industries can get benefited from this idea. The applicability of MPLS technology in devices makes them effective as compared to traditional IP architecture. Now, with the inclusion of MPLS technique, the routing functionality of devices is relaxed as they don't need to perform complex IP header lookup. As compared to the previous resource management approaches in the data center, the hybrid SDN-MPLS architecture makes the overall data center simplified and efficient. It allows us to take advantage of both SDN control at the controller side and MPLS forwarding features at the physical data plane. Finally, the key contributions of this paper are as follows-

- Enforcement of hybrid SDN-MPLS architecture in the data center network.
- Energy saving in the data center network by switching off inactive elements.
- Dynamic provisioning of network elements based on traffic demand data in a real-time manner.

Rest of the paper is organized as follows: Section 1 discusses an introductory portion of SDN technology and data center mechanism. Section 2 gives the literature survey carried out in the SDN field along with its data center part and MPLS technique. In section 3, we provide details of the proposed architecture of the system. Section 4 deals with the methodology of the proposed approach. In section 5, simulation setup and Result are discussed. Finally, we present conclusions and future work in section 6.

II. LITERATURE SURVEY

SDN is a result of continuous effort and the activity, started around 1997 as Active networks [16]. Forwarding and control element separation (ForCES) [17], Ethane [18], Network Control Point [19] were the similar counterparts of SDN technology. These technologies suggest the separation of control and data plane in forwarding devices. However, they failed to address the clean separation image and only provide a platform for the SDN technique. Till 2008, the development of SDN was in its partial development stage. OpenFlow was one of the major milestones in SDN research which was introduced in 2008. It was a southbound protocol and works as a mediator between the controller and the underlying data plane. The first time, SDN was officially introduced by Martin Casado at Stanford University during his research program. After his successful deployment of SDN, researchers and industry experts got attracted towards SDN. Soon the majority of software giants like Cisco, Verizon, Microsoft, Google etc. started working on SDN and performed the transition of their network architecture to SDN. In 2011, these IT giants collaborated with each other for the opensource deployment of SDN and its related devices. They came up with a solution called as Open Networking Foundation (ONF) [20] which provides support for the opensource implementation of SDN technique. To promote the research in SDN, various simulating platforms like Mininet [21], GENI [22], VINI [23] etc. are available.

Hybrid SDN is an evolutionary technique in today's communication network. Salsano *et al.* [24] proposed a hybrid approach for packet forwarding which uses the IP routing along with the SDN forwarding control in the core network. They designed a Hybrid IP/SDN architecture known as Open Source Hybrid IP/SDN (OSHI). In [25], Y. Guo et al. tested the traffic engineering approach in SDN/OSPF hybrid network architecture. Through this approach, they suggested to modify the weights and flows of OSPF routing from the centralized SDN controller. Lopez *et al.* [26] presented various alternative solutions for the control plane. Specifically, they focused on the SDN usability in the multi-vendor carrier network. An incremental deployment model for Hybrid SDN networks was proposed in [27]. This model consists of both legacy as well as SDN switches. The model suggests an approach to upgrade the legacy switches into SDN compatible switches and interoperability among them to meet the various traffic engineering requirements e.g. performance, load balancing, failure recovery etc.

Hybrid SDN-MPLS is a hot topic in communication field covering both academic and industry research domain. Several approaches have been suggested to integrate the MPLS technology [28] along with the SDN. Various research activities are carried out on this Hybrid implementation approach to achieve significant effectiveness in communication. Sgambelluri *et al.* [29] worked on the two-segment routing (SR) approach for MPLS and SDN network. They considered two different networks in their network architecture. The first network was built from SDN components like OpenFlow switches and routers. The second network consists of MPLS nodes like PCE, MPLS Router etc. SR controller contains the control logic for both of the network. So, on the controller side, this implementation is hybrid but on data plane side it is individual. Casado *et al.* [30] proposed a new design

TABLE 1. Advancement, drawback and major contribution of previous DCN architecture.

DCN Strategy	Advancement	Drawback	Major Contribution
MNLR DCN [12]	Energy efficient routing algorithm based on multinomial logit model	The data plane works on traditional IP scenario hence increases the overall complexity of packet inspection.	Priority-based selection of a path in the multi-path environment.
WDM Optical Switched DCN [13]	All optical switching within entire DCN	Lack of dynamic traffic pattern consideration	DCM oriented and WDM enhanced optical circuit switching method, Investigate the per-network-layer application of optical switching
PINE [14]	Introduces the photonically interconnected data center Elements, Design of PINE silicon photonic physical layer using multi-component parameter optimization	Optical design of the data center leads to higher configuration cost demand	Disaggregated architecture which is useful in the application such as machine learning.
HOLST [15]	Time slot switching based optical data center network, 50% energy consumption reduction over the conventional architecture	Optical switches lacks of SDN technology, The data plane works in a traditional manner.	Improvement in optical circuit utilization using TDM and optical circuit multiplex function

paradigm for SDN controller to meet the effective switching capabilities. They suggested two different controllers. First, the edge controller that provides various network services to the core network. And second, fabric controller that prepares a separate control plane which deals with the packet forwarding function. Das *et al.* [31] presented a complete Hybrid scenario of MPLS/SDN architecture. They retained the standard MPLS data plane at the forwarding layer while the control functionality has been moved to the centralized controller using SDN functionality. The control logic is pushed as a flow entries in the MPLS data plane devices. They argue about its effectiveness as this cuts down the implantation and configuration complexity. They didn't address the performance aspects of the architecture. In [32], Hui et al. defined the design of HybNET which was an automated network management framework for hybrid network infrastructure. They provided best effort compatibility between legacy and SDN switches while retaining most of the advantages of SDN technology. In [33], Paliwal et al. reviewed the classical state of art SDN controller over performance matrices such as throughput, latency delay etc.

Energy conservation is one of the hot topics in the Data Center Network environment. In literature, there are various works carried out which intent to energy saving in the data center network. In [34], multipath TCP along with the segment routing is proposed for SDN based data center network which mainly addresses the storage issue of ternary content addressable memory(TCAM). In [35], workload-aware task scheduling has been suggested in the context of data center network which takes into account the routing latency as well as VM migration latency. In [36], SDN based virtual machine management scheme is suggested which aims to reduce the communication cost in the cloud data center. In [37], the author considered the issue of energy efficiency in the context of both control and data plane. They suggested elastic multi-controller architecture which dynamically adjusts the energy requirement in large scale data center. Table 1 discusses the advancements, drawbacks and major contribution of the previous approaches in DCN design.

III. PROPOSED ARCHITECTURE

The proposed architecture is designed on the hybrid SDN-MPLS concept [38]. As we know that the MPLS is an efficient technique to forward the packets in the core network, this core network packet forwarding approach can be enhanced by applying the concept of SDN. The control functionality of core network components can be managed from the centralized controller. The MPLS data forwarding capability along with the SDN orchestration capability can give better results regarding throughput and efficiency.

Fig. 1 represents the architectural overview of the proposed system. The core network is made of hybrid routers which comprise both SDN and MPLS functionality. The edge forwarding devices are also equipped with the SDN functionalities. These edge forwarding devices are the entry and exit points to the core MPLS network. The control functions of all the forwarding devices are taken into a centralized location which is called as a controller. The controller posses certain modules e.g. Topology Manager, Route Discovery, Device Manager etc. which monitors the functioning of forwarding devices and direct them to forward the packets. The edge forwarding devices are further connected to the corresponding subnet switches and hosts. The Network Operating System [39] running inside the controller provides a set of utilities for effective communication between the controller and underlying forwarding devices.

OpenFlow [40] operates as an intermediate protocol between controller and forwarding devices. This protocol is responsible for different flow table operation from the controller to forwarding devices and vice versa. Similarly, the Network Administrator communicates with the controller via northbound API. The northbound interface is supported by the REST APIs [41].

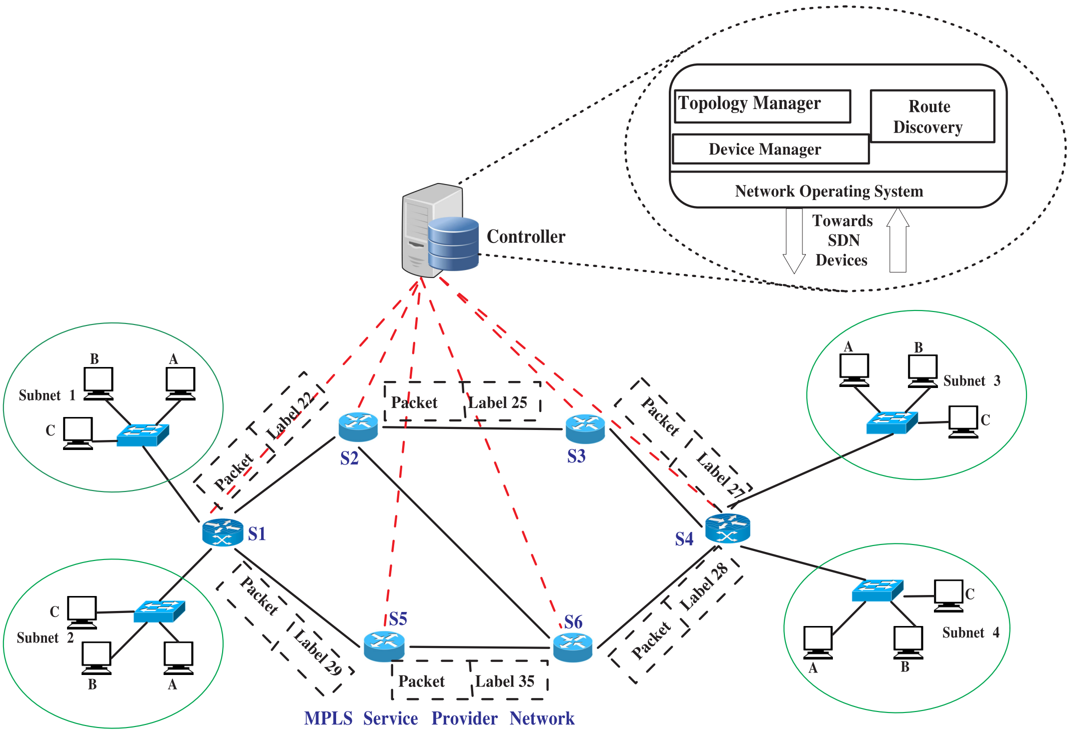

FIGURE 1. Architectural overview of proposed system.

The packet size has been taken as 64 bytes which is a standard packet size format in the IP network. The MPLS labels, which are 4 bytes in size, are attached with each of the packets. The labels are ranging from 16 through 1,048,575 ($2^{20} - 1$) which are the unreserved labels for MPLS network.

IV. PROPOSED METHODOLOGY

The proposed approach searches for the traffic pattern of data center and makes statistics for the different time intervals. These statistics are carried out on nodes as well as links to the network. We collectively call these as network elements. The resultant data stored in the matrix form is called a traffic demand matrix (TDM) [42]. The Network elements are ranked according to their traffic and their information is stored in priority list like data structure. Once this information is ready, we pick the highest priority element from the list and apply the switch ON/OFF procedure on it. The network element, having the lowest traffic from itself, has the highest priority among all. The network element is served in a step by step manner. The highest priority node is selected first among the all available nodes. For this node, all the connected links are served one by one. It has to be noticed that the links are again served on the basis of priority starting from highest to lowest. Each link and node has assigned a predefined threshold limit. If the traffic goes down or up from that limit, the network elements will be switched OFF and ON accordingly.

During the switch OFF and ON procedure, it is necessary to migrate the current traffic from network elements to some other alternate elements. We are dealing in multi-path network architecture where from a given source to a destination some alternate paths are available. These alternate paths are discovered during this process and stored in the data structure to fetch out further. This process is continued until the system is in a stable state with minimum functioning resources. Periodic checking for the network traffic and update to TDM is necessary for effective functioning of the system.

A. SWITCH ON/OFF MODULE

The Switch ON/OFF module takes care of selecting an appropriate network element for ON/OFF. During choosing network elements, this module ensures that the network performance should be maintained at the appropriate level. The quality of service (QoS) constraints are defined by the overprovisioning criteria. Ultimately, our final goal is to reduce the energy consumption of the network and put the attention on the small subset of the network.

The steps which have to be carried out in the Switch ON/OFF modules are discussed below.

1) **TDM Calculation & Overprovisioning factor determination -** To predict the traffic pattern in the different time domain, we need to prepare a Traffic Demand Matrix (TDM) which is a data structure containing information related to each network element. We can use CISCO Wan Automation Engine [43] or Wireshark [44] to calculate the TDM. We can accurately predict the network information e.g. changes in topology, overall utilization before and after failure etc. Once we are done with the TDM calculation then we proceed to determine the overprovisioning factor. The overprovisioning factor defines the bandwidth amount which is to be overprovisioned so that the network

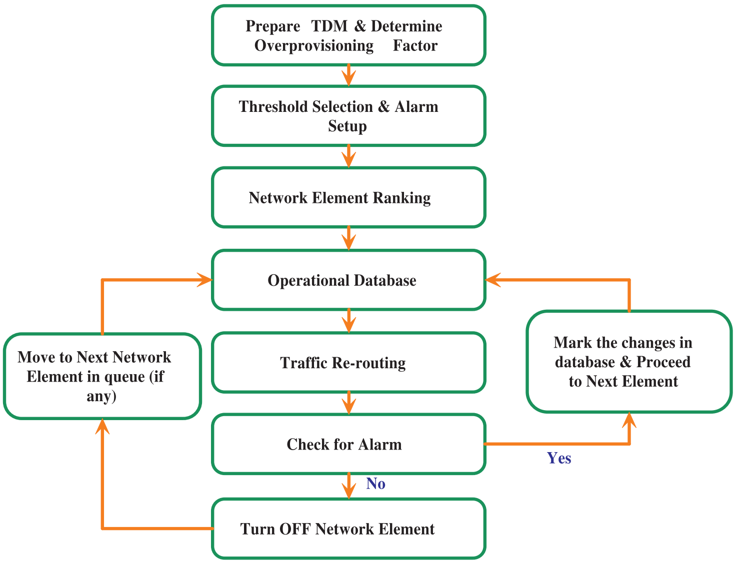

FIGURE 2. Flow diagram of Switch ON/OFF procedure.

administrator can meet the service level agreement (SLA) requirements. SLA requirements are based upon the matrices e.g. delay, jitter etc. These factors can be manually defined by the network administrator or can be automatically defined. The different time domain, on which traffic pattern varies, are calculated based on this information. These time domains are helpful to identify the least and most busy operational hours in the organization.

2) **Threshold Selection & Alarm Setup** – On the basis of TDM and overprovisioning factor calculated in the previous step, we set up the threshold for the bandwidth amount. For a link/node, if the bandwidth amount is below the given threshold then we generate an alarm that tells us to switch OFF the network element. This threshold checking should be done periodically so that non-operational elements can be turned ON during the traffic increase.
3) **Network element Ranking** – For effective operation of the system, it is necessary to switch ON/OFF network element in a specific order. To get this specific order, we rank the network elements based on various factors associated with them. For Network links, we identify the factors e.g. throughput, average utilization etc. Similarly, for network nodes, we identify the factors e.g. CPU utilization, Memory utilization etc.
4) **Database Creation** – We create a database instance to keep track of information regarding the priority of network elements determined in the previous steps. For each time domain, we create two database instance. The first instance is called the primary database instance. This instance initially saves the information for each time domain. The second instance is called a Secondary instance. At the start, a secondary instance consists of the same copy as a primary instance. During the operation of switch ON/OFF and re-routing procedure, the periodic refresh operation is carried out on the secondary instance. On the other hand, the primary instance store the priority information of nodes/links so that they get processed in proper order. The node/link having the highest priority is served first. To define the node/link priority, we use the traffic amount information. The node/link which has the least traffic amount passing through it represents the highest priority element.
5) **Traffic Re-routing** – During switch ON/OFF operation, it is necessary to migrate the traffic of the node/link to some alternate node/link. We use the MPLS traffic engineering feature for this operation. RSVP-TE is a well-known signaling protocol for MPLS-TE. Using RSVP-TE, we find the alternate LSPs so that the traffic of the current node which is going to be switched OFF can be migrated successfully. This alternate path selection ensures that the switched OFF node should not be included in the list.
6) **Alarm Check** – After the alternate path has been identified by constraint shortest path first (CSPF) algorithm, the traffic re-route from that node. During this process, we check for any alarm which ensures that the overall bandwidth utilization is maintained after re-routing. If at any point we find a reduction in overall bandwidth utilization of the system then we set the previous path for the traffic.
7) **Switch ON/OFF node** – Finally, when all the previous steps are compiled successfully, we switch OFF the node and continue the process for remaining nodes in the queue.

Fig. 2 represents the flow diagram of the switch ON/OFF procedure and illustrates the idea behind it. Algorithm 1 denotes the overall switch ON/ OFF procedure. To stay in switch ON condition, the procedure assumes the minimum threshold for any network element as 20.

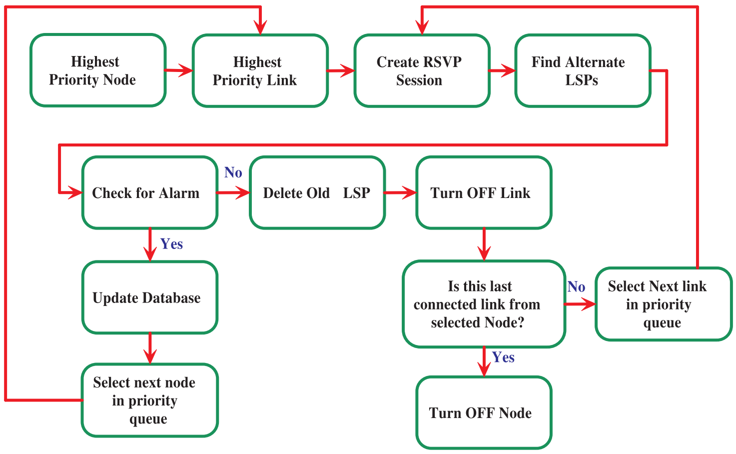

FIGURE 3. Flow diagram of re-routing procedure.

Algorithm 1 Switch ON/OFF Procedure

1: **procedure** Switch(TDM, OF)
2: $TH_{min} = 20$;
3: *Create* $DB(N_i, L_i)$
4: $callRR()$;
5: *if* $TH_{HPN_j} \leq TH_{min}$ *then*
6: *shut down* HPN_j;
7: *update* DB;
8: *goto step* 4 *& repeat procedure*;
9: **end procedure**

Algorithm 2 Re-Routing Procedure

1: **procedure** RR(N, L, $TF[N]$, $TF[L]$)
2: $HPN = N_1$;
3: **for** i=N_2 to N_n
4: *if* $TF[i] < TF[HPN]$ *then*
5: $HPN = i$;
6: **end for**
7: $HPL = HPN_{L_1}$;
8: **for** $j = L_2$ to L_n
9: *if* $TF[j] < TF[L_1]$ *then*
10: $HPL = HPN_j$;
11: **end for**
12: *return* HPL;
13: *call* $RSVP()$;
14: *if* $HPN_j == HPN_{ll}$ *then*
15: *break*;
16: *else*
17: *goto step* 8 *& repeat procedure on remaining links*
18: *if* $HPN == HPN_{ln}$ *then*
19: *break*;
20: *else*
21: *goto step* 3 *& repeat procedure on remaining nodes*
22: **end procedure**

B. TRAFFIC RE-ROUTING MODULE

This traffic Re-routing module ensures that the current traffic of the node (being switched OFF) must be transferred to alternate label switch paths. Following steps are performed during the re-routing process.

1) **Highest Priority Node Selection** – Based on priority queue obtained from TDM, we select highest priority node. A node having the least traffic among all is considered as highest priority node. The network element ranking module provides necessary information for the highest priority node selection.
2) **Highest Priority Link Selection** – Once highest priority node is identified, the next step is to select the highest priority link associated with this node. For a node, the number of active interfaces define the links. This priority information is taken from the priority queue discussed in previous stages.
3) **RSVP session creation & Alternate LSP selection** – After the selection of highest priority node and link, the next step is to create a Resource reservation protocol (RSVP) session for them. During this session, we find the alternate paths which can be a replacement for the given path. The alternate label switched paths (LSPs) are stored in a database so that they can be monitored in subsequent states.
4) **Alarm Check** – An Alarm test is performed to check the consistency of migration. In this step, we ensure that our migration procedure should not disturb the bandwidth utilization of the system and should be free from packet loss.
5) **Last connected link test** – In this step, we check whether the link, which is going to be switched OFF, is the last connected link to the node. If this is the case, then the procedure will continue with the next highest priority node in the queue. On the other hand, the failure will lead to the continuation of this procedure with the next available links.
6) **Turn OFF node & End Task** – The procedure will conclude when all the nodes and their corresponding links are satisfied along with the service level agreement of the architecture. If no further element switch OFFs are required, then we exit from the procedure.

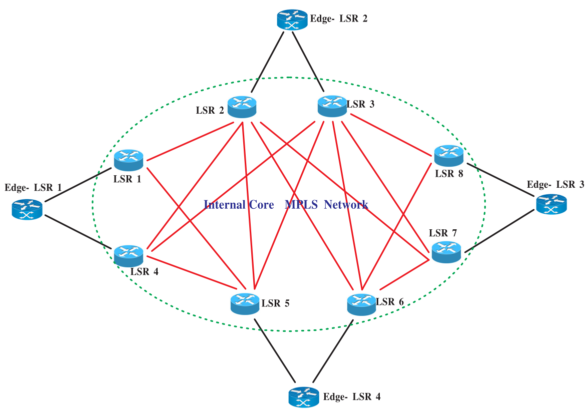

FIGURE 4. Topological overview of Network.

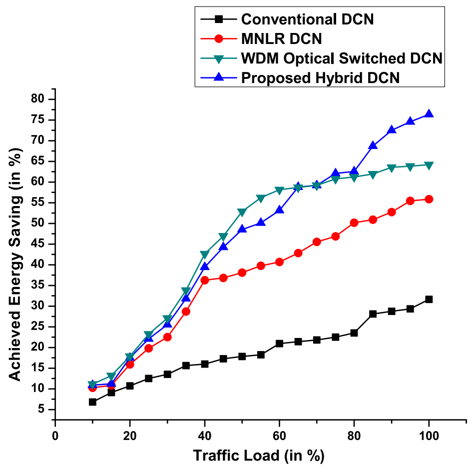

FIGURE 5. Energy saving v/s Traffic Load in Fat-Tree DCN.

Fig. 3 represents the flow diagram of the re-routing procedure and illustrates the idea behind it. Algorithm 2 denotes the re-routing procedure.

V. SIMULATION SETUP AND RESULTS

The proposed network architecture is simulated over Network Simulator-3 [45], [46] to test the effectiveness of the system. The sample topology in fig. 4 represents the topological view of the telecommunication service provider. The network traffic has been captured over Wireshark [44].

The topology is divided into two regions. The inner region consists of the forwarding devices which prepare the core of the network. The outer region consists of the devices which perform the edge functionality of the network. The number of devices in the inner region are more than the outer region. The inner region along with the outer region prepares the complete core network of the service provider. We took the inner core network for resource overprovisioning because of two reasons. First, the core region contains many alternative LSPs so there is a chance of high resource overprovision. Second, a large amount of energy is consumed by the core network so we can save a significant amount of energy by shutting down some forwarding devices. The core network is organized in full mesh architecture. On the other hand, the edge forwarding devices are connected to at least one core forwarding device. We can consider partial Mesh topology

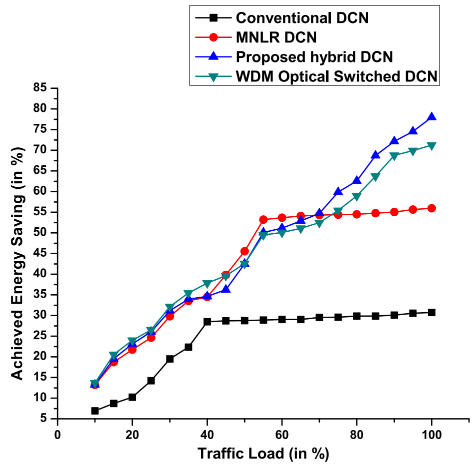

FIGURE 6. Energy saving v/s Traffic Load in BCube DCN.

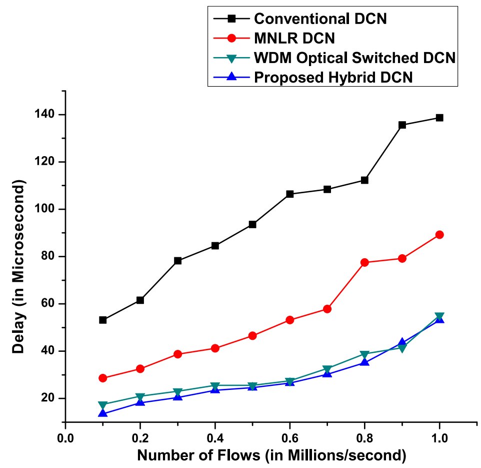

FIGURE 7. Average Round Trip Delay v/s number of flows in Fat-Tree DCN.

among the core edge forwarding devices. Table 2 discusses the simulation parameters and their values which are used in the simulation.

Fig. 5 shows the graphical representation of achieved energy saving through proposed hybrid DCN architecture over the other DCN approaches in Fat-Tree DCN topology. In full load condition, the proposed hybrid DCN provides 76.35% energy saving which is 12% greater than the best comparative approach. Similar, in fig. 6, we present the achieved energy saving of proposed hybrid DCN in BCube DCN topology environment. The result shows around 78% energy saving in full load condition. On observing both scenarios, near about 77% energy saving is achieved by the proposed hybrid DCN architecture. The results also state that on the high load condition the proposed approach performs better than other comparative approaches.

The delay experienced by the packet during the journey from one subnet to another subnet in Fat-Tree and BCube DCN architecture is represented in fig. 7 and 8 respectively. In the Fat-Tree DCN environment, the average delay is calculated as 28.881 μs which is comparatively 50-80% lower than the other approaches. Similarly, in the BCube DCN environment, the average delay is observed as 37.014 μs

TABLE 2. Simulation parameters and their values.

Parameter Name	Notation	Value
Total Number of Flows in the DCN	S	10000
Fixed power consumption by the SDN switch	P_{fix} (W)	45
Port power consumption for the SDN switch	P_{port} (W)	3
Maximum link capacity	C	1Gbps
BCube DCN topology level	L	2
Number of ports per SDN switch in BCube	BP	4
Number of ports per SDN switch in Fat-Tree	FP	8

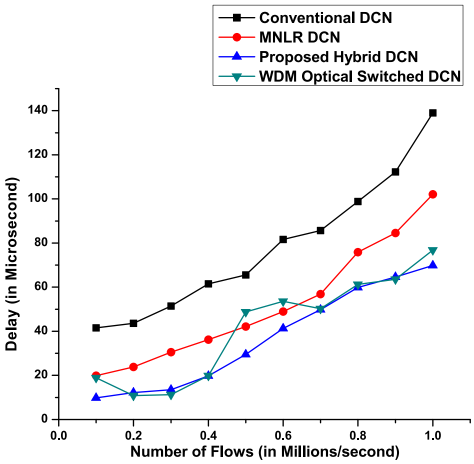

FIGURE 8. Average Round Trip Delay v/s number of flows in BCube DCN.

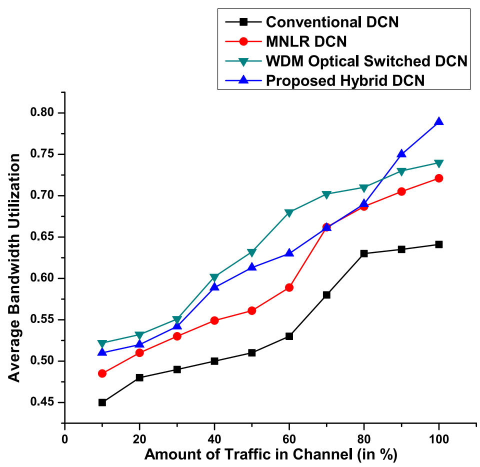

FIGURE 9. Bandwidth Utilization v/s Traffic Load in Fat-Tree DCN.

which is 35-57% lower than the comparative strategies. The graph restricts the number of flows in the network up to 1 million/second because of the limitations of many SDN routers. So taking into account both of the topological scenarios, the proposed hybrid DCN architecture presents average delay in the range of 25-40 μs.

Fig. 9 and 10 represents the average bandwidth utilization achieved through the proposed method in Fat-Tree and BCube DCN environment. The average bandwidth utilization in Fat-Tree DCN is 63% which is nearly 10 to 15% greater than the previous DCN architectures. A similar case is observed in BCube DCN environment where average bandwidth utilization is 59% and it is about 9-13% greater than the other comparative approaches. On combining both the results, the average bandwidth utilization is ranging from 59 to 63%.

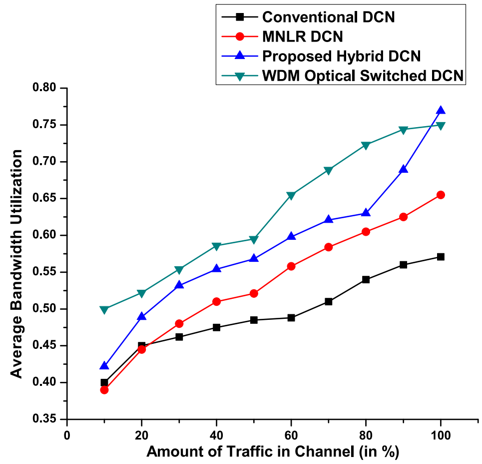

FIGURE 10. Bandwidth Utilization v/s Traffic Load in BCube DCN.

VI. CONCLUSIONS

In this paper, an energy efficient routing approach for data center network is proposed. It suggests switching OFF certain network elements (links/devices) during low traffic demand to save the high energy demand of DCN. The method suggests a re-routing procedure where we can transfer the current traffic of switched OFF element to some alternate running element. The method works on the MPLS based DCN where the core forwarding devices are taken for switching OFF process. By observing the simulation results which are carried out on Fat-Tree and BCube DCN, we expect around 77% reduction in the overall energy consumption of the network. The architecture reduces the average round trip delay by 37% and enhanced the bandwidth utilization by 63%. The re-routing module ensures that no active traffic loss should happen during switching OFF process. The proposed architecture can meet the growing traffic demands of the user in the organization. It also provides a way to optimize the network element utilization.

REFERENCES

[1] J. Qadir, N. Ahad, E. Mushtaq, and M. Bilal, "SDNs, clouds, and big data: New opportunities," in *Proc. 12th Int. Conf. Frontiers Inf. Technol. (FIT)*, 2014, pp. 28–33.

[2] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges," *IEEE Commun. Surveys Tuts.*, vol. 8, no. 1, pp. 24–37, 1st Quart., 2006.

[3] K. Fall, "A delay-tolerant network architecture for challenged Internets," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2003, pp. 27–34.

[4] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[5] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[6] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.
[7] C.-C. Chuang, Y.-J. Yu, and A.-C. Pang, "Flow-aware routing and forwarding for SDN scalability in wireless data centers," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1676–1691, Dec. 2018.
[8] T. Hafeez, N. Ahmed, B. Ahmed, and A. W. Malik, "Detection and mitigation of congestion in SDN enabled data center networks: A survey," *IEEE Access*, vol. 6, pp. 1730–1740, 2018.
[9] G. Sun, H. Yu, V. Anand, L. Li, and H. Di, "Optimal provisioning for virtual network request in cloud-based data centers," *Photonic Netw. Commun.*, vol. 24, no. 2, pp. 118–131, 2012.
[10] S. Bahrami, V. W. S. Wong, and J. Huang, "Data center demand response in deregulated electricity markets," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2820–2832, May 2018.
[11] J. G. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, Oakland, CA, USA, Tech. Rep., 2011, vol. 9.
[12] M. Wei, J. Zhou, and Y. Gao, "Energy efficient routing algorithm of software defined data center network," in *Proc. IEEE 9th Int. Conf. Commun. Softw. Netw. (ICCSN)*, May 2017, pp. 171–176.
[13] A. Pilimon, A. Zeimpeki, A. M. Fagertun, and S. Ruepp, "Energy efficiency benefits of introducing optical switching in data center networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, 2017, pp. 891–895.
[14] K. Bergman, J. Shalf, G. Michelogiannakis, S. Rumley, L. Dennison, and M. Ghobadi, "PINE: An energy efficient flexibly interconnected photonic data center architecture for extreme scalability," in *Proc. IEEE Opt. Interconnects Conf. (OI)*, Jun. 2018, pp. 25–26.
[15] M. Hirono, T. Sato, J. Matsumoto, S. Okamoto, and N. Yamanaka, "HOLST: Architecture design of energy-efficient data center network based on ultra high-speed optical switch," in *Proc. IEEE Int. Symp. Local Metrop. Area Netw. (LANMAN)*, Jun. 2017, pp. 1–6.
[16] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A survey of active network research," *IEEE Commun. Mag.*, vol. 35, no. 1, pp. 80–86, Jan. 1997.
[17] A. Doria *et al.*, "Forwarding and control element separation (ForCES) protocol specification," IETF, Fremont, CA, USA, Tech. Rep. RFC 5810, 2010.
[18] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1–12, 2007.
[19] D. Sheinbein and R. P. Weber, "Stored program controlled network: 800 service using SPC network capability," *Bell Syst. Tech. J.*, vol. 61, no. 7, pp. 1737–1744, Sep. 1982.
[20] *Open Networking Foundation*. Accessed: Feb. 10, 2018. [Online]. Available: https://www.opennetworking.org/
[21] *Mininet*. Accessed: Apr. 10, 2018. [Online]. Available: https://www.mininet.org/
[22] L. Peterson, T. Anderson, D. Blumenthal, D. Casey, D. Clark, D. Estrin, J. Evans, D. Raychaudhuri, M. Reiter, J. Rexford, S. Shenker, and J. Wroclawski, "Geni design principles," *Computer*, vol. 39, no. 9, pp. 102–105, 2006.
[23] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In VINI veritas: Realistic and controlled network experimentation," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 3–14, 2006.
[24] S. Salsano, P. L. Ventre, L. Prete, G. Siracusano, M. Gerola, and E. Salvadori, "OSHI—Open source hybrid IP/SDN networking (and its emulation on mininet and on distributed SDN testbeds)," in *Proc. 3rd Eur. Workshop Softw. Defined Netw. (EWSDN)*, 2014, pp. 13–18.
[25] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in SDN/OSPF hybrid network," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols (ICNP)*, Oct. 2014, pp. 563–568.
[26] V. Lopez, L. M. Contreras, O. G. de Dios, and J. P. F. Palacios, "Towards a transport SDN for carriers networks: An evolutionary perspective," in *Proc. 21st Eur. Conf. Netw. Opt. Commun. (NOC)*, 2016, pp. 52–57.
[27] D. K. Hong, Y. Ma, S. Banerjee, and Z. M. Mao, "Incremental deployment of SDN in hybrid enterprise and ISP networks," in *Proc. Symp. SDN Res.*, 2016, Art. no. 1.
[28] E. Mannie, "Generalized multi-protocol label switching (GMPLS) architecture," IETF, Fremont, CA, USA, Tech. Rep. RFC 3945, 2004.
[29] A. Sgambelluri, F. Paolucci, A. Giorgetti, F. Cugini, and P. Castoldi, "SDN and PCE implementations for segment routing," in *Proc. 20th Eur. Conf. Netw. Opt. Commun. (NOC)*, 2015, pp. 1–4.
[30] M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian, "Fabric: A retrospective on evolving SDN," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 85–90.
[31] S. Das, A. R. Sharafat, G. Parulkar, and N. McKeown, "MPLS with a simple OPEN control plane," in *Proc. Opt. Fiber Commun. Conf./Nat. Fiber Optic Eng. Conf. (OFC/NFOEC)*, 2011, pp. 1–3.
[32] H. Lu, N. Arora, H. Zhang, C. Lumezanu, J. Rhee, and G. Jiang, "HybNET: Network manager for a hybrid network infrastructure," in *Proc. Ind. Track 13th ACM/IFIP/USENIX Int. Middleware Conf.*, 2013, Art. no. 6.
[33] M. Paliwal, D. Shrimankar, and O. Tembhurne, "Controllers in SDN: A review report," *IEEE Access*, vol. 6, pp. 36256–36270, 2018.
[34] J. Pang, G. Xu, and X. Fu, "SDN-based data center networking with collaboration of multipath TCP and segment routing," *IEEE Access*, vol. 5, pp. 9764–9773, Jun. 2017.
[35] H. Yuan, J. Bi, M. Zhou, and K. Sedraoui, "WARM: Workload-aware multi-application task scheduling for revenue maximization in SDN-based cloud data center," *IEEE Access*, vol. 6, pp. 645–657, 2017.
[36] R. Cziva, S. Jouët, D. Stapleton, F. P. Tso, and D. P. Pezaros, "SDN-based virtual machine management for cloud data centers," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 212–225, Jun. 2016.
[37] K. Xie, X. Huang, S. Hao, M. Ma, P. Zhang, and D. Hu, "E^3MC: Improving energy efficiency via elastic multi-controller SDN in data center networks," *IEEE Access*, vol. 4, pp. 6780–6791, 2016.
[38] S. Vissicchio, L. Vanbever, and O. Bonaventure, "Opportunities and research challenges of hybrid software defined networks," *ACM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 70–75, Apr. 2014.
[39] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 1–6.
[40] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.
[41] W. Zhou, L. Li, M. Luo, and W. Chou, "REST API design patterns for SDN northbound API," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, 2014, pp. 358–365.
[42] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 161–174, 2002.
[43] *Cisco Wan Automation Engine*. Accessed: Mar. 17, 2018. [Online]. Available: https://www.cisco.com/c/en/us/products/routers/wan-automation-engine/index.html
[44] *Wireshark*. Accessed: Apr. 9,2018. [Online]. Available: https://www.wireshark.org/
[45] *Network Simulator-3*. Accessed: Apr. 12, 2018. [Online]. Available: https://www.nsnam.org/
[46] G. F. Riley and T. R. Henderson, "The *ns*-3 network simulator," in *Modeling and Tools for Network Simulation*. Berlin, Germany: Springer, 2010, pp. 15–34.

MANISH PALIWAL received the B.E. degree in computer science and engineering from UIT-RGPV, Bhopal, in 2012, and the M.Tech. degree from the Government College of Engineering Amravati, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Visvesvaraya National Institute of Technology, Nagpur, India.

DEEPTI SHRIMANKAR received the B.Tech. degree in computer technology and the M.Tech. degree in image processing from Nagpur University, in 1997 and 2007, respectively, and the Ph.D. degree in parallel computing from the Visvesvaraya National Institute of Technology, Nagpur, India, where she is currently an Assistant Professor with the Department of Computer Science and Engineering.