

Packet Optimization of Software Defined Network Using Lion Optimization

Jagmeet Kaur¹, Shakeel Ahmed², Yogesh Kumar³, A. Alaboudi⁴, N. Z. Jhanjhi⁵ and Muhammad Fazal Ijaz^{6,*}

¹Department of Computer Science & Engineering, Chandigarh University, Mohali, Punjab, 140413, India

²Department of Computer Science, College of Computer Sciences and Information Technology, King Faisal University, Al-Ahsa, 31982, Saudi Arabia

³Department of Computer Science & Engineering, Chandigarh Group of Colleges, Mohali, Punjab, 140307, India

⁴College of Computer Science, Shaqra University, Saudi Arabia

⁵School of Computer Science and Engineering, SCE, Taylor's University, Subang Jaya, 47500 Malaysia

⁶Department of Intelligent Mechatronics Engineering, Sejong University, Seoul, 05006, Korea

*Corresponding Author: Muhammad Fazal Ijaz. Email: fazal@sejong.ac.kr

Received: 31 January 2021; Accepted: 13 April 2021

Abstract: There has been an explosion of cloud services as organizations take advantage of their continuity, predictability, as well as quality of service and it raises the concern about latency, energy-efficiency, and security. This increase in demand requires new configurations of networks, products, and service operators. For this purpose, the software-defined network is an efficient technology that enables to support the future network functions along with the intelligent applications and packet optimization. This work analyzes the offline cloud scenario in which machines are efficiently deployed and scheduled for user processing requests. Performance is evaluated in terms of reducing bandwidth, task execution times and latencies, and increasing throughput. A minimum execution time algorithm is used to compute the completion time of all the available resources which are allocated to the virtual machine and lion optimization algorithm is applied to packets in a cloud environment. The proposed work is shown to improve the throughput and latency rate.

Keywords: Software-defined network; cloud computing; packet optimization; energy efficiency; lion optimization; minimum execution time

1 Introduction

The best-known network-related problems can be solved using software-defined networks (SDNs) [1]. The SDN has a straightforward, simple architecture. The combination of applications and a network operating system (NOX) make up an SDN control plane that runs on top and manages the network [2]. Network elements such as routers and switches make up the forwarding plane. An OpenFlow protocol enables communication between the two planes [3] so that administrators and network engineers can quickly respond to requirements. Cloud computing is widely accepted, and is built around core concepts such as on-demand, pay-for-use computing resources, reduced operational expenses, elastic scaling, and up-front investment elimination. Cloud



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

computing technology is still evolving, and offers service under different models, such as network as a service (NaaS), software as a Service (SaaS), infrastructure as a service (IaaS), and platform as a service (PaaS) [4].

Cloud computing is pervasive and ubiquitous. Visualized applications are executed in the private or public cloud by one mobile user, and devices can be accessed throughout the world. Visualization is a core technology that dramatically increases the importance of network infrastructure. In distributed fashion, a combination of routers, switches, and other devices provides scale and reliability. However, in many scenarios, the network architecture cannot be used, which makes it difficult to satisfy requisites such as end-to-end quality of service load balancing and usage of VLANs [5]. To allocate new applications, end-to-end services becomes complex in cloud computing. The burden on management increases, and data center performance declines by the difference shown by the inconsistency of the management interface of network and heterogeneous physical machines.

SDN is an emerging network architecture that decouples network forwarding and control of functionality, as shown in Fig. 1. Control of network service and applications is migrated from individual equipment to the network [6]. Cloud computing becomes a common solution for application deployment due to its high capacity to leverage developers from software management tasks, which reduces the cost of services and time to market. The cloud has a global footprint, with increased bandwidth consumption, packet optimization, and communication latency, enabling users to realize various advantages.

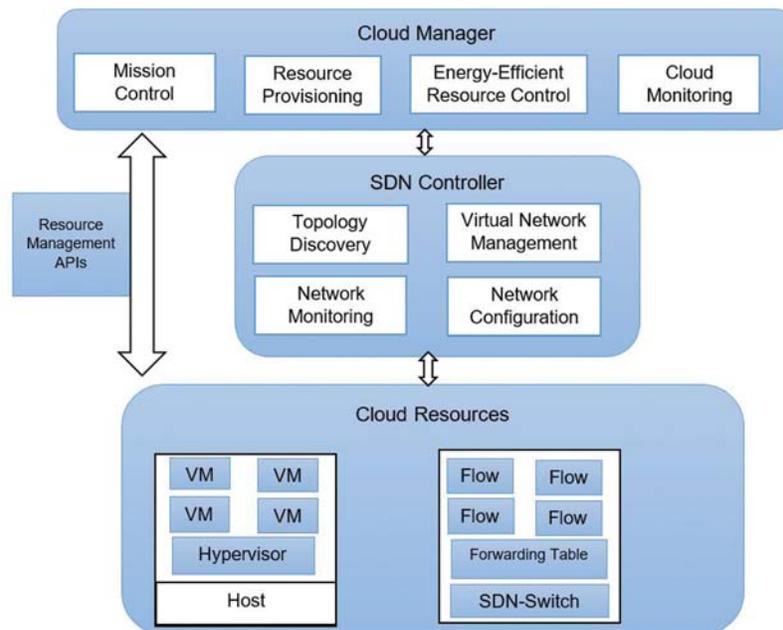


Figure 1: SDN-cloud system architecture [6]

The introduction of SDN in cloud computing improves complexity and packet optimization issues and enables the hosting of millions of virtual networks without VLAN-like common separation isolation methods [7]. Network administrators can manage network services from a central management tool, and network connectivity is virtualized in a logical network. Other common

advantages are cost reduction, intelligent global connections, granular security through a central point of control to distribute policy and security information, reduced downtime, and optimal packet usage for data transfer [8]. The provision of new end-to-end services and applications in the cloud environment becomes complex. The burden on management increases, and data center performance declines. This work studies issues of packet optimization in the SAAS module with SDN. Based on this, an approach is proposed to reduce bandwidth consumption and achieve energy efficiency using SDN. A new routing table code is introduced by optimizing the source code of the OpenFlow controller, making the backend environment more manageable.

2 Background Study

We provide a brief overview of the SDN-cloud network and discuss applications, objectives, and issues that motivate our work.

2.1 SDN-Based Cloud Network and Applications

SDN is a programmable network architecture that decouples network control and forwarding functionality [9]. Logically centralized and accessible computing devices enable the abstraction of infrastructure. An access network and comprehensive view of resources is provided by a centralized SDN control plane [10]. It has the following building blocks.

- Cloud backbone enabled edge nodes connect the cloud provider and enterprise.
- Core nodes switch traffic between edge nodes [11].
- An SDN-based controller enables configuration of forwarding tables in cloud backbone nodes and provides WAN network virtualization.
- Hybrid orchestration and cloud operation software manages provider and enterprise data center federation, resource management, and workflow, and facilitates multi-vendor networks between data centers, service providers, and users [12].

Customers can choose the best vendors and avoid vendor lock-in. DWDM, a PON-like technology, can be accessed with dynamic bandwidth. Timely migration and processing of inter-data center workload is possible. On-demand bandwidth is automated, and customer requirements are derived from intelligent service provisioning. Fig. 2 shows SDN applications used to solve problems of routing, QoS, network management, and network performance-related problems.

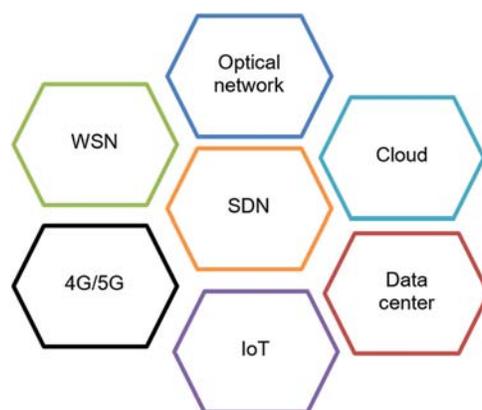


Figure 2: Various networks use SDN [12]

SDN can be used in data centers, the internet of things, cellular networks, and cloud computing to solve network-related issues.

2.2 Packet Optimization in the Cloud with SDN

Various issues are associated with packet optimization in the cloud with SDN.

- Current cloud networking architectures rely on the paradigm of one size fits all.
- Application bandwidth requirements should be specified by cloud tenants for consistent performance.
- Guaranteed bandwidth is required for various tiered applications [12].
- A wider variety of security appliances is needed, along with applications to accelerate tasks, cache, and balance loads.
- Access control and traffic isolation for end-users impact switch and router configurations [13].
- Changes in requirements and various protocols can bring challenges to the operation and building of a cloud network [14,15]. To match traffic requirements, a data center network topology is tuned, and depends on L2 and L3 utilization for effective network capacity.
- For network-dependent failover mechanisms and IP addresses, out-of-box applications can necessitate reconfiguration and rewriting before deployment of the cloud.

3 Reported Work

Much work has been done in the fields of SDN and cloud computing. [Tab. 1](#) presents this work in the categories of optimization in the cloud computing environment, in SDN, and in the cloud with SDN.

Table 1: Work in optimization in cloud computing, SDN, and their combination

Author	Approach	Purpose	Outcome
Optimization in Cloud Computing			
Rocha et al. [16]	Hybrid optimization model	Establishment of VM placement strategy for data centers	Balance between energy efficiency, network quality, and infrastructure
Liu et al. [17]	M/Geo/1 and M/Geo/m models based on queuing theory with embedded Markov chain and Z-transform	Optimize resource provisioning cost while satisfying customer QoS requirements	The proposed model can set up an affordable cloud firewall
Raad et al. [18]	Adaptive control framework	Improve access of distributed data center fabric delivered cloud services	Reduce transfer time by 80% to guarantee better user experience
Tian et al. [19]	Compressed TRIES algorithm	Phases of Hadoop and its improvement with the help of tuning	Solve problem of low efficiency

(Continued)

Table 1: Continued

Author	Approach	Purpose	Outcome
Indrawati et al. [20]	MINLP and Lingo software application	The model provides optimum solution that maximizes the revenue and provides cloud services as per the need and request.	Reduce cost and demand for internet and increase provider profit
Optimization in SDN Qu et al. [21]	Optimization such as by grouping a short field in the search and merge phases and early termination	Large-scale packet classification on multicore processors	33 million packets per second and 2000 ns per-packet latency with 49 of peak throughput for 32 K 15-field rule set by optimized decomposition-based approach
Yao et al. [22]	Particle swarm optimization (PSO) merging with maximum flow	Satisfying objectives of global network like maximum flow with forwarding table size limitation	A certain level of QoS fairness among flows, with better performance in data center and backbone network
Budhrajya et al. [23]	Ant colony optimization technique and ML	Minimize data-transmission risks related to compliance and privacy	Preserve target compliance; use rules of intelligent eviction to obtain routing downloaded to switch
Aurizzi et al. [24]	Network function virtualization (NFV) and EHF	Counteract tropospheric fading	Use of SDN is good for intelligent management of proposed system
Raouf et al. [25]	ACOSDN	Optimize dynamic routing in SDNs	Handle dynamic network changes, reduce network congestion, and achieve high throughput with lower delay and packet loss rates
SDN-Cloud Integration Optimization Amiri et al. [26]	Analytic hierarchy process-based game aware routing (AGAR), Dijkstra	Find optimum transmission path of a packet in a data center while maximizing bandwidth utilization and minimizing delay	AGAR was able to reduce end delay by 9.5 compared to Hedera routing, ECMP routing, and Dijkstra based on delay

(Continued)

Table 1: Continued

Author	Approach	Purpose	Outcome
Phan et al. [27]	M/M/k and M/M/1/ ∞ models	Proactive optimal resource allocation for elastic security VNFs in chaining service function	Defined mathematical requirements to analyze VNF resource allocation function and estimate number of packets in system
Tajiki et al. [28]	Exact solution and fast suboptimal one schemes	Estimate flow matrix	Simulation results show packet loss reduced by 20 compared to conventional approaches; throughput increased by 30
Abdulqadder et al. [29]	PSO and chaotic secure hashing	User authentication and improved quality of service	OMNeT++ simulator implemented proposed Sec SDN-cloud and tested it in terms of bandwidth, latency, end-to-end delay, and packet loss, showing better performance

The comparative study shows that optimization algorithms such as PSO, Chaotic, MINLP, and their combinations can improve results in terms of different parameters.

4 Methodology

SDN enables the smooth integration of application provisioning in the cloud through automated interfaces, and fits naturally in cloud infrastructures such as IaaS, PaaS, and public or private clouds. SDN-based cloud networking solutions have advantages such as flexibility and scalability, but they support only certain network environments. The steps of the proposed approach are shown in Fig. 3.

Step 1: Specifications are considered, such as of machines and jobs allocated per machine.

Step 2: A plot shows the deployment of machines in the simulation. The SDN controller monitors and configures the network along with the discovery of topology.

Step 3: In the next part, we will show the transferring of allocated jobs in the same simulation as well as the backend processing required for it.

Step 4: The algorithm is applied to reduce bandwidth consumption and response time. The MET algorithm is applied, and the result is send as input to the lion optimization algorithm.

Step 5: Performance is evaluated to achieve high system efficiency based on latency, execution time, and throughput.

4.1 Algorithms Used in Proposed Work

We have analyzed that in the offline cloud scenario, machines are deployed to process user requests where the applied queue environments evaluate the communication for efficient

scheduling. Performance was evaluated to increase throughput and reduce latency, execution times, as well as bandwidth. During the first phase of implementation, minimum execution time algorithm has been applied to get the successful completion of allocated task, processes in the short interval of time without any error rate and Lion optimization has been used to reduce the bandwidth consumption and latency.

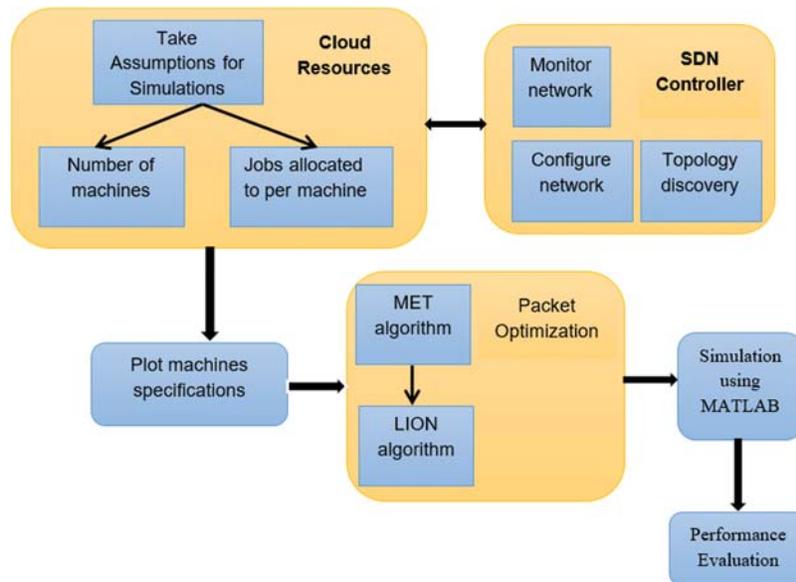


Figure 3: Methodology used in proposed work

4.1.1 Minimum Execution Time

MET computes the completion time of the resources that are being allocated to the virtual machines. The task is assigned to machines concentrated on the least execution time using essential facts of MET that sometimes deal with the great imbalance of the weight. The job is not conditional on machine accessibility.

- Initially, the jobs are being provided to the machines from where we listed out the total quantity of jobs which are further handpicked on priority basis.
- The current development id and time are used to know the execution and completion time of the process respectively.
- Evaluate the performance time of the process by the machine.
- Evaluate the estimated time at which machine m is available to execute task t .
- The predictable or minimum completion time is estimated using the sum of the execution time to complete the processes by the machine and the initial execution time of the current task on the current processor or machine.
- Run this process for the completion of iterations and stop until all the jobs are completed.
- Figure out the minimum execution time of the task by the machine until all the jobs get exhausted.

4.1.2 Lion Optimization

The process of the lion algorithm is given below [30]:

- Initialize random populations
- Initialize prides and lions
- For each lion particle:
 - Select a random female lion for hunting
 - Each female lion select the best position in the pride
 - Weakest lion pride out from the population and become the nomad
 - Each pride evaluate the imigration rate and become the nomad
- Evaluate the fitness function to select the best females and fill the mepty places which of the female lions which are migrated from the territory.

In [Tab. 2](#) of proposed work, we have made some assumptions regarding the specifications of cloud computing based on the best values achieved by evaluating it in MATLAB. Five queues are defined, in which the job count is 100 and the total time is 50 s. These five queues are deployed in different locations for loops according to the machine's specified count. Then comes a weighted round-robin step in which queues are selected according to priority, and scheduling is done using the MET algorithm, which locates the task with the minimum execution time. The scheduling task is carried out based on first come first served and available resources, and their load is not considered in this algorithm. Lion optimization is applied to the output of the MET algorithm. Before that, various parameters are defined for the initialization of LOA. From this, we get the best solution, which is used to find parameters such as band consumption, latency rate, and throughput for different execution times.

Table 2: Steps of the proposed algorithm

Step 1:	Assumptions for specifications of cloud computing: Job count = 100; Total time = 50; Five queues x_i ($i = 1-5$);
Step 2:	Deployment of the 5 queues with their locations: For $i = 1:x_i$ ($i = 1-5$) Plotting of machines specifications End
Step 3:	Selecting Queue according to the priority which is the weighted round robin step
Step 4:	Apply MET algorithm: Loop1 start: Loop2 start: Current_job=job_specification(3, i); current_time=job_specification(2, i) while total_time>0 arrival_time_job(i) = current_time; avail_time_mach(i) =(10-5).*rand(1, 1) + 5;

(Continued)

Table 2: Continued

	<p>Estimated time the amount of time the resource r will take to execute the task</p> <pre> estimated_time_mach = (time_per_job*job_specification(4,:))/arr_time_job(i); ready_time = (avail_time_mach(i)*job_specification(4,:))/arr_time_job(i); MET_val=est_time_mach + ready_time; minimum_computation_time(p) = min(MET_val); band_consumption(p) = sum(min_comp_time)*job_count; total_time = total_time-1; end1 end2 </pre>
Step 5:	<p>Apply Lion optimization algorithm on result obtained by applying MET. [rs, cs] = take the size of band consumption obtained from MET. iterations = rs; no_lions = cs; for 1:iterations normad_lions = band_consump(1:25); pride_lions =band_consump(25:50); hunters = numel(pride_lions); initial_nomad_pop = normad_lions(1:20); rest_nomad = normad_lions(numel(init_nomad_pop):end); Loop1: For i= 1: numel value of pride lions if i= numel value of normad lions break; end Find the best fitness and updated nomad end; end end Loop2: j= 1: numel value of updated nomad Get the best solution of lions End</p>
Step 6:	<p>Use the final value obtained from Lion optimization for finding the different parameters.</p> <pre> Band_consumption = min_comp_time*(sum(best_sol_lions)/job_count); latency_met = (band_consump_met*sum(min_comp_time))/job_count; throughput_met=((abs((band_consump_met))./(sum(min_comp_time)*job_ count)))* 100); end </pre>

The worst-case time complexity of the proposed algorithm is considered. This depends on the number of iterations, function evaluations, population size, and several loops. The ant lion's position initialization in a population of size NP has time complexity $O(NP)$. Furthermore, the time complexity is used to calculate the cost values of the ant lions as $O(NP)*O(F(x))$, where $F(x)$ is the object/cost function. The complexity of the lion also depends on the iteration [31].

4.2 Dataset

The due/packet-delivery dataset is used for the implementation. The dataset contains measurements of data delivery performance of a 802.15.4 link in an indoor scenario in which throughput, loss, delay, and energy were measured over six months under around 50,000 parameter configurations of seven key stack parameters, including around 200 million metadata transmission packets. There are two trace sets: delay and packet-metadata. The delay trace set has a pre-packet delay of packet transmission under various configurations of stack parameters. Metadata of each packet consist of noise floor, actual retransmission, arrival time, LQI, deliver success/fail, RSSI, actual queue size, and overflow or number traces.

5 Results and Discussion

We discuss below the generation of a cloud-based scenario using the MATLAB simulator with SDN, and how the proposed algorithms are used to optimize the performance for packet optimization.

5.1 Simulation Results

To realize the proposed algorithm, a framework was developed and simulated in MATLAB.

Fig. 4 shows the primary access points, which are green in color, queues in blue, and machines in red. The primary access point feeds the user request for resources to the queues, and the request will be transferred to machines to execute the jobs.

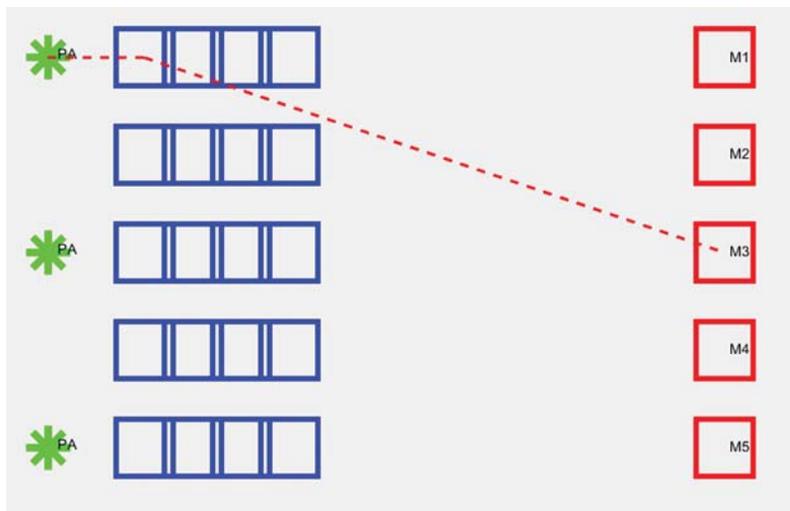


Figure 4: Deployment process

Fig. 5 shows the throughput of the computing network and simulation time, in seconds. The throughput must be high for efficient task execution. Fig. 6 shows the energy consumption achieved by the proposed approach and the total bandwidth consumption.

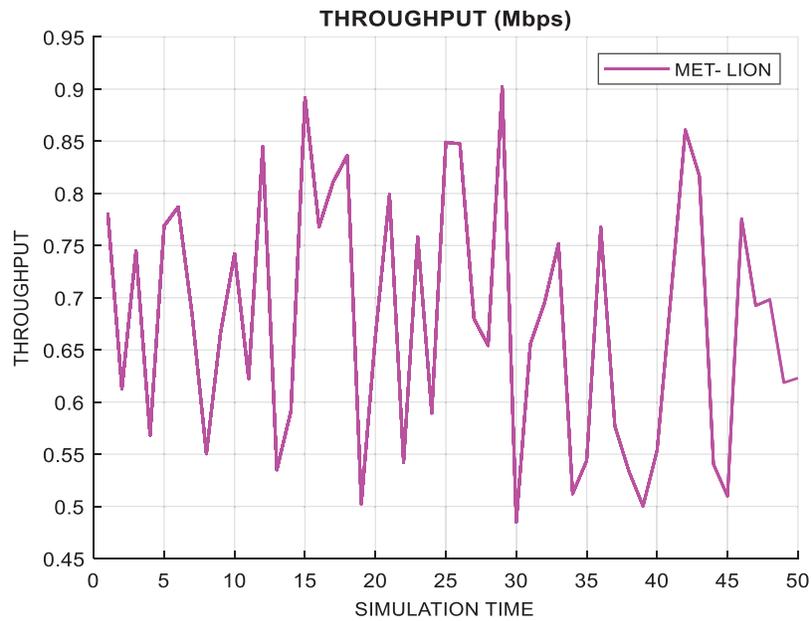


Figure 5: Throughput (Mbps)

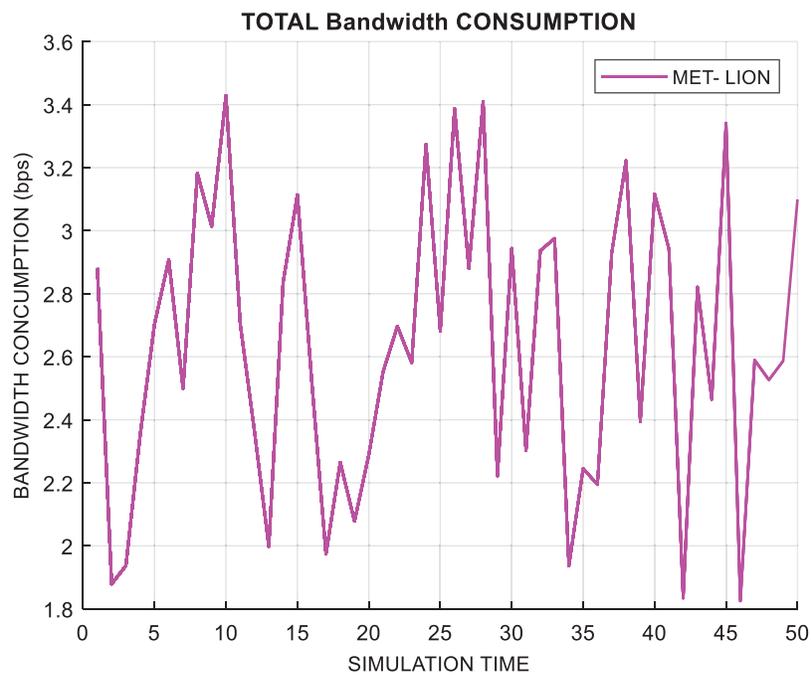


Figure 6: Bandwidth consumption

Fig. 7 shows the execution time, which must be low, and is important to the execution of tasks in short intervals. Tasks must be completed quickly because traffic is huge, and latency increases the queue, which will degrade performance. Fig. 8 shows the latency of the system which is going low with the increasing time. a. Increased latency degrades system performance, increases system overloading and bandwidth consumption.

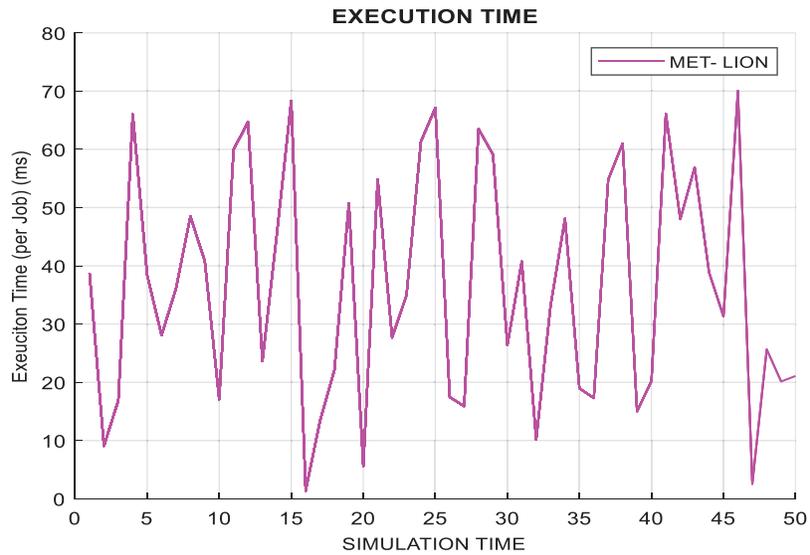


Figure 7: Execution time

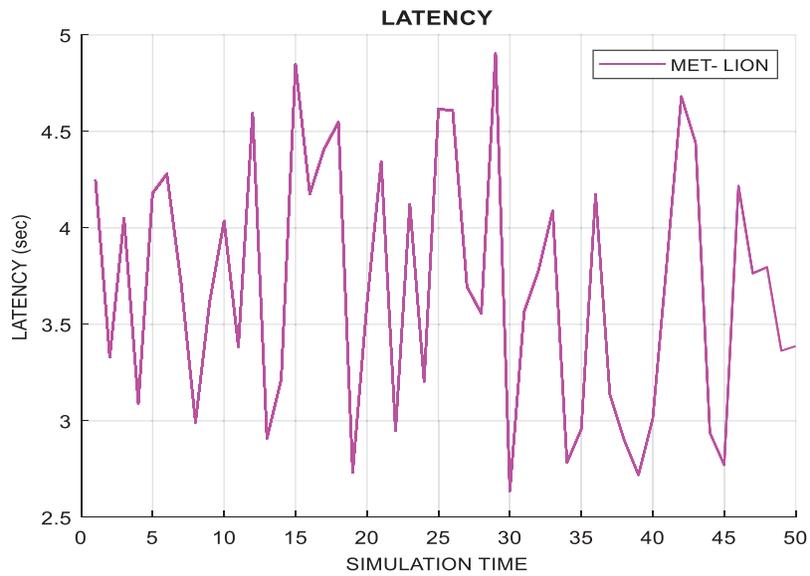


Figure 8: Latency

Tab. 3 shows the results using the proposed approach in terms of latency, execution time, total bandwidth, and throughput, in sec, ms, bps, and Mbps, respectively.

Table 3: Performance evaluation

Simulation time	Latency	Execution time (ms)	Total bandwidth (bps)	Throughput (Mbps)
10	4	15	3.4	0.75
20	3.5	5	2.1	0.6
30	3.8	27	2.9	0.46
40	2.5	18	3.1	0.54
50	3.7	22	3.2	0.65

Fig. 9 evaluates the proposed work in different simulation times. We considered five simulation times at 10-second intervals, and obtained results in terms of latency, execution time, total bandwidth, and throughput. For 10 simulation times, the latency obtained was 4 s, execution time was 15 ms, total bandwidth was 3.4 bps, and throughput was 0.75 Mbps. In the same way, for a 20 simulation time, the latency, execution time, total bandwidth, and throughput obtained are 3.5, 5, 2.1, and 0.6, respectively. In the same way, all parameters are executed for simulation times of 30, 40, and 50 and obtained different results for each.

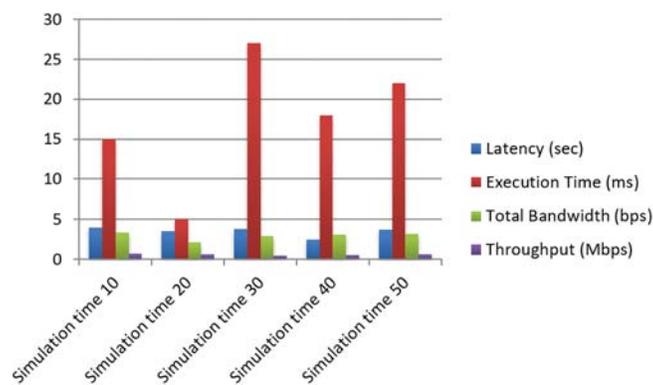


Figure 9: Performance evaluations in different simulation times

5.2 Comparison with Past Studies

Conventional networks are transformed into SDN by various enterprises to provide cost efficient and flexible network. But attacks and security breaches have exposed the weakness of SDNs. The most common, and most dangerous such attack is distributed denial of service (DDoS). Arivudainambi et al. [32] proposed an effective and accurate DDoS detection method using a lion optimization algorithm which is used to detect a DDoS attack with an accuracy of 96%. Srinivasa Ragavan et al., [33] proposed a hierarchical cluster-based routing scheme in which the sensor nodes are clustered using lion optimization algorithms and establish the routes from transmitting the data. The performance of the proposed approach was evaluated in terms of network lifetime,

energy consumption, packet delivery rate, and average latency, and the approach was found to improve the quality of service of the network in SDN.

SDN has become one of the most widely used network architectures to divide the control and forwarding planes. In the control plane, SDN centrally observes and regulates the network through software control, and manages the software defined by WAN, so there is a need for multiple controllers to handle the reliability and scalability issues of the network. To improve network performance, controller placement problem (CPP) is deployed which is threefold in nature i.e., the number of controllers to be placed in a network, the locations of these controllers and the assignment function of controllers to switches, in which all of them are important for the design of an efficient control plane. To get a reliable CPP, Verna-based optimization (VBO) was proposed by Ashutosh Kumar Singh et al. (2020) to reduce the total average latency of SDN [34]. The proposed approach was also compared with teacher-learning-based optimization, particle swarm optimization (PSO), and the Jaya algorithm in finding a reliable CPP. The best results were obtained using the VBO algorithm. In the load balancing work of SDN, Hai Xue et al. (2019) also used the ant colony optimization (ACO) metaheuristic approach [35]. The key criteria of ACO are optimal solutions and convergence latency. They integrated the genetic algorithm (GA) with ACO to enhance the performance of their proposed load balancing scheme in SDN. It combines the fast global search of GA and the efficient search of an optimal solution of ACO. Computer simulation results showed that the proposed scheme substantially improves the round robin and ACO algorithms in terms of the rate of search for an optimal path, round trip time, and packet loss rate [36]. Tab. 4 defines the comparative analysis of network B/W dependent DMMM algorithm latency, execution and LION optimization latency and execution.

Table 4: Comparative results

Simulation time	Network B/W dependent DMMM algorithm latency (ms)	Network B/W dependent DMMM algorithm execution (ms)	LION optimization latency (ms)	LION optimization execution time (ms)
10	50	0.88	4	15
20	10	0.54	3.5	5
30	15	0.33	3.8	27
40	30	0.90	2.5	18
50	25	0.64	3.7	22

Although the pride used in LOA is a stable social unit, it is a fission-fusion social group whose individuals are found in a range of subgroup compositions and sizes [37]. For residency of pride, male coalitions challenge one another, and males may use lethal aggression to tip the balance of power. The numerical advantage in territorial competition is a primary benefit of groups, and may have been important in the evolution of lion society [38]. The dependent offspring are evicted or killed by incoming males sired by the previous coalition to accelerate the return of mothers to sexual activity [39].

6 Conclusion and Future Directions

In this paper, specifications were taken as assumptions, such as the number of machines and allocated jobs per machine. Machines were deployed in MATLAB, and the minimum execution time along with LION optimization were used to solve the problem of packet optimization. Simulations were performed at five times with 10-second intervals, and results were obtained in terms of latency, execution time, bandwidth, and throughput. The results showed that the proposed method can solve the problem, and it gives better results than existing work. We generated a scenario in MATLAB that justifies the applicability of the proposed scheme in a real scenario. Although MET with LOA gave good results, it requires further testing. Premature convergence in LOA causes poor performance, and a strategy is needed that will not decrease the search space and will return to the original search space after a long period. The MET algorithm has limitations such as severe load imbalance. In the future, we will apply different hybrid metaheuristics and include the concepts of fog and edge computing. Testing is also needed on real-time datasets.

Funding Statement: This research was supported by the Sejong University Research Fund Korea and University of Shaqra, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson *et al.*, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Communication*, vol. 38, no. 2, pp. 69–74, 2008.
- [2] N. Kapoor and Y. Kumar, "The efficient management of renewable energy resources for vanet cloud communication," *Nature-Inspired Computing Applications in Advanced Communication Networks*, vol. 5, pp. 228–253, 2020.
- [3] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado *et al.*, "Nox: Towards an operating system for networks," *SIGCOMM Comput. Communication*, vol. 38, no. 3, pp. 105–110, 2008.
- [4] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski *et al.*, "Onix: A distributed control platform for large-scale production networks," in *Proc. OSDI*, vol. 10, no. 9, pp. 351–364, 2010.
- [5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson *et al.*, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [6] C. J. Sher Decusatis, A. Carranza and C. M. Decusatis, "Communication within clouds: Open standards and proprietary protocols for data centre networking," *Communications Magazine*, vol. 50, no. 9, pp. 26–33, 2012.
- [7] Y. Vigfusson and G. Chockler, "Clouds at the crossroads: Research perspectives," *XRDS: Crossroads the ACM Magazine for Students*, vol. 16, no. 3, pp. 10–13, 2010.
- [8] N. Kiran, C. Pan and Y. Changchuan, "Reinforcement learning for task offloading in mobile edge computing for sdn based wireless networks," in *Seventh Int. Conf. on Software Defined Systems (SDS)*, vol. 4, no. 9, pp. 268–273, 2020.
- [9] S. Azodolmolky, P. Wieder and R. Yahyapour, "SDN-based cloud computing networking," in *15th Int. Conf. on Transparent Optical Networks (ICTON)*, vol. 4, no. 7, pp. 1–5, 2013.
- [10] A. Autenrieth, J. P. Elbers, P. Kaczmarek and P. Kostecki, "Cloud orchestration with SDN/OpenFlow in carrier transport networks," in *15th Int. Conf. on Transparent Optical Networks (ICTON)*, vol. 3, pp. 1–4, 2013.
- [11] J. P. Elbers and A. Autenrieth, "Using openFlow to extend software defined networking," *SDN Journal Article*, vol. 6, no. 5, pp. 302–336, 2013.

- [12] A. Alnoman and A. Anpalagan, "A SDN-assisted energy saving scheme for cooperative edge computing networks," in *IEEE Global Communications Conf. (GLOBECOM)*, vol. 4, no. 6, pp. 1–6, 2020.
- [13] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, C. N. Correa, S. C. de Lucena *et al.*, "Virtual routers as a service: The route flow approach leveraging software-defined networks," in *Proc. of the 6th Int. Conf. on Future Internet Technologies*, ACM, vol. 11, no. 3, pp. 34–37, 2011.
- [14] C. E. Rothenberg, M. R. Nascimento, M. R. Salvador, C. N. A. Correa *et al.*, "Revisiting routing control platforms with the eyes and muscles of software-defined networking," in *Proc. of the First Workshop on Hot Topics in Software Defined Networks*, ACM, vol. 4, no. 12, pp. 13–18, 2012.
- [15] A. Montazerolghaem and M. H. Yaghmaee, "Green cloud multimedia networking: NFV/SDN based energy-efficient resource allocation," *Trans. on Green Communications & Networking*, vol. 4, no. 5, pp. 873–889, 2020.
- [16] L. A. Rocha and E. Cardozo, "A hybrid optimization model for green cloud computing," in *IEEE/ACM 7th Int. Conf. on Utility and Cloud Computing*, vol. 14, no. 7, pp. 11–20, 2014.
- [17] M. Liu, W. Dou, S. Yu and Z. Zhang, "A decentralized cloud firewall framework with resources provisioning cost optimization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 621–631, 2015.
- [18] P. Raad, S. Secci, C.-D. Phung and P. Gallard, "PACAO: A protocol architecture for cloud access optimization in distributed data center fabrics," in *Proc. of the 1st IEEE Conf. on Network Softwarization*, vol. 4, no. 7, pp. 1–9, 2015.
- [19] X. Tian, B. Cui, J. Deng and J. Yang, "The performance optimization of hadoop during mining online education packets for malware detection," in *10th Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing*, vol. 10, no. 1, pp. 305–309, 2016.
- [20] I. Indrawati, F. M. Puspita, S. Erlita, I. Nadeak and B. Arisha, "LINGO-based optimization problem of cloud computing of bandwidth consumption in the internet," in *Int. Conf. on Information and Communications Technology (ICOIACT)*, vol. 2, no. 9, pp. 436–441, 2018.
- [21] Y. Zhu, J. Zhang, L. Li and W. Peng, "Multiple ant colony routing optimization based on cloud model for WSN with long-chain structure," in *6th Int. Conf. on Wireless Communications Networking and Mobile Computing (WiCOM)*, vol. 12, no. 3, pp. 1–4, 2010.
- [22] Y. R. Qu, S. Zhou and V. K. Prasanna, "Performance modeling and optimizations for decomposition-based large-scale packet classification on multi-core processors," in *IEEE 15th Int. Conf. on High Performance Switching and Routing (HPSR)*, vol. 6, no. 3, pp. 154–161, 2014.
- [23] X. Yao, H. Wang, C. Gao and S. Yi, "Maximizing network utilization for SDN based on particle swarm optimization," in *IEEE 28th Int. Conf. on Tools with Artificial Intelligence*, vol. 6, no. 9, pp. 921–926, 2016.
- [24] K. K. Budhraj, A. Malvankar, M. Bahrami, C. Kundu *et al.*, "Risk-based packet routing for privacy and compliance-preserving SDN," in *IEEE 10th Int. Conf. on Cloud Computing*, vol. 3, no. 2, pp. 761–765, 2017.
- [25] M. M. Aurizzi, S. Milana, T. Rossi, E. Cianca and M. Ruggieri, "SDN for smart gateway diversity optimization in high throughput satellite systems," in *IEEE Aerospace Conf.*, vol. 12, no. 7, pp. 1–5, 2019.
- [26] M. Amiri, A. Sobhani, H. A. Osman and S. Shirmohammadi, "SDN-enabled game-aware routing for cloud gaming datacenter network," *IEEE Access*, vol. 99, no. 1, pp. 18633–18645, 2017.
- [27] T. V. Phan, N. K. Bao, Y. Kim, H.-J. Lee and M. Park, "Optimizing resource allocation for elastic security VNFs in the SDNFV-enabled cloud computing," *Int. Conf. on Information Networking (ICOIN)*, vol. 5, no. 2, pp. 163–166, 2017.
- [28] M. M. Tajiki, B. Akbari and N. Mokari, "QRTP: QoS-aware resource reallocation based on traffic prediction in software defined cloud networks," in *8th Int. Sym. on Telecommunications*, vol. 8, no. 5, pp. 527–532, 2017.
- [29] I. H. Abdulqadder, D. Zou, I. T. Aziz, B. Yuan and W. Li, "Secsdn-cloud: Defeating vulnerable attacks through secure software-defined networks," *IEEE Access*, vol. 6, no. 99, pp. 8292–8301, 2018.

- [30] K. Kaur and Y. Kumar, "Swarm intelligence and its applications towards various computing: A systematic review," in *Int. Conf. on Intelligent Engineering and Management (ICIEM)*, IEEE, vol. 2013, no. 7, pp. 1–6, 2020.
- [31] H. Yuana, J. Bib, B. H. Lia and W. Tanc, "Cost-aware request routing in multi-geography cloud data centres using software-defined networking," *Enterprise Information Systems*, vol. 11, no. 3, pp. 359–388, 2015.
- [32] D. Arivudainambi, K. A. Varun Kumar and S. Sibi Chakkaravarthy, "LION IDS: A meta-heuristics approach to detect DDoS attacks against software-defined networks," *Neural Computing and Applications*, vol. 31, no. 3, pp. 1–18, 2019.
- [33] P. S. Ragavan and K. Ramasamy, "Software defined networking approach based efficient routing in multihop and relay surveillance using lion optimization algorithm," *Computer Communications*, vol. 150, no. 15, pp. 764–770, 2020.
- [34] A. K. Singh, S. Maurya, N. Kumar and S. Srivastava, "Heuristic approaches for the reliable SDN controller placement problem," *Trans Emerging Tel Tech*, vol. 31, no. 2, pp. 1–18, 2020.
- [35] H. Xue, K. T. Kim and H. Y. Youn, "Dynamic load balancing of software-defined networking based on genetic-ant colony optimization," *Sensors*, vol. 19, no. 2, pp. 1–17, 2019.
- [36] S. Khan, M. A. Irfan, K. Khan, M. Khan, Khan *et al.*, "ACO based variable least significant bits data hiding in edges using IDIBS algorithm," *Symmetry*, vol. 12, pp. 781–788, 2020.
- [37] M. Yazdani and F. Jolai, "Lion optimization algorithm (LOA): A nature-inspired metaheuristic algorithm," *Journal of Computational Design and Engineering*, vol. 3, no. 1, pp. 24–36, 2016.
- [38] M. Alam and K. A. Shakil, "An NBDMMM algorithm based framework for allocation of resources in cloud," *Resource Allocation in Cloud Environment*, vol. 14, no. 3, pp. 1–12, 2014.
- [39] O. A. Raouf and H. Askr, "ACOSDN-Ant colony optimization algorithm for dynamic routing in software defined networking," in *14th Int. Conf. on Computer Engineering and Systems (ICCES)*, vol. 2019, no. 5, pp. 141–148, 2019.