RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt2531

An Energy-Efficient Branch and Bound Optimization Algorithm for Large-Scale Software-Defined Networks

S. Sathish [a, *], J. Poongodi [b], G. Sudha [c], S. Sindhuja [d]

[a] Department of Computer Science and Business Systems, Bannari Amman Institute of Technology, Sathyamangalam-638 401, Tamil Nadu, India.

[b] Department of Computer Science Engineering, Dr.N.G.P Institute of Technology, Coimbatore-641 048, Tamil Nadu, India.

[c] Department of Computer Science Engineering, Sona College of Technology, Salem-636 005, Tamil Nadu, India

[d] Department of Computer Science Engineering (AI&ML), Sri Eshwar College of Engineering, Coimbatore-641 202, Tamil Nadu, India.

* Corresponding Author Email: sathesapcse@gmail.com

DOI: https://doi.org/10.54392/irjmt2531

Received: 29-08-2024; Revised: 17-03-2025; Accepted: 01-04-2025; Published: 08-04-2025

Abstract: Software-defined network offers a programmable networking environment that redefines the management of conventional networking and can provide potential solutions for their well-known challenges. In SDN circumstances, energy becomes a major threatening factor that affects both the reliability of the network and the sustainability of its connections. Energy depletion in SDN is still a major concern considering the ever-changing network constraints and rapid growth in the number of networking devices. This research article introduces a novel energy-efficient branch-and-bound optimization (EE-BBO) algorithm, designed for large-scale software-defined networks to overcome the challenges faced by existing approaches. The objective of the EE-BBO algorithm is to minimize the energy consumption across the SDN networks and improve the network performance. The proposed algorithm computes the lower bounds for prioritizing nodes, classifies them based on their probable, and intelligently forwards packets to choose the most energy-efficient route. The algorithm is implemented on Mininet, using Floodlight as the SDN controller and OpenFlow as the communication protocol. The results of simulations showed that the proposed EE-BBO algorithm outperforms the current benchmarked methods in terms of energy consumption by 9-19%, packet loss by 15-28%, and enhancing network lifetime by 14-25%.

Keywords: Software-Defined Network, Energy, Branch and Bound, Optimization, Network Lifetime

1. Introduction

Wireless sensor networks (WSNs) are progressively playing a substantial role in present-day networking. Conventional WSNs face a lot of challenges, [1, 2], due to the swiftly growing networking conditions and various networking devices. These challenges include energy consumption, limited computational power, scalability, heterogeneity, rigid network management, and challenges in regulating new technologies in the comeback to altering network requirements. Software-defined networks (SDN) [3]. [4], a revolutionary standard that provides real-world answers to the problems faced by WSNs, was developed as a result of the growth of software-based programmable networks.

SDN's flexibility, centralized control, improved programmability, and efficient network administration have made it a vital element of many applications today [5, 6]. SDN architecture [7], includes three planes: the application plane, which is at the top, the control plane, which makes conclusions about data forwarding over the network, and the data plane, which manages data packet transmission among devices. Through centralization of network control and management, the SDN architectural design simplifies network administration and maximizes network resources by undertaking away the necessity for individual devices to manage network operations. Figure 1 [7], displays the simplified structure of SDN.

Even though SDN has plentiful profits for WSNs, there are several weaknesses as well [8, 9]. Performance, architectural design, security, and energy efficiency are some of the main problems SDN expressions. From these, energy [10], is critical for preserving the nodes' lifespan, which is essential to keep the network running. The energy consumption of nodes is powerfully predisposed by data transmission, processing activities, and environmental boundaries. It is hard to grow the lifespan of SDN nodes due to their inadequate energy resources.

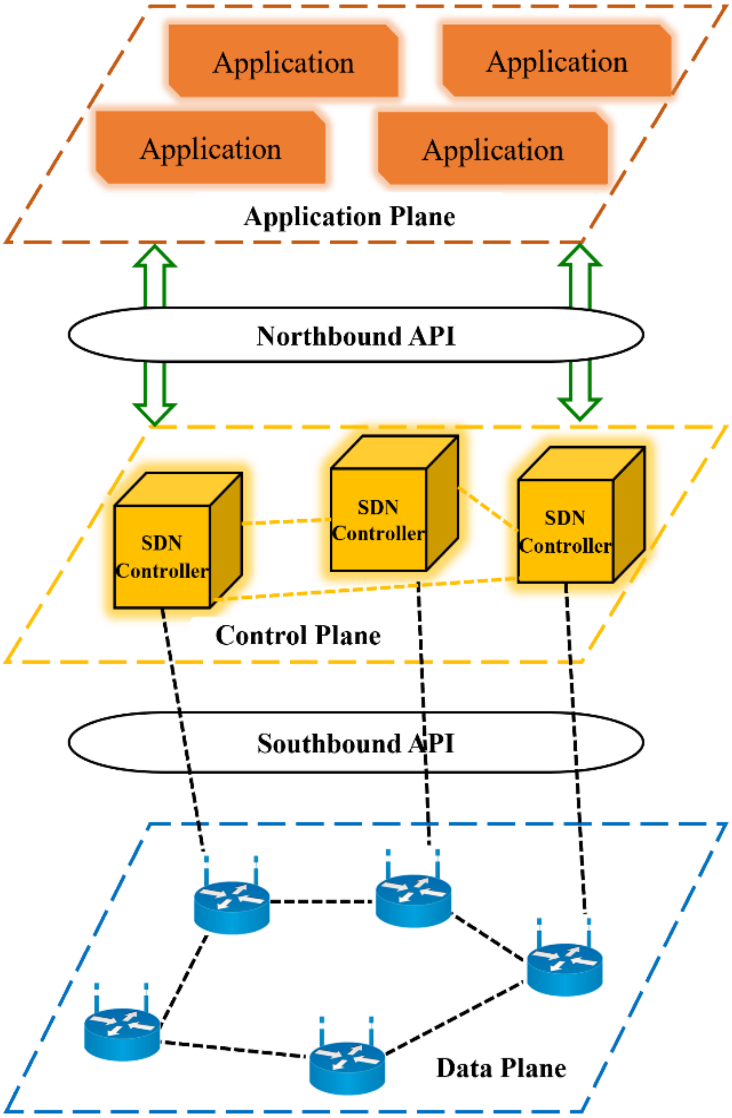

Figure 1. SDN architecture

Taking these things into account, making an effective energy management strategy is central for SDN settings, particularly in large-scale applications.

A novel energy-efficient branch-and-bound optimization (EE-BBO) algorithm is proposed in this research with the goal line of lowering energy consumption in software-defined networks, enhancing network performance, and extending network lifetime. To improve routing paths and decrease energy consumption, the EE-BBO algorithm uses branch-and-bound optimization techniques in conjunction with "dynamic source routing" (DSR) [11] protocol. The following are the contributions made by the EE-BBO algorithm:

- Initial Energy Calculation: During the packet broadcasting process, the algorithm computes the initial energy required by the source node to transmit and the energy required in the reception process by all the neighboring nodes.
- Lower Bound Calculation: The EE-BBO algorithm further calculates the lower bounds of each node and prioritizes their selection. Then it computes the total energy consumption of node traversal along the selected path.
- Energy Consumption During Forwarding: The algorithm computes the energy consumption of each node during the packet-forwarding phase.
- Overall Energy Calculation: The total energy consumption of the routing process is determined by summing the energy levels of node broadcasting, traversal, forwarding, and finalization.

These phases significantly enhance the energy efficiency of the proposed EE-BBO algorithm by ensuring the selection of optimal routing paths during the routing process in software-defined networks.

The remainder of the paper is structured as follows: Section 2 presents an extensive review of different energy efficiency techniques in SDN. The problem formulation is presented in section 4. Section 4 explicates the EE-BBO algorithm. Section 5 discusses the performance of the EE-BBO algorithm, with results compared against existing approaches. At last, section

5 concludes the research article by highlighting the key contributions and the scope for future improvements.

2. Literature Review

Many authors [10, 12, 13] have contributed to enhancing energy efficiency in SDN-based applications. This section critically reviews the major contributions aimed at improving energy efficiency in SDN.

Iqbal *et al.* [14], developed a stochastic model for analyzing end-to-end delay in OpenFlow-based networks, addressing a gap in existing research on this topic. The methodology involves conducting Internet-scale experiments on platforms like Mininet, GENI, and OF@TEIN, utilizing a log-normal distribution to model delays. Huin *et al*.[15], proposed an approach known as SENAtoR, which encompasses tunneling, traffic variation detection, and integer linear programming techniques to optimize energy consumption without compromising service quality.

Vijaygokul *et al*. [16] address traffic congestion and packet loss in Software Defined Networking (SDN) by proposing a fuzzy logic-based method to reduce table flow occupancy. The methodology involves detecting large traffic flows and applying functions to professionally decrease packet loss.

The interoperability of programmable network ideas, particularly SDN and NFV, in cloud-based LoWPANs such as 6LoWPAN and ZigBee was examined by Al-Kaseem *et al.* [17] By enlightening packet processing, letting down latency over active routing, and reducing the requirement for control packet exchanges, the SD-NFV technique advances end-to-end interruption and makes 6LoWPAN networks more responsive and effective.

The end-to-end delay in Software-Defined Networking (SDN) was inspected by Zhang et al. [18], to recognize the roots of delays in large-scale networks. The process requires generating an analytical model for delay, evaluating the effect of TCAM updates on packet delay, and calculating parameters associated with flow setup rates. The Naive Bayes method was used by EL-Garoui *et al.* [19], to suggest a routing protocol that decreases end-to-end delay in smart city environments, particularly in intelligent transportation systems, by integrating Software-Defined Networking (SDN) and Machine Learning (ML). Through this technology, a substantial dataset is produced from open data in Montreal, processed for analysis, and then implemented into the RYU SDN controller.

Priyadarsini *et al.* [20], observed into in what way to enhance route choice and considerably lesser the network's energy usage by merging an effective load-balancing method, a sleep-active mode mechanism, and a heuristic-based routing algorithm. An offline-phase ant colony optimization algorithm was created by Torkzadeh *et al.* [21] to professionally poise the load and satisfy QoS requirements in crowded networks with little flow. Zhao *et al.* [22], presented PESD-DCN, a correlation-aware flow consolidation technique that progresses scheduling and routing in SDN. It reduces state transitions, avoids link congestion, and decreases the need to wake idle devices, all of which grow energy efficiency.

Akbar *et al.* [23], recommended a method that uses an adaptive switching mechanism built into an SDN controller to advance packet loss and delay in Internet of Things applications. By improving the communication routes between source and destination nodes, this method effectively depresses packet losses and end-to-end delays. The method discovers Pareto-optimal pathways that strike stability among lowering transmission time and maximizing path dependability by using a multi-objective optimization algorithm. Additionally, the system model assurances that the selected routes satisfy quality of service (QoS) standards, which is vital for applications that are sensitive to delays.

For Software-Defined IoT (SDIoT) networks, Saha *et al*. [24], projected a traffic-aware QoS routing system designed to meet the QoS needs of heterogeneous flows, mainly packets that are loss-sensitive and delay-sensitive. The methodology calculates the finest routing paths while taking these QoS criteria into account consuming a greedy heuristic based on Yen's K-shortest paths algorithm. In order to match network topology with traffic demands and maximize energy consumption during periods of low traffic, Wang *et al*. [25], advised dynamic topology switching and reliability measures.

Compared to round-robin and random methods, the server metric collection load balancing algorithm (SMC-LB), which was offered by Reddy *et al.* [26], reduces energy consumption by putting servers into sleep mode whereas they are not processing requests. The hybrid spreading load algorithm (HSLA), proposed by Galan-Jimenez *et al*.[27], minimizes network power consumption while maximizing link utilization in IP/SDN hybrid networks by solving a multi-objective optimization problem. Ghani *et al.* [28] optimize packet forwarding in routers using the Particle Swarm Optimization (PSO) based Weighted Round-Robin (psoWRR) algorithm to reduce packet loss and starvation, particularly for low-priority packets. This approach involves simulating a multilevel queuing system where the PSO algorithm dynamically adjusts the weights of priority queues based on traffic conditions.

Forghani *et al.* [29], proposed the krill herd metaheuristic algorithm (KHMA) an optimization-based approach to improve energy efficiency in SDN networks by effectively balancing the load. This algorithm collects virtual machine specifications, establishes network topology, checks the number of links required for individual tasks, computes krill herd values, and selects the best virtual machine based on these values to compute load-balancing metrics and update the weights until all tasks are processed.

Deo *et al.* [30], apply the flow statistics from OpenFlow switches to address bandwidth allocation and minimize packet loss in bursty traffic scenarios using an algorithm that leverages OpenFlow meters. This approach optimally allocates bandwidth to various Quality of Service (QoS) classes based on their current needs. However, a limitation is that while the algorithm guarantees initial minimum bandwidth allocation, it does not minimize packet loss in all cases.

Various studies have aimed to optimize energy efficiency in distinct applications such as data centers [31], smart homes [32], and IoT applications [33]. These approaches leverage SDN capabilities, virtual networks, and optimization algorithms to reduce energy consumption and enhance network performance. The summary of this literature review is presented in Table 1.

Table 1. Summary of the literature review

Reference	Techniques	Highlights	Findings	Constraints
[14]	Stochastic model, log-normal distribution	End-to-end delay analysis in OpenFlow-based networks	Log-normal distribution effectively models delay; validated through experiments on Mininet, GENI, and OF@TEIN	Focused primarily on delay modeling; does not address energy efficiency
[15]	Energy-aware routing techniques	Developing a SENAtoR with dynamic tunnel selection and one with a pre-configured set of tunnels	Power conservation without compromising performance	Insufficient investigation of energy efficiency and service quality effects across diverse network types
[16]	Fuzzy logic-based method	Reduces table flow occupancy in SDN	Effectively minimizes traffic congestion and packet loss	Applicability limited to SDN environments
[17]	SDN-NFV integration	Interoperability in cloud-based LoWPANs (e.g., ZigBee, 6LoWPAN)	Enhanced end-to-end delay, reduced latency through efficient routing	Focused on 6LoWPAN networks; less generalizability
[18]	Analytical delay model	Investigated factors contributing to delays in SDN	Quantified delay impact from TCAM updates and flow setup rates	Limited scope to large-scale SDN environments
[19]	SDN-ML integration (Naive Bayes)	Routing protocol for smart cities and transportation systems	Reduced end-to-end delay using datasets from real-world scenarios	Heavy reliance on dataset quality for results
[20]	Load balanced techniques	Energy-efficient load distribution with sleep-wake cycles and heuristic routing	Equilibrium between performance and energy optimization	Validation is primarily required within SDN environments
[21]	Energy-aware routing techniques	Two-phase routing utilizing Ant Colony Optimization	Decreased energy consumption, improved link management, reduced network congestion	Inadequate evaluation of load distribution effectiveness
[22]	Energy-aware routing	PESD-DCN: Flow routing considering correlations	Enhanced energy efficiency through minimization of active components	Service quality implications not thoroughly examined

[23]	Adaptive switching mechanism, multi-objective optimization	Enhanced delay and packet loss in IoT applications	Identifies Pareto-optimal paths balancing delay and reliability	Dependent on QoS-sensitive applications
[24]	Traffic-aware QoS routing, Yen's K-shortest path	Routing scheme for SDIoT	Optimized routing for delay-sensitive and loss-sensitive packets	Greedy heuristic might not perform optimally in highly dynamic traffic
[25]	Topology driven techniques	Dynamic topology switching and reliability measures for SDNs	Optimized energy usage during low-traffic periods	Insufficient assessment of broader applicability
[26]	Energy-aware routing techniques	Adaptive load balancing employing SDN and linear programming	Improved performance and efficient workload distribution	Application limited to data center environments
[27]	Load balanced techniques	HSLA for load balancing and energy minimization	Suitable for large-scale topologies with efficient link deactivation	Emphasis on the transition from traditional IP to SDN networks
[28]	PSO-based Weighted Round Robin	Packet forwarding optimization in routers	Reduced packet loss and starvation in priority queues	May introduce additional processing delays
[29]	Load balanced techniques	Krill herd meta-heuristic algorithm for SDN optimization	Enhanced performance and reduced energy consumption	Limited consideration of network heterogeneity
[30]	OpenFlow meter-based algorithm	Bandwidth allocation for bursty traffic	Optimally allocates bandwidth to QoS classes	Does not minimize packet loss in all scenarios

These gaps highlight the need for a dynamic, scalable, and energy-efficient solution for SDNs that can address heterogeneous and large-scale environments. This necessitates the formulation of a problem that captures these challenges while providing a basis for innovation. The problem formulation, derived from these insights, is detailed in the following section.

3. Problem Formulation

The limitations and gaps identified in the literature review serve as the foundation for formulating the problem addressed in this research. In recent years, Software-Defined Networking (SDN) has emerged as a promising technology for enhancing the flexibility and efficiency of network management, particularly in large-scale applications. However, as SDN scales, energy efficiency becomes a critical challenge due to the increasing demands on network devices, which must operate continuously to manage traffic routing, control, and data aggregation tasks. Although numerous approaches have been proposed to improve energy efficiency, most focus on minimizing active network components, optimizing routing paths, or balancing loads using various heuristic and optimization techniques.

Despite these advancements, current methodologies primarily emphasize shortest-path routing or selective activation of high-utilization paths, often neglecting the optimality of path selection in terms of energy consumption. These limitations result in increased energy usage, higher packet loss rates, and degraded performance, particularly under high-load and diverse traffic conditions.

Additionally, existing approaches often lack adaptability to heterogeneous network conditions and are generally constrained by issues related to scalability, load distribution efficiency, or applicability to specific environments. There is a pressing need for an SDN-based energy-efficient algorithm that can balance loads across the network while minimizing packet loss and reducing delay, especially in large-scale and diverse applications.

This research article aimed to develop an energy-efficient branch-and-bound optimization (EE-BBO) algorithm for large-scale software-defined networks to overcome the challenges confronted by the existing approaches. The detailed implementation of the proposed EE-BBO is presented in the next section.

4. Proposed Methodology

The idea of the proposed EE-BBO algorithm focuses on minimizing the energy consumption of software-defined networks during routing and elongating the network lifetime. The implementation of the proposed EE-BBO algorithm encompasses the following phases:

- Initialization phase
- Bound calculation phase,
- Energy assessment and forwarding phase
- Path finalization phase.

4.1 Initialization phase

The source node is found and earmarked as the starting point for the routing process during the initialization phase. The source node's value is set to A. Next, the algorithm verifies whether the source node is a present node. If so, it appends itself to the routing path and disseminates a route request packet to its neighboring nodes. The destination node and the desired route information are included in this broadcast packet. This broadcasting process originates from the source node and alerts its neighbors to find a path to the destination. The total energy consumption during this broadcast is calculated as follows:

$$E_{br} = E_{tx}(src) + \sum_{i=neighbours\ src}^{n} E_{rx}(i) \quad (1)$$

where,

$E_{tx}(src)$ represents the energy transmission of the source node,

$E_{rx}(i)$ represents the energy required for reception by the neighboring nodes,

i is the index for each neighboring node,

E_{br} indicates the total energy consumption during the broadcast phase.

4.2 Bound Calculation and Node Traversal

In this phase, the lower bound of all nodes in set S is computed using a bounding procedure to prioritize which node should be explored first. This lower bound indicates the minimum feasible cost from the source node to the rest of the nodes in set S and is calculated as follows::

$$LB(i) = \min\left(cost(i,j) + cost(j,dst)\right) \quad (2)$$

where,

$LB(i)$ represents the lower bound for node (i),

j represents the neighboring node of (i),

dst represents the destination node.

Equation (2) computes the lower bound of node i, by considering the minimum value of the sum of the costs to move from node i, to neighboring node j and the cost from node j to the destination. Based on the lower bounds, nodes in set SSS are sorted in non-decreasing order, ensuring that nodes with the minimum cost (in terms of distance from the source) are prioritized. This process successively computes each newly generated node, relaying data packets or routing messages to subsequent nodes in the sequence. The energy consumed during node traversal is calculated as follows:

$$E_{tr} = \sum_{i\in P} E_{tx(i)} + E_{rx(i)} \quad (3)$$

where,

E_{tr} represents the energy for node traversal,

E_{tx} represents the energy transmission of the source node,

$E_{rx(i)}$ represents the energy for reception by neighboring node,

P is the set of paths.

The algorithm updates the optimal path if the current node's minimal cost falls below the overall maximum threshold, ensuring the selection of the most promising nodes for finding the optimal path.

4.3 Energy assessment and packet forwarding

The third phase involves computing the minimum threshold for the node under examination and setting an initial value for the peak energy level. The energy evaluation of the node is calculated using the following equation:

$$E_{eval}(i) = \alpha \times \frac{1}{LB(i)} + \beta \times e(i) \quad (4)$$

where.

$E_{eval}(i)$ represents the energy evaluation of node i,

α , β are coefficients representing the energy levels of node i,

$e(i)$ is the energy level of node i.

The energy evaluation balances the energy level of node *i,* and its lower bound to the destination. The algorithm then identifies the node with the highest energy value in set *S* using the following equation:

$$E_{hev} = \max e(i)\ (i \in S) \quad (5)$$

where,

E_{hev} Represents the highest energy value of node$i$.

The total energy consumption during packet forwarding is computed as follows:

$$E_{fr} = E_{tx}\,(E_{hev}) + E_{rx}(next\ node) \quad (6)$$

where,

E_{fr} represents the energy for packet forwarding,

$E_{tx}\,(E_{hev})$ is the energy consumption of the node with the highest energy value during packet transmission,

$E_{rx}(next\ node)$ Represents the energy consumed by the next receiving node during packet reception.

4.4 Path Finalization

In the final phase, all branching operations are completed, and the node with the highest energy capacity is identified. The algorithm sets a flag F to 0 and initializes it to 1. If the current node is the highest energy value node and F is 0, the path is broken. Otherwise, the optimal path is selected. The flag indicates whether the path is finalized as optimal or needs to be broken and rerouted.

The total energy consumption for finalizing the path, including possible rerouting, is calculated as follows:

$$E_{fn} = E_{tr} + \sum_{i \in Palt} E_{proc}\,(i) \quad (7)$$

where,

$E_{proc}(i) = E_{fd} + E_{update}$,

E_{fn} is the finalize energy consumption,

E_{tr} is the energy for node traversal,

$Palt$ is the set of the alternate node path,

$E_{proc}(i)$ is the sum of alternate energy paths of node i.

Equation 7 calculates the total energy consumption of the routing process as the sum of the energy consumed by node *i* during traversal and the energy required for updates, including the evaluation of alternate paths and feasible rerouting paths.

Finally, the total energy consumption of the routing process is computed as follows:

$$E_{total}(i) = E_{br}(i) + E_{tr}(i) + E_{fd}(i) + E_{update}(i) + E_{fn}(i) \quad (8)$$

where,

$E_{total}(i)$ Represents the total energy consumption of the routing process.

By combining these measures, the proposed EE-BBO algorithm minimizes overall energy consumption during the routing process, which is crucial for extending the lifespan of software-defined networks.

The proposed EE-BBO approach is outlined in Algorithm 1 and the process flows are illustrated in Figure 3.

Algorithm 1. EE-BBO Algorithm

Input:
src: Source node
dst: Destination node
S: Set of nodes
e(i): Energy value of node i
ic: Iteration counter
hev: Node with the highest energy vae
Output:
Shortest path nodes from src to dst
BEGIN
// Initialization
Set A ← src // Define source node as A
IF current node = src THEN
Add src to path
Send route requests to neighboring nodes
END IF
// Bound Calculation and Node Traversal
Determine lower bounds for each node in set S
Sort nodes in S by ascending lower bounds
FOR each node in S DO
Explore new nodes in sequence
Relay packets or route information to the next node
IF new node's lower bound < current node's lower bound THEN
Update path to include the new node
END IF
END FOR
// Energy Assessment and Packet Forwarding
Evaluate the lower bound of the current node
Set initial maximum energy level
Find hev in S
Forward route requests along the current path
// Path Finalization
Complete all branching operations
Identify the node with the highest energy
IF route is broken THEN
Select an optimal alternative route

END IF

End current process

Reset all routing flags

IF termination conditions are met THEN

End the path

ELSE

Choose the next optimal route

END IF

END

The physical network architecture and topology of the proposed EE-BBO approach are illustrated in Figure 2. The data plane consists of multiple nodes, where S represents the source node and D represents the destination node. The source node S broadcasts a route request packet to all neighboring nodes N*i*. It then computes the energy evaluation value for each node *i* in the network, which helps determine the priority of nodes for packet forwarding. Nodes with higher energy values are more favorable for forwarding. In Figure 2, the red-colored path S-N14-N13-N12-D has a low cost and high energy value, making it the optimal path for forwarding packets.

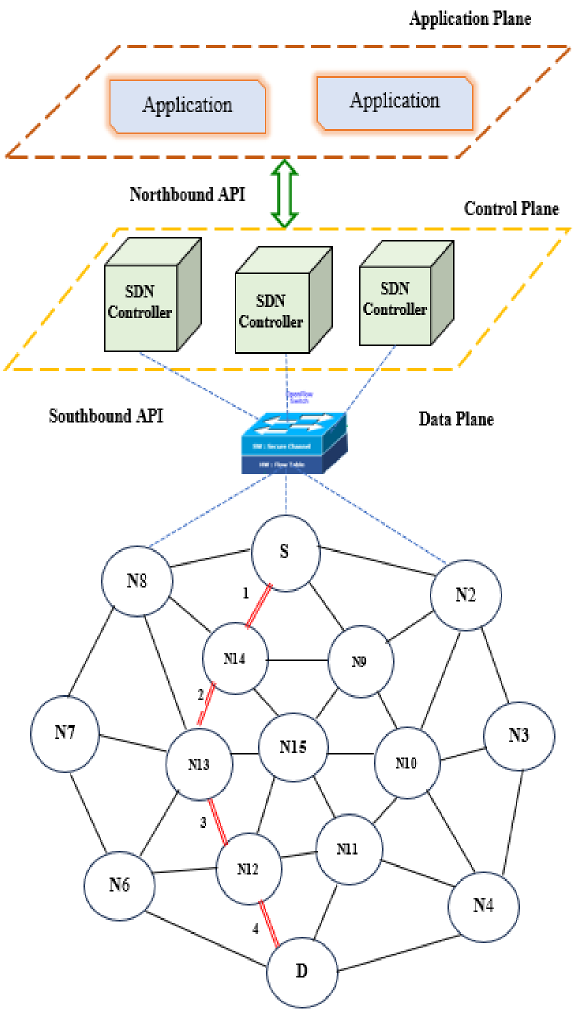

Figure 2. The physical network architecture of proposed EE-BBO.

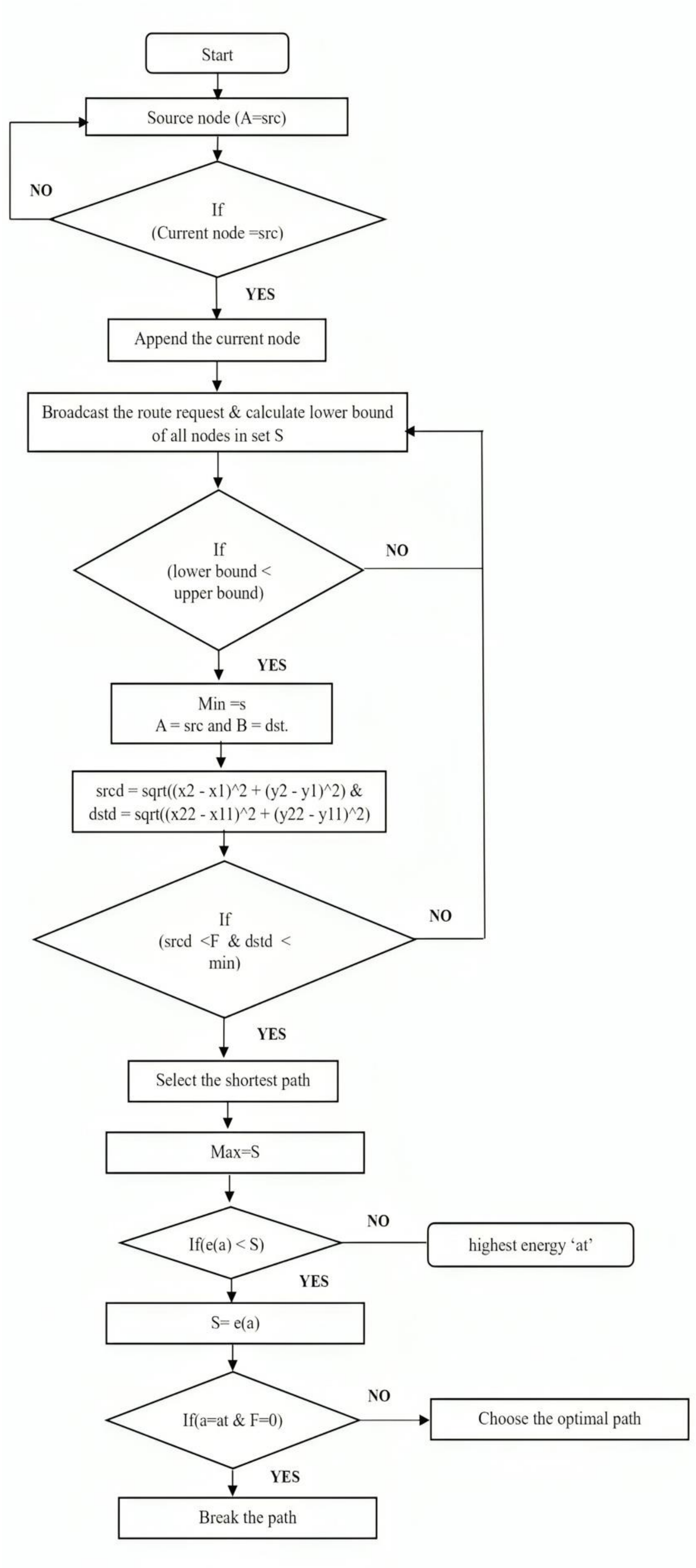

Figure 3. Process flow of EE-BBO algorithm

5. Results and Discussion

This segment of the research article assesses the performance of the EE-BBO algorithm in terms of energy consumption across different node densities, packet loss, and network lifetime. The results are compared with established methods such as TSR [25], HSLA [27], and KHMA [29]. The simulation environment employed Mininet 2.2 [34], Floodlight 1.2 as the SDN controller, and OpenFlow 1.3 [35] for communication. Detailed simulation parameters are provided in Table 2.

Table 2. Simulation setup

Variables	Measurements
Nodes	250
Links	164
Switch	42
Bandwidth	10 MB
Rate	20 bps-250 kbps
Delay	1-150 milliseconds
Initial energy	2J
Loss	0-15%
Packet size	512 Bytes
Threshold	5%
Sample	10
Simulation time	2000 (in seconds)
Packet range	35 packets/s

Figures 4(a)–4(e) present energy consumption trends for networks of varying sizes (50 to 250 nodes) using the proposed EE-BBO algorithm, compared to benchmarked approaches (KHMA, HSLA, TSR). The initial energy of each node is set to 2J, and the remaining energy is measured after completing packet transmissions and receptions. The x-axis represents simulation time, while the y-axis reflects energy consumption. The results indicate a significant reduction in energy consumption as the network size increases, demonstrating EE-BBO's scalability and efficiency in optimizing energy usage. This performance aligns with theoretical expectations, as the algorithm dynamically prioritizes energy-efficient paths.

The EE-BBO algorithm and KHMA show comparable performance for higher node densities (150–250 nodes) and extended simulation times, primarily due to the saturation of network resources and the convergence of energy-efficient routing strategies in both models. However, EE-BBO's energy-efficient branch and bound optimization framework dynamically adapts to changing network topologies and traffic conditions, unlike KHMA, which relies on static metaheuristic optimization. On average, EE-BBO achieves a 9% improvement in energy efficiency over KHMA, directly contributing to prolonged network lifetime. While EE-BBO and KHMA show similar performance in high-density networks, EE-BBO's deterministic optimization approach, lower computational overhead, and consistent energy efficiency gains across various configurations make it more suitable for dynamic and real-time SDN environments.

Figure 4(f) highlights the average energy usage per node, further emphasizing EE-BBO's energy-aware optimization. By employing branch and bound techniques, the algorithm identifies high-energy paths and reroutes data through efficient alternatives, balancing the load across nodes and avoiding overburdened routes. This approach minimizes energy depletion on critical nodes, reducing overall transmission energy while ensuring network longevity. Compared to KHMA, HSLA, and TSR, EE-BBO achieves total energy consumption reductions of 9%, 15%, and 19%, respectively, as shown in Figure 4(f). These results underscore the superior energy efficiency of EE-BBO, which is particularly crucial for real-world applications, such as IoT networks, where maintaining node energy is essential for extended operational lifetimes. By reducing energy consumption, EE-BBO supports more stable and long-lasting SDN deployments in large-scale, energy-constrained environments.

The proposed EE-BBO algorithm was also evaluated under varying traffic loads such as 50 Mbps (low), 100 Mbps (medium), and 150 Mbps (high). These loads were achieved by adjusting the packet generation rate in the simulation environment. As summarized in Figure 5, the algorithm maintains superior energy efficiency across all traffic conditions. Under low traffic conditions (50 Mbps), EE-BBO exhibited a 6% reduction in energy consumption. Under high traffic conditions (150 Mbps), it achieved similar reductions. Overall, the EE-BBO algorithm reduced energy consumption by 7%, 13%, and 18% compared to KHMA, HSLA, and TSR, respectively.

Packet loss is an influential metric for enhancing the performance of SDNs. It indicates the number of packets that are not successfully delivered against the total number of packets sent to the destination node. In an energy-constrained SDN environment, it is crucial to minimize packet loss which significantly avoids unnecessary packet retransmission in the network. The proposed EE-BBO algorithm reduces packet loss by instigating an energy-aware routing path that dynamically reroutes the traffic depending on the node's energy level and lower bound computations. By choosing a dynamically energy-aware routing path, the EE-BBO algorithm diminishes the congestion-based packet loss, minimizes packet retransmissions, and maximizes network performance.

Figure 6, shows the percentage of packet loss attained by the EE-BBO algorithm against the different number of nodes ranging from 50 to 250 nodes. The EE-BBO algorithm reduces the packet loss by 15%, 26%, and 28% compared to the existing approaches KHMA, HSLA, and RST respectively.

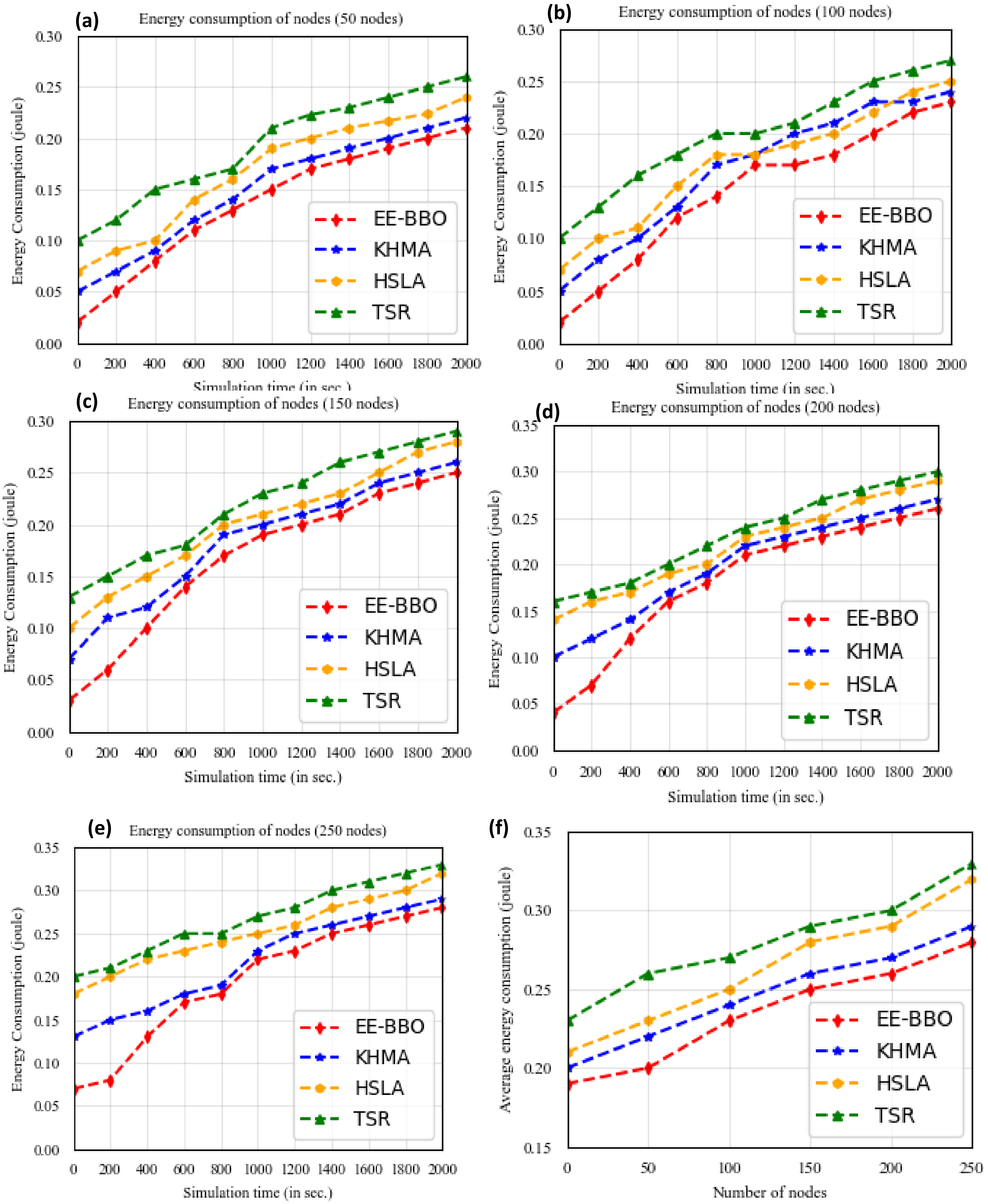

Figure 4. Average energy consumption vs different number of nodes

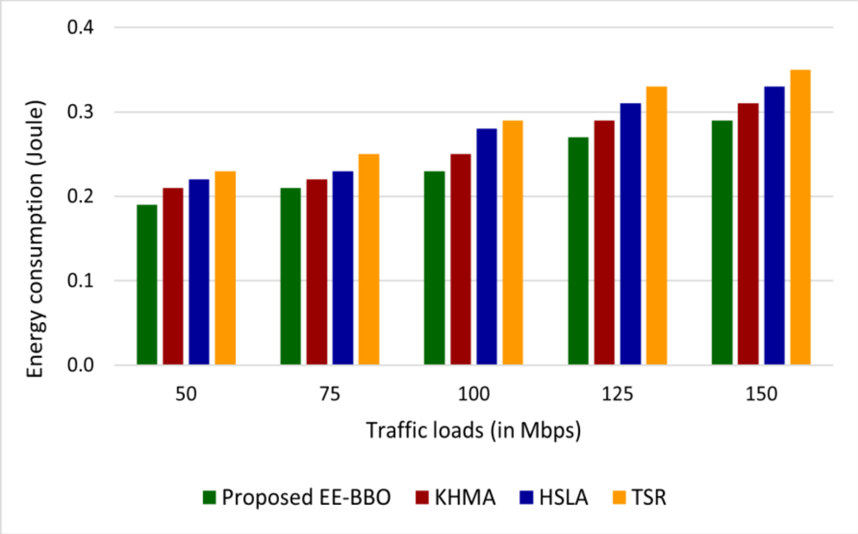

Figure 5. Energy consumption vs traffic load

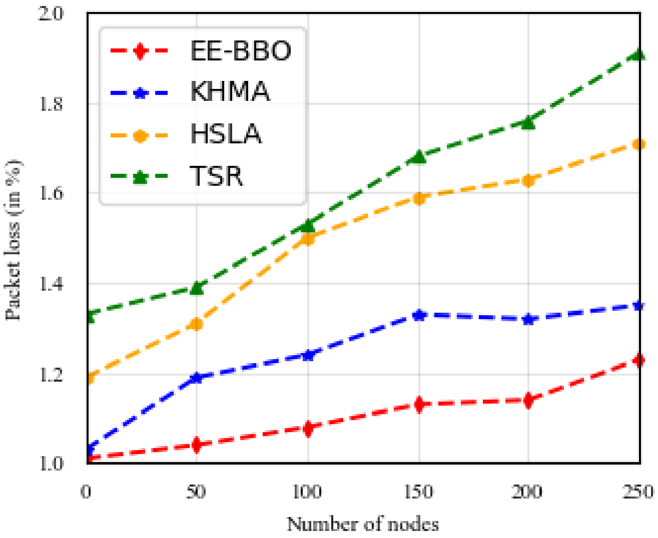

Figure 6. Packet loss (%)

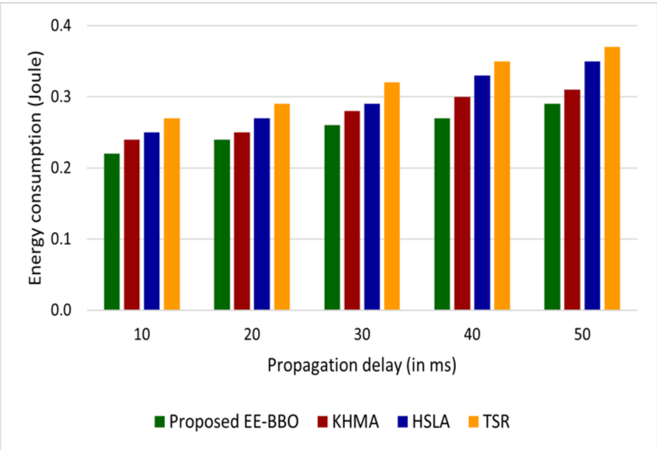

Figure 7. Energy Consumption vs propagation delay

.

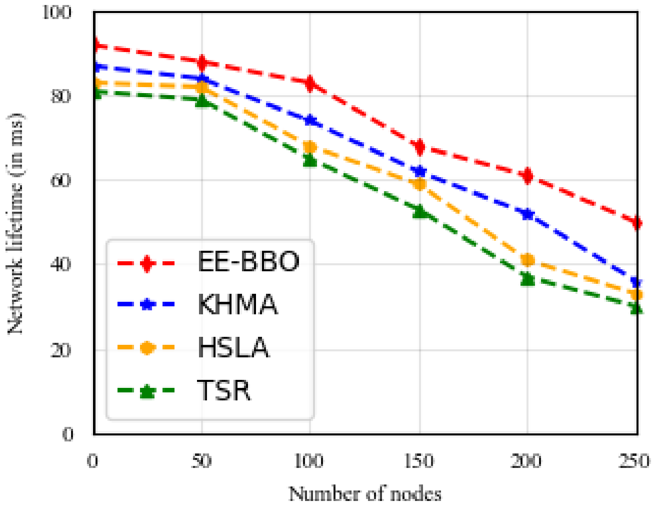

Figure 8. Network Lifetime

This reduction of packet loss minimizes the energy consumption for retransmission and fortifies the network performance and lifetime.

Propagation delay significantly affects SDN performance, particularly in large-scale networks where communication between nodes spans multiple hops. The EE-BBO algorithm was evaluated under varying propagation delays, ranging from 10 ms (low) to 50 ms (high), to simulate diverse network environments. Figure 7 shows that as propagation delay increases, energy consumption slightly rises due to prolonged routing decisions and packet retransmissions.

However, EE-BBO reduces energy consumption by 7%, 14%, and 20% compared to benchmark algorithms across all delay conditions.

Network lifetime, measured as the duration until the first node exhausts its energy, is a key indicator of sustainability in SDNs. The EE-BBO algorithm extends the network lifetime by leveraging energy-efficient branch and bound techniques that reduce energy consumption across the network. During the Initialization Phase, EE-BBO selects paths through nodes with higher energy, preventing early failures.

To balance the energy consumption throughout the network, the bound calculation phase gives priority to routes with the least traversal energy. The Bound Calculation Phase prioritizes routes with minimal traversal energy, balancing energy usage across the network. Then, the energy assessment phase dynamically reroutes the packets through higher-energy levels paths to evade the low-energy paths that cause the node diminution. The EE-BBO algorithm selects an energy-aware routing path, ensures reliable transmission between the source and destination in the path finalization phase, and reduces unforeseen path failures.

The network lifetime achieved by the proposed EE-BBO algorithm is presented in Figure 8. The EE-BBO algorithm implemented the branch and bound techniques which dynamically reroute the routing path according to the energy levels of the nodes. EE-BBO algorithm effectually stretches the network lifetime by complementing the energy levels throughout the network. The outcomes divulged that the EE-BBO algorithm elongates the network lifetime by 14%, 19%, and 25% compared to KHMA, HSLA, and TSR respectively.

6. Conclusion

The research article introduces a pioneering and energy-efficient branch-and-bound optimization (EE-BBO) algorithm intended to improve the SDN network life span, reliability, and overall performance of large-scale SDN environments by reducing energy consumption. The first significant contribution of EE-BBO involves choosing nodes and computing from the source to its broadcast by leveraging the plausibility of neighboring nodes. After that, it computes the lower bound to prioritize nodes and dynamically forward packets. Following this, the EE-BBO algorithm computes the total energy consumed during packet forwarding. At last, the proposed EE-BBO algorithm computes the total energy consumption of the routing process by accumulating the energy levels from node broadcasts, traversal, forwarding, and finalization. The EE-BBO algorithm successfully reduces total energy consumption throughout the routing process by featuring these strategies.

The findings revealed that the proposed EE-BBO algorithm outperforms the current benchmarked methods. The EE-BBO algorithm reduces packet loss by 15%, 26%, and 28% when compared to KHMA, HSLA, and TSR, respectively. Additionally, it reduces overall energy consumption by 9%, 15%, and 19%, and improves network lifetime by more than 14%, 19%, and 25% compared to KHMA, HSLA, and TSR, respectively. The effectiveness of the proposed EE-BBO algorithm in enhancing energy efficiency, reliability, and longevity makes it a promising solution for large-scale SDN environments. The proposed algorithm focuses on minimizing energy consumption in SDN networks rather than addressing other Quality of Service (QoS) parameters. Future work will aim to enhance the QoS parameters to improve network performance under dynamically changing conditions.

References

[1] S. Sharma, R.K. Bansal, S. Bansal, Issues and challenges in wireless sensor networks. in Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, IEEE, India. https://doi.org/10.1109/ICMIRA.2013.18

[2] D. Salman and Q. Yas, Challenges and Issues for Wireless Sensor Networks: A Survey, Journal of Global Scientific Research, 6 (2020) 1079-1097.

[3] B.A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys and Tutorials, 16(3), (2014) 1617–1634. https://doi.org/10.1109/SURV.2014.012214.00180

[4] D. Kreutz, F.M.V. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), (2015) 14–76. https://doi.org/10.1109/JPROC.2014.2371999

[5] Y. Gong, W. Huang, W. Wang, Y. Lei, A survey on software defined networking and its applications. Frontiers of Computer Science, 9, (2015) 827–845. https://doi.org/10.1007/s11704-015-3448-z

[6] J. Jiang, (2021) SDN Technology Analysis and Application Research. 7th Annual International Conference on Network and Information Systems for Computers (ICNISC), IEEE, China. https://doi.org/10.1109/ICNISC54316.2021.00014

[7] W. Xia, Y. Wen, C. H. Foh, D. Niyato, H. Xie, A Survey on Software-Defined Networking. IEEE Communications Surveys & Tutorials, 17(1), (2015) 27 – 51. https://doi.org/10.1109/COMST.2014.2330903

[8] S. Saraswat, V. Agarwal, H.P. Gupta, R. Mishra, A. Gupta, T. Dutta, Challenges and solutions in Software Defined Networking: A survey. Journal of Network and Computer Applications, 141, (2019) 23–58. https://doi.org/10.1016/j.jnca.2019.04.020

[9] S. Xu, X. W. Wang, M. Huang, Software-Defined Next-Generation Satellite Networks: Architecture, Challenges, and Solutions. IEEE Access, 6, (2018) 4027–4041. https://doi.org/10.1109/ACCESS.2018.2793237

[10] S. Rout, K. S. Sahoo, S.S. Patra, B. Sahoo, D. Puthal, Energy Efficiency in Software Defined Networking: A Survey. SN Computer Science, 2(4), (2021) 308. https://doi.org/10.1007/s42979-021-00659-9

[11] A.P. Patil, L.C.M. Hurali, Analysis of routing protocols for software-defined vehicular ad hoc networks. International Journal of Networking and Virtual Organisations, 24(2), (2021) 161-181. https://doi.org/10.1504/IJNVO.2021.114731

[12] M.F. Tuysuz, Z.K. Ankarali, D. Gözüpek, A survey on energy efficiency in software defined networks. Computer Networks, 113, (2017) 188-204. https://doi.org/10.1016/j.comnet.2016.12.012

[13] C. Shivakeshi, B. Sreepathi, A Review on Efficient Energy Consumption in Software-Defined Networking Using Routing Aware Protocols., 10(5), (2023) 478-490. https://doi.org/10.15379/ijmst.v10i5.2532

[14] A. Iqbal, U. Javed, S. Saleh, J. Kim, J. S. Alowibdi, bM. U. Ilyas, Analytical Modeling of End-to-End Delay in OpenFlow Based Networks. IEEE Access, 5, (2017) 6859–6871. https://doi.org/10.1109/ACCESS.2016.2636247

[15] N. Huin, M. Rifai, F. Giroire, D. Lopez Pacheco, G. Urvoy-Keller, J. Moulierac, Bringing Energy Aware Routing Closer to Reality with SDN Hybrid Networks. IEEE Transactions on Green Communications and Networking, 2(4), (2018) 1128–1139. https://doi.org/10.1109/TGCN.2018.2842123

[16] S. Vijaygokul, J. Dani Regan Vivek, B. Naresh Kumaar, N. Vignesh, Reduction of table flow occupancy and packet loss detection in SDN switch. Journal of Physics: Conference Series, Institute of Physics Publishing, 1362(1), (2019) 012051. https://doi.org/10.1088/1742-6596/1362/1/012051

[17] B.R. Al-Kaseem, Y. Al-Dunainawi, H.S. Al-Raweshidy, End-to-end delay enhancement in 6LoWPAN testbed using programmable network concepts. IEEE Internet of Things Journal, 6(2), (2019) 3070–3086. https://doi.org/10.1109/JIOT.2018.2879111

[18] T. Zhang, B. Liu, Exposing End-to-End Delay in Software-Defined Networking. International Journal of Reconfigurable Computing, (2019)(1), (2019) 7363901. https://doi.org/10.1155/2019/7363901

[19] L. El-Garoui, S. Pierre, S. Chamberland, A new sdn-based routing protocol for improving delay in smart city environments. Smart Cities, 3(3), (2020) 1004–1021. https://doi.org/10.3390/smartcities3030050

[20] M. Priyadarsini, S. Kumar, P. Bera, M.A. Rahman, An energy-efficient load distribution framework for SDN controllers. Computing, 102(9), (2020) 2073–2098. https://doi.org/10.1007/s00607-019-00751-2

[21] S. Torkzadeh, H. Soltanizadeh, A.A. Orouji, Energy-aware routing considering load balancing for SDN: a minimum graph-based Ant Colony Optimization. Cluster Computing, 24(3), (2021) 2293–2312. https://doi.org/10.1007/s10586-021-

03263-x

[22] Y. Zhao, X. Wang, Q. He, B. Yi, M. Huang, W. Cheng, Power-Efficient Software-Defined Data Center Network. IEEE Internet of Things Journal, 8(12), (2021) 10018–10033. https://doi.org/10.1109/JIOT.2020.3048524

[23] A. Akbar, M. Ibrar, M. A. Jan, A. K. Bashir, L. Wang, SDN-Enabled Adaptive and Reliable Communication in IoT-Fog Environment Using Machine Learning and Multiobjective Optimization. IEEE Internet of Things Journal, 8(5), (2021) 3057–3065. https://doi.org/10.1109/JIOT.2020.3038768

[24] N. Saha, S. Bera, S. Misra, Sway: Traffic-Aware QoS Routing in Software-Defined IoT. IEEE Transactions on Emerging Topics in Computing, 9(1), (2021) 390–401. https://doi.org/10.1109/TETC.2018.2847296

[25] Y. Wang, H. An, J. Ba, P. Yu, Y. Feng, Z. Wei, M. Kadoch, M. Cheriet, Energy-efficient method based on dynamic topology switching and reliability in SDNs. IEEE Transactions on Sustainable Computing, 7(2), (2021) 427-440. https://doi.org/10.1109/TSUSC.2021.3116325

[26] V.K.A. Reddy, R.B.K. Nagappasetty, Leveraging Software-Defined Networks for Load Balancing in Data Centre Networks using Linear Programming. International Journal of Computing, 22(3), (2023) 404–411. https://doi.org/10.47839/ijc.22.3.3237

[27] J. Galán-Jiménez, M. Polverini, F.G. Lavacca, J.L. Herrera, J. Berrocal, Joint energy efficiency and load balancing optimization in hybrid IP/SDN networks. Annales des Telecommunications/Annals of Telecommunications, 78(1–2), (2023) 13–31. https://doi.org/10.1007/s12243-022-00921-y

[28] R.F. Ghani, L. Al-Jobouri, Packet Loss Optimization in Router Forwarding Tasks Based on the Particle Swarm Algorithm. Electronics, 12(2), (2023) 462. https://doi.org/10.3390/electronics12020462

[29] M. Forghani, M. Soltanaghaei, F. Zamani Boroujeni, Dynamic optimization scheme for load balancing and energy efficiency in software-defined networks utilizing the krill herd meta-heuristic algorithm. Computers and Electrical Engineering, 114, (2024) 109057. https://doi.org/10.1016/j.compeleceng.2023.109057

[30] K. Deo, K. Chaudhary, M. Assaf, Adaptive quality of service for packet loss reduction using OpenFlow meters. PeerJ Computer Science, 10, (2024) e1848. https://doi.org/10.7717/peerj-cs.1848

[31] T.F. Oliveira, S. Xavier-De-souza, L.F. Silveira, Improving energy efficiency on SDN control-plane using multi-core controllers. Energies (Basel), 14(11), (2021) 3161. https://doi.org/10.3390/en14113161

[32] A. Nazari, F. Tavassolian, M. Abbasi, R. Mohammadi, P. Yaryab, An Intelligent SDN-Based Clustering Approach for Optimizing IoT Power Consumption in Smart Homes. Wireless Communications and Mobile Computing, 2022(1), (2022) 8783380. https://doi.org/10.1155/2022/8783380

[33] R. Mohammadi, S. Akleylek, A. Ghaffari, SDN-IoT: SDN-based efficient clustering scheme for IoT using improved Sailfish optimization algorithm. PeerJ Computer Science, 9, (2023) e1424. https://doi.org/10.7717/peerj-cs.1424

[34] F. Keti, S. Askar, (2015) Emulation of Software Defined Networks Using Mininet in Different Simulation Environments. in Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, IEEE, Malaysia. https://doi.org/10.1109/ISMS.2015.46

[35] F. Hu, Q. Hao, K. Bao, A survey on software-defined network and OpenFlow: From concept to implementation. IEEE Communications Surveys & Tutorials, 16(4), (2014) 2181 – 2206. https://doi.org/10.1109/COMST.2014.2326417

Authors Contribution Statement

S. Sathish - methodology design, investigation, formal analysis, data curation, validation, software implementation, resource acquisition, and writing of the original draft. J. Poongodi - , formal analysis, review, supervision and editing of the manuscript. G. Sudha - data curation, formal analysis, review, and editing of the manuscript. S. Sindhuja - resource management, formal analysis, review, and editing of the manuscript. All authors have read and approved the final version of the manuscript.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes

About the License