A Novel Energy-aware SDWSN Controller Placement Scheme

Bassey Isong  
Computer Science Dept.   
*North-West University*Mafikeng, South Africa  
isong.bassey@ieee.org

Adnan Abu-Mahfouz  
Council for Scientific and  
*Industrial Research (CSIR)*Pretoria, South Africa  
a.abumahfouz@ieee.org

Nosipho Dladlu  
Computer Science Dept.   
*North-West University*Mafikeng, South Africa  
nosipho.dldlu@nwu.ac.za

RRS Molose  
Computer Science Dept.   
*North-West University*Mafikeng, South Africa  
moloserrs@gmail.com

*Abstract*—This paper presents a novel algorithm for placing multiple controllers in a software defined wireless sensor network (SDWSN). The algorithm is designed to address scalability and energy efficiency issues in the network. We propose a hybrid algorithm that utilises an automatic particle swarm algorithm and k-means algorithm with TEEN protocol to reduce the energy consumption of sensor nodes while minimizing the propagation latency between controllers and sensor nodes in the SDWSN environment. The proposed techniques are evaluated based on network latency and lifetime in a distributed network condition. The results obtained show the effectiveness of the approach in minimizing network delay and maximizing energy efficiency. However, the proposed scheme is open to significant improvement for real-work SDWSN network deployment.

Keywords—SDWSN, Latency, Energy efficiency, TEEN, PSO.

# Introduction

Wireless Sensor Networks (WSNs) is an infrastructure-less wireless networks with numerous sensor nodes that are small, low-powered equipped with sensing, computation and communication capabilities [1, 2]. The sensor nodes are deployed in an ad-hoc manner and collaborate to monitor and gather information from the surrounding environment. WSNs have widespread applications such as the Internet of Things (IoT), healthcare, smart cities, surveillance, environmental sensing, healthcare, agriculture, industrial automation, landslide detection etc. However, it is faced with several challenges such as security, reliability, scalability, limited power and energy, limited processing, and storage capability and so on. Several techniques have been introduced to address these challenges such as routing and medium access control (MAC) protocols [1] and software-defined networking (SDN) [2].

SDN is a computing technology that decouples the control plane from the data or forwarding plane in a network. This allows network administrators to centrally manage and control the network, with more flexibility, programmability, scalability, and innovation. The SDN controller is a critical component in the SDN architecture which manages and orchestrates the network resources, implements network policies, and handles the control-data planes communication via the OpenFlow protocol [3, 4]. SDN is widely applied to different networks such as Optics networks, wireless networks,

WSNs, cloud data centres, etc. Its application in WSN results in the software-defined wireless sensor network (SDWSN) model. SDWSN applies SDN principles to WSN to enable efficient resource allocation, dynamic routing, fine-grained control over network operations, flexibility, programmability, and adaptability [2, 4, 5]. SDWSN has a significant impact on the ‘smart things’ world or IoT which involves dynamic networks and where several application requirements have increased tremendously. For instance, the increase in network traffic results in several challenges such as energy conservation as battery-powered sensor nodes run at a limited power level. Moreover, like the SDN and the traditional WSN, SDWSN is not isolated from challenges such as scalability, security, reliability, and network management [2, 3, 6, 7].

To address the scalability issue in the SDWSN, several strategies have been proposed and developed including multiple controllers which are also used to enhance the reliability and resiliency of the network. However, the use of multiple controllers introduces a new challenge known as the controller placement problem (CPP) [4, 7, 8]. CPP is a major design choice of the SDN-SDWSN control plane due to its effect on a wide range of network requirements or factors such as minimum latency, maximum reliability, minimum energy consumption, maximum load balance, network topology, controllers’ capabilities, and traffic flows [1, 9]. It is deemed a challenging task or an NP-hard problem because it involves determining the number of controllers, the topology of controllers placement and the assignment of controllers and switches which are critical to meeting the important network requirements[4]. For instance, minimized latency relies on the speed of controllers and switches interaction, and between controllers to handle the load in the network which in turn relies on controllers' location and allocation which is crucial to network scalability, reliability and load balancing [4]. Thus, the deployment of a distributed controller is required as a fast recovery mechanism to increase reliability and resilience [10].

Recently, the management and processing of a network have been greatly improved by modern technology [11, 12]. Thus, to combat these trends, effective, dynamic strategies are essential. Though several CPP approaches exist from the perspective of SDN [13-16] using several algorithms such as heuristics, metaheuristics, machine learning (ML), game theory, mathematical programming, bio-inspired, etc., an efficient and dynamic CPP algorithm is still lacking [4, 8]. Only a few CPP solutions have been dedicated to SDWSN, even with WSN and IoT advances' inherent difficulties [17]. One of the important trends in solving or discovering the optimal solution in a short amount of time while offering good quality of service (QoS) and minimal resource utilization is an efficient and dynamic algorithm [4, 8]. Therefore, this paper proposes a novel efficient controller placement technique that minimizes latency and maximizes energy efficiency and reliability. It utilizes multi-objectives to enhance network performance and QoS. These metrics comprise optimum controller location and allocation, reliability, and energy efficiency aware load balancing.

The remaining parts of the paper are structured as follows: Section IV presents the CPP background, Section III presents some of the related works, Section IV presents the proposed CPP approach and Section V presents the evaluation. Sections VI and VII present the paper's discussion and conclusion.

# Controller Placement Problem

CPP is the challenge associated with determining the optimal location of SDN-SDWSN controllers within the network to meet important network requirements. This challenge is brought about by using multiple controllers to address the single point of failure syndrome of a centralized controller. A network with the specified nodes and links, its topology is modelled as an undirected graph, G = (V, E) where V is the set controllers or switches and E is the connection between them [3, 4]. The distance between v and s is denoted as d(v,s), the shortest path between v and s where, v, s ∈ V. CPP in the SDN-SDWSN is aimed at achieving efficient network management, control, and communication while minimizing latency, maximize reliability, overheads and resource utilization [4, 7]. Several CPP approaches exist [4], while most approaches utilized single objectives such as latency, reliability, and cost, several other studies utilized multi-objectives [3, 4] to enhance the optimal placement.

Some of the performance objectives critical for achieving CPP in SDWSN are as follows:

## Latency-aware

The placement of controllers can significantly affect network performance. For instance, the proximity of the controller to the network devices impacts the latency of communication. Thus, controllers should be placed strategically to achieve minimized network latency and enhanced response time for control messages [19]. Two forms of network latencies are considered in the SDN-SDWSN during CPP: propagation latency (S2C communication) and controller processing latency(C2C communication) [19] [4, 7, 8].

Propagation latency involves how face C2S or S2C communication can be which is dependent on the distance between them, topology, etc. During CPP, the goal is to achieve minimum propagation latency and avoid packed loss, buffer overflow, or flow setup delay [4, 19, 20]. Minimum propagation latency is critical for network reliability and QoS. It can be measured either by the average case or worst-case. While average-case latency is the average time taken for the message to be received from the switch to its assigned controller, the worst-case latency is the maximum time taken [19, 20]. Equ. 1 and 2 present the average and worst-case latencies.

Processing latency is the duration of a controller to process a message received from the switches/sensors or other controllers. This depends on the controller capacity, load and scheduling policy [19, 20]. A good controller placement that minimizes processing latency is important and can enhance reliability and QoS [4, 7, 8]. The average C2C processing latency is given in Eq. 3:

The average round-trip time for a packet to transmit from one sensor node to another in the network is known as wireless average latency. This constitutes the key performance metric on which the efficiency and responsiveness of wireless communication are measured [12]. It depends on several factors such as the d(s,s), bandwidth, interference, processing and queuing delays and packet size [20]. The goal is to minimize it since the lower it is, the faster the data transmission and the more network response. The average latency for SDWSN CPP involving sensor nodes is presented in Eq..4 [21]:

Where *t(l)* is the sum of the total time related to transmission, processing, propagation and queuing while N is the number of packets.

## Energy-aware

Efficient utilization of constraint resources is critical to the optimal survival of WSNs. Thus, it is vital to maximize network lifetime via energy-efficient techniques [22]. This can be achieved by maximizing various objectives based on coverage, connectivity, the total number of alive nodes and the underlying application [22, 23]. Other important performance optimizations are cost, scalability, fault tolerance/reliability, load balancing, network topology awareness, and so on, which are critical in achieving a practical CPP solution.

# Related Works

Some of the current research on CPP solutions in the SDN-SDWSN realm is presented in this section. We focus on latency-aware and energy-aware approaches.

## Latency-aware

Some of the SDN-based CPP schemes that considered network latency and reliability are discussed: Sahoo *et al.* [24] developed the Firefly algorithm (FFA) and Particle Swarm Optimization (PSO) as meta-heuristic algorithms to address a multi-objective combinatorial optimization problem. The objectives were S2C or C2S latency and C2C latency and survivability as objectives to achieve minimum latency and ensure reliability in terms of connectivity. Similarly, Liao *et al.* [25] developed a density cluster-based methodology that avoids the flaw of heuristic algorithms that get stuck in locally optimum solutions. Also, Killi *et al.* [26] suggested a Game theory-based network partitioning algorithm that is collaborative and is based on propagation delays in the worst-case and average scenarios. W*ang et al..* [27] also suggested a clustering technique based on the k-means algorithm to minimize maximum latency between the nodes and specific centroid of a subnetwork, while authors in [28] minimized the C2S latency and C2C latency utilizing a heuristic approach based on the Dijkstra algorithm and k-means. In the same vein, authors in [29] considered the average latency based on balanced cost-latency placement to improve the QoS using several algorithms while Wang *et al.* [30] created a clustering network partition mechanism to reduce queueing and end-to-end latency.

In terms of meta-heuristic-based CPP, Radam *et al..* [13] utilized various meta-heuristic approaches to achieve optimal multi-controller placement including an FFA and a hybrid algorithm: harmonic search algorithm and particle swarm optimization algorithm (HSA-PSO). The method demonstrated success in the following metrics: propagation latency, round trip time, time session, delay, dependability, and throughput. Similarly, authors in [14] also suggested another revised meta-heuristic algorithm which achieved the same set of network performance objectives as in [13] using simulated annealing for multi-controllers in SDN and evaluated with the HSA-PSO. In another study, Kim [15] proposed an artificial bee colony algorithm with an effective and optimal controller selection scheme to minimize propagation latency and communication with improvement seen in throughput and reliability. The naked Mole-Rat algorithm is another metaheuristic algorithm proposed by Sapkota *et al.* [16], while authors in [16] proposed evolutionary algorithms to achieve optimal controller placement by minimizing latency as their network performance objective.

## Energy-aware

This subsection highlights some of the energy aware CPP schemes. Gante *et al*. [31] suggested a base station with five levels for the SDWSN environment: physical, media, access, NOS, middleware, and application. The controller, which is a piece of middleware with mapping capabilities, uses monitoring messages to gather data on sensor node energy levels and conserve energy. It uses a mapping function to process the data and is saved in a mapping information module so that it can be used by the application layer to find and create precise sensed data for the upkeep of the sensor node location data [31, 32]. Moreover, Huang *et al..* [33] developed a machine learning method to increase energy efficiency while Zeng *et al.* [34] with the aid of mathematical integer linear programming, a proposed online method with local optimization to improve the energy usage of a multitasking SDWSN has been made. In recent papers seen in [35-37], the authors utilized modern approaches including system designs from hardware devices, network architectural designs and hybrid optimized algorithms. Energy efficiency with other performance objectives including control overhead [37], fitness value and stability period [35] were evaluated with results showing satisfactory evidence with outperformance of existing methods.

While the above studies highlighted some of the CPP approaches for SDN and SDWSN in terms of delay and energy efficiency as the major objectives, from the perspective of SDWSN, an effective and dynamic CPP algorithm is still missing. We, therefore, address this challenge in this paper.

# Proposed Efficient Controller Placement

This section proposed an efficient energy-aware CPP algorithm (EE-CPA). EE-CPA is achieved with two algorithms and the performance objectives considered are the latency, cost, and energy consumption. The proposed technique is in two phases: the latency-aware and the energy-aware CPP algorithms which are based on the automatic PSO algorithm [17] and the energy-efficient TEEN protocol. Equ. 5 is the formulated problem that the proposed algorithm solved. As shown, the goal is to minimize the sum of the cost of deployment in terms of the number of controllers and switches assignments, the energy consumption, and the sum of average latency. The cost minimization objective is based on the CAPEX of the controller deployment. This is seen as optimizing location controller deployment before the physical deployment of controllers in the network. Additionally, the network lifespan in equation 5 is calculated from the number of nodes that are still alive and have energy remaining after a particular amount of time or iteration time. The evidence is asserted in the results and analysis section of the paper.

Where W is the wireless latency, S2C is the latency from switch to controller or controller to switch communication and EC is energy consumption.

## Energy and Latency-aware CPP Algorithm

In this algorithm, we employed the clustering strategy based on automatic PSO and k-means algorithms to achieve minimal latency based on Euclidean Distance. In this case, an undirected G of sensor nodes within a randomly positioned n\*n area network constitutes the algorithm's first input. As depicted in Fig. 2, these nodes are grouped based on the automatic PSO method. The network is then divided into *k* clusters, the centres of which are the local controllers, and the *k* value is determined statically. Since there are initially no controllers in the network, the coordinates of these centres are returned once the clustering has been carried out. Choosing a controller's optimal placement before putting it in a network, minimizes the cost while taking the cost objective into account. The TEEN protocol was employed, an energy-efficient routing technology, to achieve this. A set of k clusters from the first phase is accepted by the procedure as well as the designated controllers as a clustering technique. With the assignment and/or selection of cluster heads (CH) in an isolated WSN cluster with a dedicated controller (local controller), these SDWSN controllers allow each local controller to independently execute an energy-efficient protocol, TEEN. Each cluster's dedicated CH may be updated while the TEEN protocol is running, up until the termination of the protocol after n iterations. To prevent data loss, data from the relevant sensor node is forwarded to the current CH update if a node with depleted energy is exposed in the cluster. Fig. 1 presents the proposed algorithm.

|  |
| --- |
| **Efficient Energy-aware Controller Placement Algorithm** |
| **Input:** G = (V, E) |
| **Output:** G (CPP) = (, E) |
| **Steps:** |
| 1. **Initialization** 2. PSO-kmeans ( , ) // *hybrid function of PSO-kmeans*   // *function accepts the number of nodes in the network topology and clusters of choice*   1. number of sensor nodes,  number of clusters 2. **If** =! 0 **then**   **return**  **else**  PSO-kmeans ( , ) // *execute/run function.*  **return** PSO-kmeans () // *output the centroids of each cluster.*   1. = PSO-kmeans () 2. = (, E) // *New (x,y) topology of optimized placement location* |
| 1. TEEN ( , ) // *Routing Protocol function of Teen* 2. // *function accepts the number of SN’s of a cluster and Sink node as the centroid(s) or local controller from Phase I* 3. **while** iteration =! 0 **do** 4. TEEN (*iteration = n*) *// teen routing protocol execution for n rounds*   **If** *energy () = 0* then  **If** *Data\_Queued() == true* **then**  **return** *push* *Data\_Queued()*  **else** terminate CH  **else**  insert (input ( = (, E)))  **end**   1. V = // *Number of* *alive nodes* 2. = (V, E) 3. **end** // *termination of the algorithm (phase II)* |

1. EE-CP algorithm

# Evaluation

## Simulation Setup

The performance of the EE-CPA is evaluated using simulations. We used the benchmarking technique where a  lightweight tool to gather data about the controller's network connection is placed in an emulated controller. The evaluation metrics are the network lifetime (i.e., the time when all sensors are active or alive) and the propagation latency or delay (i.e., the difference in data transmission sending and receiving times) as network performance objectives. The benchmarks were executed in a MacBook Air, Apple M1 chip with 8 cores, 8GB of memory, and 250GB of flash storage. TABLE I provides an overview of the simulated parameters.

Since the control plane is utilized via a central/master controller, here is how the data was collected from the control plane. Before deploying the local controllers, we first determined the number of clusters or aggregates in the network. This lets us decide in advance how to minimize costs. In this case, PSO was employed for clustering based on K-means to obtain the centroids. The geo-coordinates of the centroids were those where local controllers may be positioned. To get the optimal placement location for the controllers, a distributed routing method is used. We used local controllers to carry out the TEEN protocol for their clusters, as opposed to conventional WSN routing, where a sink node serves as the primary network master for routing protocols. The local controllers communicated directly with the sink as the master controller, thus, communicating with each other.

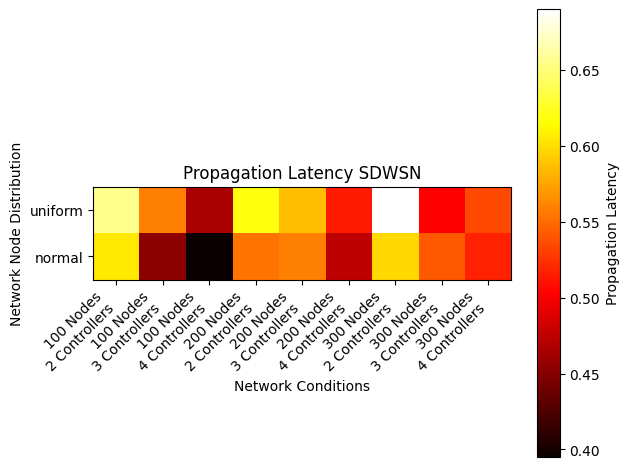
1. Simulation parameters

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| SDN CPP approach | PSO algorithm, K-Means clustering |
| WSN Protocol | TEEN protocol |
| Simulator | Python: Jupyter and NetworkX  Matlab (Python kernel) |
| Patch | SD-WSN |
| Simulation Area | 100\*100 meters |
| Simulation Time | 100-3000 rounds/iteration |
| Topology | Palmetto (The Internet Topology Zoo) |
| Per-node initial energy | 100 J |
| The total number of local controller (s) | 2-4 |
| Number of Sensor(s) | 100-300 |

## Results and Analysis

The outcomes of the simulations are presented in this subsection. The simulations were conducted based on the parameters highlighted in Table 3 and data was collected accordingly. The hybrid CPP scheme utilized the optimized PSO-k-means algorithm based on clustering. These partitions provided the opportunity for an efficient allocation or number of controllers while minimum propagation latency is achieved via the longitude and latitude location of the controllers. We designed the network condition using different numbers of controllers between 2 and 4 with nodes between 100 and 300. In addition, the distribution of the topology was based on “uniform” and “normal” distribution. Random data points were produced using the uniform distribution, which was constructed in a normal distribution with a mean and standard deviation of 0.5 and 0.2, respectively.

### A picture containing text, screenshot, line, diagram Description automatically generatedPropagation Latency: In Phase I, spatially distributed SDWSN are implemented using the PSO-based K-means algorithm. To reduce network propagation delay, the algorithm optimizes the placement of controller positions. The strategy optimizes the placement of the controller locations to lower the maximum propagation delay across all data points in the network. The Euclidean distance between data points and the positions of their closest controllers is used to calculate the propagation latency.

 Existing cluster-based approaches [27, 30] have shown a reduction in propagation latency. The methods have shown at least a reduction of latency ranging from 6 ms to 10 ms [27, 30]. Utilizing meta-heuristic techniques to improve network performance has improved in recent years. The proposed SA-MCSDN outperforms most existing approaches with maximum iterations of 100 and a reduction of latency by 5 ms [14]. However, all these approaches do make use of methods our proposed scheme utilized. Thus, based on the results and evidence in Fig. 2, the approach still outperforms all these approaches as the maximum propagation latency it recorded is 0.75 ms with a maximum of 300 nodes in a normal or uniform network. The most performing test case is 4 controllers with 100 nodes by recording an optimal 0.35 ms reduction in latency which is similar to the CPP scheme in [38].

1. Network latency

### Network lifetime: The effectiveness of the TEEN protocol is assessed in the proposed EE-CPA based on energy usage, network lifetime, and data transmission efficiency. Each simulation round's broadcast to the base station and cluster heads is tracked by the variables Packets\_to \_Base station and Packets\_to\_CH, respectively. These metrics evaluate both the algorithm’s and the network’s performance in transmitting data. For choosing CHs, individual node energy levels are stored throughout each simulation round. The number of dead nodes and the network’s overall energy levels are taken into account when estimating the network lifespan. Fig. 3 presents the results of cluster 1 against the number of dead nodes. Clusters 2 to 4 are not shown here since they don’t differ from Cluster 1. As shown, there are no dead nodes due to the execution of a lower round/iteration count. However, Fig. 4 depicts the level of energy in each cluster where the colour scheme presents a decline in energy levels from ~0.5 to ~0.48 J. This is a slight decrease considering the size of the topology and the individual cluster.

1. Network lifetime

A picture containing text, screenshot, diagram, colorfulness

Description automatically generated

1. Energy level of sensor nodes for clusters 1-4 from 0 to 50 rounds

Accordingly, when we increased the iteration/round time from 50 to 3000 under the same network condition, there was an increase in the number of dead nodes as shown in Fig. 5 for cluster 1. Consequently, the energy consumption levels increased from ~0.480 J as shown in Fig. 6.

A picture containing text, screenshot, line, diagram

Description automatically generated

1. Network lifetime

A picture containing screenshot, text, diagram, colorfulness

Description automatically generated

1. Energy level of sensor nodes for clusters 1-4 from 50 to 300 rounds

# Discussion

This paper proposed an EE-CPA which is a hybrid scheme of PSO-k-means algorithms to aggregate the topology to establish a hierarchical architecture for SDWSN by determining the ideal number and location of controllers. The goal was to minimize propagation latency and cost while maximizing energy efficiency. To assess the performance, simulations were conducted with a clustered controller placement network topology and the results obtained are shown in Fig. 2 to 6. The number of clusters concerning the total number of network nodes was one of the several network conditions employed. The cluster number is critical to performing the optimal controller number and location placement leading to achieving minimum propagation latency. The results show that the proposed EE-CPA was able to effectively minimize both propagation latency and energy consumption in the SDWSN network.

These results corroborated with existing works such as [31, 35] where there was a significant reduction in energy consumption. Most SDWSN centralized algorithms or approaches are still in their infant phase. However, the most recent work [35] a very similar results to the proposed EE-CP algorithm. It showed effectiveness in network lifetime by sensor nodes surviving longer, as seen in Fig. 5, the number of dead nodes started increasing after the 1000 iteration similar to [35]. This is the case as the algorithm used is like the one used in EE-CP, PSO, however, the proposed approach used a cluster-based hybrid PSO-k-means and utilized the TEEN protocol for energy efficiency. The approach used in this paper is not random, but a tree-based cluster-based approach and TEEN protocol were used they are energy-aware approaches. They were tested and evaluated to support the use of these methods in the proposed EE-CP [39, 40]. However, despite the results, the proposed approach is still open for great improvement with consideration of comparing the approach to other central SDWSN algorithms and the use of other performance metrics like security, single point of failure and load balancing to address overhead.

Some of the directions that potential researchers might consider in future to support the improvement of this approach are as follows:

### Implementation tools: There are few simulation tools available for SDWSN and are more centred on creating frameworks than evaluating algorithms. The inability of academics to efficiently design and install SDWSN-based networks limits their ability to contribute to research. Although there has been less focus on centralized algorithms in these evaluations, it is critical to investigate tools with controllers and user-friendly interfaces, such as NS2/NS3, OMNeT++, Mininet, MATLAB, and Cooja, to address this issue [17, 41-43].

### Topology discovery protocol: The study concentrated on stationary sensor nodes. Thus, we suggest more research on mobility or dynamic topology applications. Dynamic topology calls for effective topology discovery (TD) protocols that make use of neighbourhood and southbound protocols already in place to monitor topology changes. Future work should consider creating a cost/performance CPP TD protocol that addresses connectivity problems and security flaws brought on by network gaps.

### Northbound and Southbound protocols: The shortcomings of control and data plane protocols, especially more established ones like OpenFlow and Sensor OpenFlow do not take mobility and sensor heterogeneity into account, hence difficult to design algorithms for SDWSNs. Because of the controller's indirect connections, information gathering from sensor nodes is restricted and must go through other nodes. These problems can be resolved by using a self-routing protocol, such as a MAC protocol and a neural network machine learning-based algorithm, to improve network connectivity, reduce delays, and increase network longevity.

### Security challenges: Security problems in networks, whether wired or wireless, can be exacerbated by technologies like virtualization and routing protocols intended to boost efficiency. Therefore, in the future, it is important to consider creating methods that employ machine learning to recognize and isolate risks while connecting to a sensor, ensuring topology recovery, and preventing data loss in the network.

# Conclusion

This paper proposed and presented an efficient energy-aware controller placement algorithm that minimizes propagation latency, cost and energy consumption while indirectly maximizing reliability. This is achieved by designing an EE-CPA based on a hybrid scheme of PSO-k-means algorithms and the TEEN protocol. The proposed technique was evaluated via a series of simulations based on different network conditions in a Python-simulated environment, data was collected, analyzed, and presented. The results obtained revealed the proposed technique achieved both propagation latency and energy efficiency at a minimum through a distributed approach using clustering. We also compared the results of other studies [14, 27, 30, 31, 35, 38]. Although our algorithm is efficient, it is open to thorough enhancement due to several challenges encountered. Therefore, we urge that future work should be channelled to improve key areas like security, reliability, and comparisons with other existing protocols and implementation tools in the SDWSN realm.

##### Acknowledgment

This research was supported by the UDSC, the Department of Computer Science at the North-West University Mafikeng campus and the Council for Scientific and Industrial Research (CSIR).

##### References

1. K. S. Adu-Manu, F. Engmann, G. Sarfo-Kantanka, G. E. Baiden, and B. A. Dulemordzi, "WSN Protocols and Security Challenges for Environmental Monitoring Applications: A Survey," *Journal of Sensors,* vol. 2022.
2. S. Bera, S. Misra and A. V. Vasilakos, "Software-Defined Networking for Internet of Things: A Survey," in *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994-2008, Dec. 2017.
3. B. Isong, R. R. S. Molose, A. M. Abu-Mahfouz and N. Dladlu, "Comprehensive Review of SDN Controller Placement Strategies," in IEEE Access, vol. 8, pp. 170070-170092, 2020, doi: 10.1109/ACCESS.2020.3023974.
4. J. Lu, Z. Zhang, T. Hu, P. Yi, and J. Lan, "A Survey of Controller Placement Problem in Software-defined Networking," *IEEE Access,* vol. 7, pp. 24290-24307, 2019.
5. T. Kgogo, B. Isong, F. Lugayizi and A. M. Abu-Mahfouz, "*A Survey of Resource Allocation and Controller Placement Problem in SDN-SDWSN*," 2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Windhoek, Namibia, 2021, pp. 1-8, doi: 10.1109/IMITEC52926.2021.9714659.
6. Y. Zhang, L. Cui, W. Wang, and Y. Zhang."A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications,* vol. 103, pp. 101-118, 2018.
7. B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proceedings of the first workshop on Hot topics in software defined networks*, 2012: ACM, pp. 7-12.
8. G. Wang, Y. Zhao, J. Huang, and W. Wang, "The Controller Placement Problem in Software Defined Networking: A Survey," *IEEE Network,* vol. 31, no. 5, pp. 21-27, 2017, doi: 10.1109/MNET.2017.1600182.
9. M. Ndiaye, G. P. Hancke, and A. M. Abu-Mahfouz,. "Software Defined Networking for Improved Wireless Sensor Network Management: A Survey" *Sensors,* vol. 17, no. 5, p. 1031, 2017.. https://doi.org/10.3390/s17051031.
10. T. Hirayama, T. Miyazawa, H. Furukawa, and H. Harai, "Reconstruction of control plane of distributed SDN against large-scale disruption and restoration," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2017: IEEE, pp. 253-258.
11. D. S. Rana, S. A. Dhondiyal, and S. K. Chamoli, "Software Defined Networking (SDN) Challenges, Issues and Solution," 2019.
12. I. Deva Priya and S. Silas, "A Survey on Research Challenges and Applications in Empowering the SDN-Based Internet of Things," Singapore, 2019: Springer Singapore, in Advances in Big Data and Cloud Computing, pp. 457-467.
13. N. S. Radam, S. T. F. Al-Janabi, and K. S. Jasim, "Multi-controllers placement optimization in sdn by the hybrid hsa-pso algorithm," *Computers,* vol. 11, no. 7, p. 111, 2022.
14. N. S. Radam, S. T. F. Al-Janabi, and K. S. Jasim, "Using Metaheuristics (SA-MCSDN) Optimized for Multi-Controller Placement in Software-Defined Networking," *Future Internet,* vol. 15, no. 1, p. 39, 2023.
15. K. T. Kim, "Optimal Controller Selection Scheme Using Artificial Bee Colony and Apriori Algorithms in SDN," in *International Conference on Knowledge Management in Organizations*, 2023: Springer, pp. 347-359.
16. A. Sapkota, B. B. R. Dawadi, and C. S. R. Joshi, "Multi-Controller Placement Optimization Using Naked Mole-Rat Algorithm over Software-Defined Networking Environment," *Journal of Computer Networks and Communications,* vol. 2022, 2022.
17. H. Mostafaei and M. Menth, "Software-defined wireless sensor networks: A survey," *Journal of Network and Computer Applications,* vol. 119, pp. 42-56, 2018.
18. A. K. Singh and S. Srivastava, "A survey and classification of controller placement problem in SDN," *International Journal of Network Management,* vol. 28, no. 3, p. e2018, 2018.
19. S. Mohanty, A. S. Shekhawat, B. Sahoo, H. K. Apat, and P. Khare, "Minimizing latency for controller placement problem in SDN," in *2021 19th OITS International Conference on Information Technology (OCIT)*, 2021: IEEE, pp. 393-398.
20. M. M. Kazemian and M. Mirabi, "Controller placement in software defined networks using multi-objective antlion algorithm," *The Journal of Supercomputing,* pp. 1-24, 2022.
21. A. Dvir, Y. Haddad, and A. Zilberman, "Wireless controller placement problem," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2018: IEEE, pp. 1-4.
22. R. Soua and P. Minet, "A survey on energy-efficient techniques in wireless sensor networks," in *2011 4th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011)*, 26-28 Oct. 2011 2011, pp. 1-9, doi: 10.1109/WMNC.2011.6097244.
23. I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN),* vol. 5, no. 1, pp. 1-39, 2009.
24. K. S. Sahoo, D. Puthal, M. S. Obaidat, A. Sarkar, S. K. Mishra, and B. Sahoo, "On the placement of controllers in software-defined-WAN using the meta-heuristic approach," *Journal of Systems and Software,* vol. 145, pp. 180-194, 2018.
25. J. Liao, H. Sun, J. Wang, Q. Qi, K. Li, and T. Li, "Density cluster-based approach for controller placement problem in large-scale software defined networking," *Computer Networks,* vol. 112, pp. 24-35, 2017.
26. B. P. R. Killi, E. A. Reddy, and S. V. Rao, "Cooperative game theory based network partitioning for controller placement in SDN," in *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, 2018: IEEE, pp. 105-112.
27. G. Wang, Y. Zhao, J. Huang, Q. Duan, and J. Li, "A K-means-based network partition algorithm for controller placement in software defined network," in *2016 IEEE International Conference on Communications (ICC)*, 2016: IEEE, pp. 1-6.
28. L. Zhu, R. Chai, and Q. Chen, "Control plane delay minimization based SDN controller placement scheme," in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2017: IEEE, pp. 1-6.
29. T. Li, Z. Gu, X. Lin, S. Li, and Q. Tan, "Approximation Algorithms for Controller Placement Problems in Software Defined Networks," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, 2018: IEEE, pp. 250-257.
30. G. Wang, Y. Zhao, J. Huang, and Y. Wu, "An effective approach to controller placement in software defined wide area networks," *IEEE Transactions on Network and Service Management,* vol. 15, no. 1, pp. 344-355, 2017.
31. A. De Gante, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in *2014 27th Biennial Symposium on Communications (QBSC)*, 2014: IEEE, pp. 71-75.
32. H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access,* vol. 5, pp. 1872-1899, 2017.
33. R. Huang, X. Chu, J. Zhang, and Y. H. Hu, "Energy-efficient monitoring in software defined wireless sensor networks using reinforcement learning: A prototype," *International Journal of Distributed Sensor Networks,* vol. 11, no. 10, p. 360428, 2015.
34. D. Zeng, P. Li, S. Guo, T. Miyazaki, J. Hu, and Y. Xiang, "Energy Minimization in Multi-Task Software-Defined Sensor Networks," *IEEE Transactions on Computers,* vol. 64, no. 11, pp. 3128-3139, 2015.
35. R. Ramteke, S. Singh, and A. Malik, "Optimized routing technique for IoT enabled software-defined heterogeneous WSNs using genetic mutation based PSO," *Computer Standards & Interfaces,* vol. 79, p. 103548, 2022.
36. S. Buzura, B. Iancu, V. Dadarlat, A. Peculea, and E. Cebuc, "Optimizations for energy efficiency in software-defined wireless sensor networks," *Sensors,* vol. 20, no. 17, p. 4779, 2020.
37. S. S. G. Shiny, S. S. Priya, and K. Murugan, "Control message quenching-based communication protocol for energy management in SDWSN," *IEEE Transactions on Network and Service Management,* vol. 19, no. 3, pp. 3188-3201, 2022.
38. J. Hemagowri and P. T. Selvan, "A Hybrid Evolutionary Algorithm of Optimized Controller Placement in SDN Environment," *Computer Assisted Methods in Engineering and Science,* 2023.
39. R. Molose, B. Isong, N. Dladlu, and A. Abu-Mahfouz, "Data Aggregation Schemes for Maximal Network Lifetime: Review," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 20-22 July 2022 2022, pp. 1-8, doi: 10.1109/ICECET55527.2022.9872947.
40. R. Molose, B. Isong, N. Dladlu, and A. M. Abu-Mahfouz, "Analysis of Hierarchical Cluster-based Energy-Aware Routing Protocols in WSNs for SDWSN Application," in *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 9-10 Dec. 2021 2021, pp. 1-8, doi: 10.1109/ICECET52533.2021.9698557.
41. A. Varga, "OMNeT++," in *Modeling and tools for network simulation*: Springer, 2010, pp. 35-59.
42. M. L. Rajaram, E. Kougianos, S. P. Mohanty, and U. Choppali, "Wireless sensor network simulation frameworks: A tutorial review: MATLAB/Simulink bests the rest," *IEEE Consumer Electronics Magazine,* vol. 5, no. 2, pp. 63-69, 2016.
43. J. Eriksson *et al..*, "COOJA/MSPSim: interoperability testing for wireless sensor networks," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009, pp. 1-7.