A Hybrid DEWO Approach for Enhancing QoS in Software-Defined Networks through Optimized Controller Placement

Surendra Kumar Keshari[1], Nitish Pathak[2]

[1]Lincoln Global Postdoctoral Research (LGPR) Program, Lincoln University College, Petaling Jaya 47301, Selangor Darul Ehsan, Malaysia

[1]Department of Information Technology, Krishna Institute of Engineering & Technology (KIET), Ghaziabad, Delhi-NCR, Uttar Pradesh, India

[2]Department of CSE, Bhagwan Parshuram Institute of Technology, GGSIPU, New Delhi, India

[1]pdf.surendrakeshari@lincoln.edu.my , [1]surendra.keshari@gmail.com , [2]nitishpathak@bpitindia.com,

Abstract

The new network concepts (like SDNs) decouple the plane of control logic from the plane of hardware data. Software-Defined Networking (SDN) orchestrates networked intelligent devices in a centralized fashion. Smart devices enabled with IoT collect data in real time from devices on the network. The Smart City is an evolving IoT application where the urban environment is automatically controlled without human involvement. Smart city actuators and sensors and other rapidly proliferating IoT devices generate huge amounts of data and increase network traffic. In smart cities, the Quality of Service (QoS) can be guaranteed by Software-Defined Networking (SDN) in IoT networks, which can improve the user experience. Traditional IoT networks have latency, security, and reliability issues. These challenges are mitigated and performance is improved in SDN enabled IoTs networks. Smart city networking needs number of controllers to monitor IoT devices. Software Defined IoT Networks require several controllers. Various metaheuristic strategies are used to solve controller placement problems (CPP), but premature conversions adversely affect performance due to total controllers increased in the networks. A sophisticated Hybrid Differential Evolution and Whale Optimization (DEWO) method balances exploration and exploitation of controller positioning in IoT-SDN network. The approach achieves load balancing of switches and optimal controller placement with low latency. This method reduces link breakdowns and measures end-to-end latency. The hybrid method outperforms many existing metaheuristic algorithms over as many as 40 controllers.

Keywords: SDN, IoT, Controller, smart city, optimization, metaheuristic.

Introduction

New programmable network models that decouple the management of control logic and hardware data are becoming more common [1]. SDN networks enabled by IoT provide centralized management of smart devices such as actuators and sensors. IoT network development promoted by SDN architecture. Programmable SDN networks can overcome typical IoT network limitations, including flexibility, security, configuration management, device management, etc. [2]. Smart city IoT networks are linked to a lot of gadgets, which increases the load on the network. Figure 1, Smart city IoT networks require SDN infrastructure to handle significant flow of data. The IoT enabled networks employ a number of SDN controllers to monitor flows of data from IoT devices. IoT-SDN overlap in concurrently to enhance IoT data flow stream routing [2].

Here applications require min delays, high QoS, and run time management, which traditional networking cannot provide. To meet these criteria, the modern mechanism should adopt new SDN-IoT paradigm principles. The combination of IoT and SDN environment can increase the performance and scalability of smart city networks. The traditional design of the IoT-SDN layered architecture is illustrated in Figure 2.

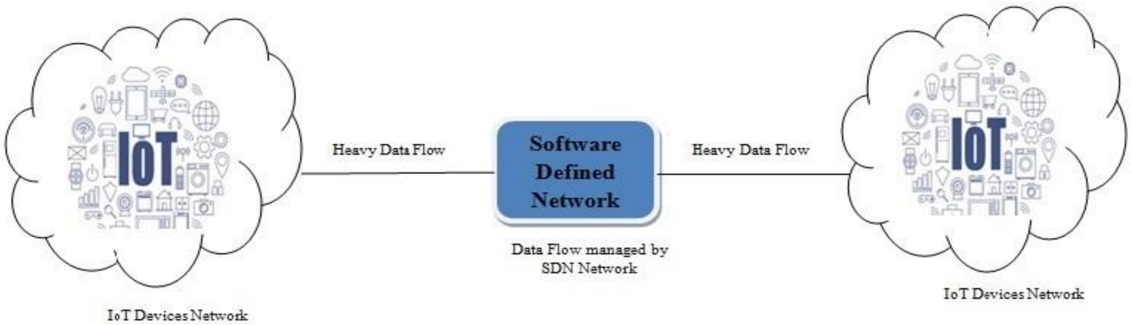

Figure 1. SDN and IoT Networks

Smart city networks require a numerous number of controllers to control IoT devices. The SDN-IoT networks require more controllers [4] to place in network. However, it is difficult to optimize many controllers in such a network to save cost and increase performance.

In rest part of paper includes related work, key contributions, method, experiments, results and last section covers the conclusion of manuscript.

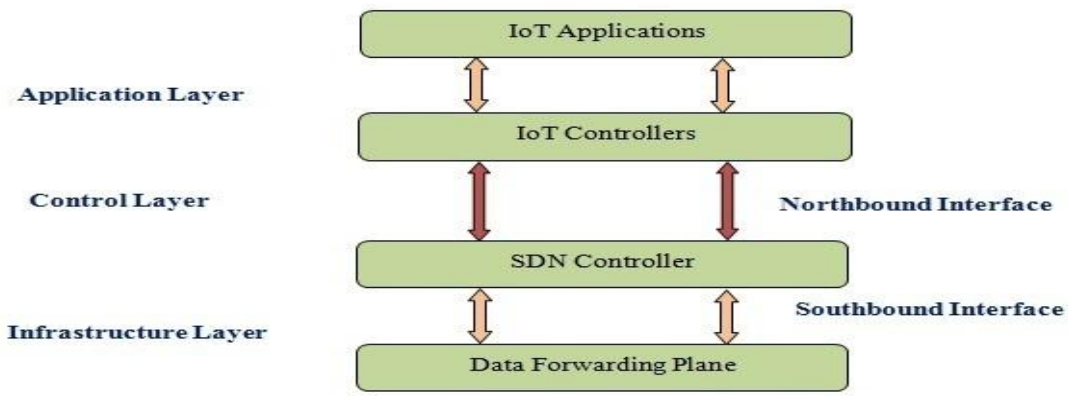

Figure 2. IoT-SDN Network Architecture

Literature review

Merits of SDN have long been attractive to recent researchers. SDN controls the IOT devices of a network. One controller cannot control all the IoT devices. SDN-IoT needs many controllers. As the minimum number and the positions of controllers are needed to handle traffic flow with minimum delay. CPP is NP-hard challenges in Industry and academia [1].

Heller et al. proposed CPP (Controller Placement Problem) and restrictions to control SDN switch devices with fault tolerance and communication latency [5]. S.K. Keshari et al. presented a rigorous SDN review addressing CPP issues, like, controller scalability, reliability, load balancing and consistency [1]. A. Jalili et al. Multi-objective Genetic Algorithm for controller placement. Greedy routing tree helps. Single controllers make it easy to manage small networks and install controllers. Single controller is not applicable to many SDN applications. Many controllers are needed for dispersed SDN-IoT networks. It is difficult to place controllers [6]. More nodes in spread out networks increases communications latency, and single controller failure. We need more controllers as IoT networks expand. All networks are topologically heterogeneous. Dixit et. al. given elastic controller distribution for distributed networks [7]. Qin. et al. [8] has given a layered IoT-SDN controller architecture to support dynamic service quality for heterogeneous IoT networks. This controller ontology supports task-level SDN-IoT flow scheduling. Olivier et al. [9] gave distributed IoT-SDN infrastructure for controller security. Optimized solutions reduce costs for SDN-IoT controller adoption and device deployment [9].

Onix solves distributed multi-controller control plane problems. Open-source controllers for networks and data centers, OpenDaylight and ONOS. Network IoT data is managed by controllers [10]. The Pareto controller placement approach by Hock accounts for the controller's overhead and latency delay. The authors proposed to use k-means and game theory to partition the controller placements. Virtual network setup adjustment for IoT performance improvement. The SDN CPP can be enhanced with cluster-based load balancing [to handle flows of sensor data[11].

This literature review mapped the research gaps. Few studies consider single-link and controller failure. The data forwarding device can select the random path when the controller is not connected to the switch, which leads to a delay in the network. It requires fast link-failure resolution. Fast switch load balancing and controller position detection using hybrid DEWO is used in this study.

Key Contribution

The paper's main contributions are:

- A DEWO algorithm for optimizing controller placements in SDN-IoTnetworks.
- The DEWO technique minimizing the average latency and enhances the performance as compared to existing techniques.

Method, Experiments and Results

Suppose the network t is like an undirected graph G(N, L), here N being a set of switch nodes and L being set of links. The shortest path d(n,c) from node n∈N to controllers c1∈C. The network graph controller numbers are m, where |C| = m. The set of all possible controllers is C = {c1, c2…cx} . Let ci, c_j ∈ C and d(c_i, c_j) be their optimal path distance. So, the average and maximum latency are defined.

Switch-Controller delays:

The max. delays among node n and controller c is:

$$\pi^{\text{max-latency}(n-c)}(C) = \max_{n\epsilon N} \min_{c\epsilon C} d(n, c) \quad (1)$$

The average delays among node n and controller c is:

$$\pi^{\text{avg-latency}(n-c)}(C) = \frac{1}{|N|} \sum_{n\epsilon N} \min_{c\epsilon C} d(n, c) \quad (2)$$

Controller-controller delays:

The average delays among controller-c_i and controller-c_j is:

$$\pi^{\text{avg-latency}(c-c)}(C) = \frac{1}{|C|} \sum_{c_{i^`}, c_j \epsilon C} \min d(c_i, c_j) \quad (3)$$

The maximum delays among controller-c_i and controller-c_j is:

$$\pi^{\text{max-latency}(c-c)}(C) = \max_{c^`_i, c_j \epsilon C} d(c_i, c_j) \quad (4)$$

Total average latency: eq (1) +eq (2)

$$\text{Total}^{\text{avg-latency}}(C) = \frac{1}{|N|} \sum_{n\epsilon N} \min_{c\epsilon C} d(n, c) + \frac{1}{|C|} \sum_{c_i, c_j \epsilon C} \min d(c_i, c_j) \quad (5)$$

Total maximum latency: eq (3) +eq (4)

$$\mathrm{Total}^{\mathrm{max-latency}} = \max_{n \epsilon N} \min_{c \epsilon C} d(n, c) + \max_{c_i, c_j \epsilon C} d(c_i, c_j) \quad (6)$$

The goal here is to reduce the total average latency Totalavg-latency(C).

Proposed Methodology

A hybridized DEWO algorithm is proposed. Smart city data is carried by IoT and software-defined networks. Lower average, max delay. Links assigned, controller switch nodes. DE and WO are DEWO-hybridized to balanced switch load and low latency controller Typical procedure of network design is shown in Figure 3. The TataNld topology is considered from Internet Topology Zoo. After node detection, the population is started at 100. The Fitness function is estimated over iterations up to 200.

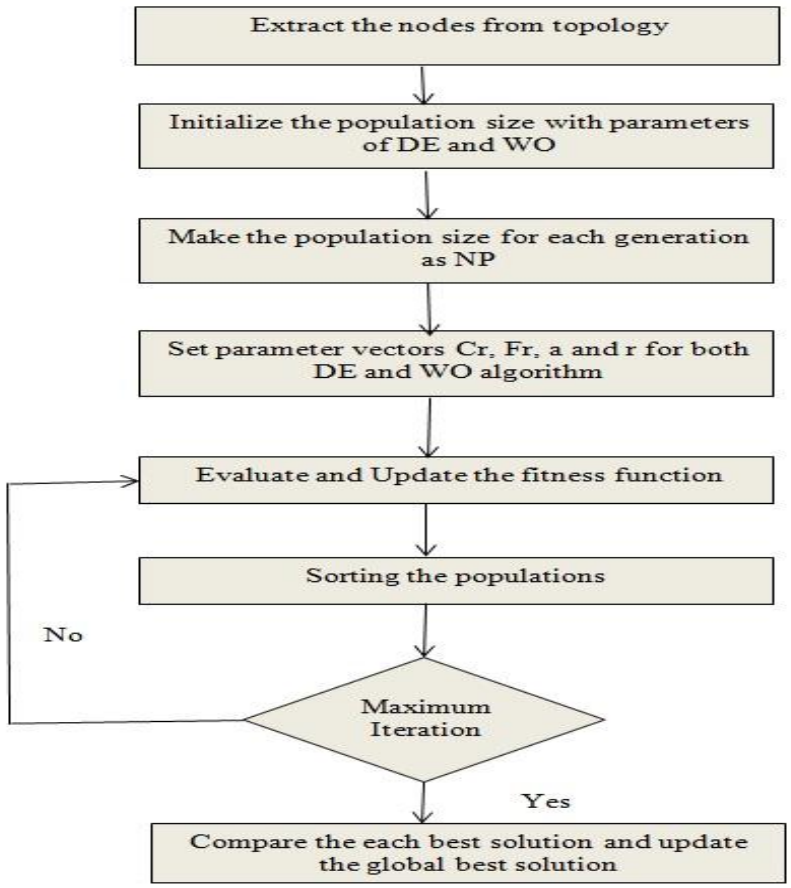

Figure 3. Workflow of the DEWO algorithm

Result and Discussion

The discussed DEWO algorithm is implemented and assessed in matlab 2019 on TataNld topology. Particle populations can be iterated 100-200 times. About 50 to 150 switch nodes and 1 to 40 controllers are considered. Simulations were performed on a Windows 10 computer with a 64-bit Intel Core i5-4 processors and 8-GB RAM. Number of controllers influences network velocity. Controllers and switch nodes are defined by latitude and longitude. More controller nodes mean greater scalability, processing latency and fault tolerance, but also more complexity.

Few controllers degrade performance with increased controllers. Figures show that more controllers change latency substantially. DEWO is better for ideal topology solutions The average delays of the topology in given Figure. 4. As the count of controllers increases, Hybrid DEWO algorithm has lower latency than other algorithms. FFA method similar to DEWO. Worst-case delay for the topology is shown in Figure 5. Worst-case latency is topology-dependent if controller number increase. DEWO has the lowest delay variation among all techniques in simulated TataNld topology.

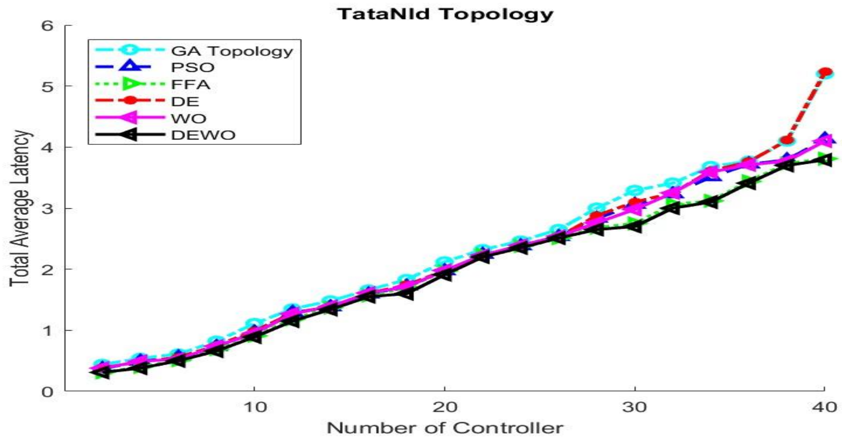

Figure 4. Average Latency

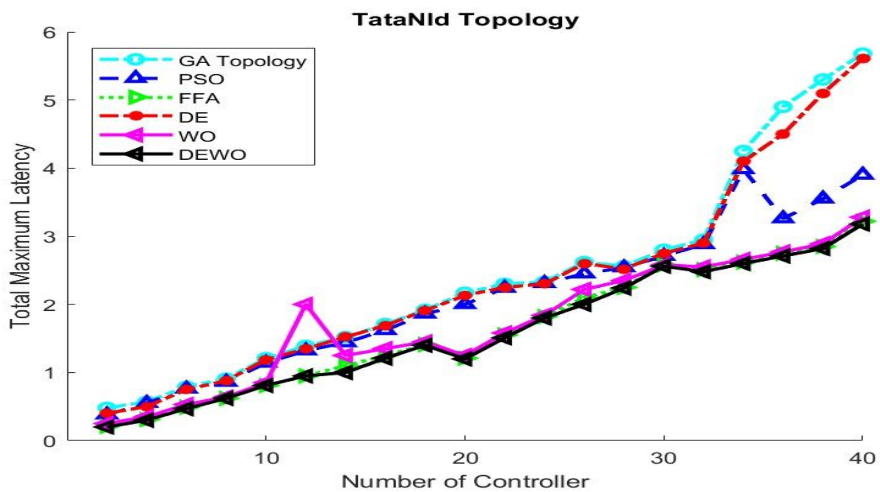

Figure 5. Worst case Latency

Conclusions

Modern networks such as SDNs separate the forwarding hardware plane and control routing decision making plane. SDN is used to manage smart networked IoT devices. Smart City, an IoT application, manages the city autonomously. IoT sensors and actuators that are growing in smart cities generate data and increase network traffic. In fast growing SDN-IoT networks, controlling many switches with a one controller is not possible. Different optimization algorithms are used to handle CPP problem. But approach performance reduces with the increase of controllers in networks. A novel Hybridized DEWO method better balancing the exploitation and exploration search. In this manuscript, hybrid DEWO is used to find the ideal position of the controllers and balance the overhead of the data forwarding devices with minimal delay. The hybridized method is tested by 40 controllers. The experimental results show minimal average and worst-case latencies. DEWO outperforms other existing algorithms in terms of latency and topology solution.

References

1. S.K. Keshari, V. Kansal, S. Kumar “A Systematic Review of Quality of Services (QoS) in Software Defined Networking (SDN)”, Wireless Personal Communications, Vol. 116, issue-3, 2593-2614 (2021).
2. K. Sood, S. Yu, and Y. Xiang, “Software-defined wireless networking opportunities and challenges for Internet-of-Things: A review,” IEEE Internet Things J., vol. 3, no. 4, 453–463, Aug. 2016.
3. X. Xu, Q. Huang, X. Yin, et al., Intelligent offloading for collaborative smart city services in edge computing, IEEE Internet Things J. (2020).
4. Wei Ren, Yan Sun, Hong Luo and Mohsen Guizani “A Novel Control Plane Optimization Strategy for Important Nodes in SDN-IoT Networks” IEEE Internet of Things Journal, Vol. 6, No. 2, April 2019.
5. B. Heller, R. Sherwood, N. McKeown. —The controller placement problem. In: ACM. ; pp. 7–12, 2012.
6. B. Heller, R. Sherwood, N. McKeown. —The controller placement problem. In: ACM. ; pp. 7–12, 2012.
7. Dixit, A.; Hao, F.; Mukherjee, S.; Lakshman, T.V.; Kompella, R.R. Elasticon an elastic distributed SDN controller. In Proceedings of the 2014 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), Marina del Rey, CA, USA, 20–21 October 2014; pp. 17–27.
8. Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, “A software defined networking architecture for the Internet-of-Things,” in Proc. IEEE Netw. Oper. Manag. Symp. (NOMS), 2014, pp. 1–9.
9. F. Olivier, G. Carlos, and N. Florent, “New security architecture for IoT network,” Procedia Comput. Sci., vol. 52, no.1, pp. 1028–1033, 2015.
10. R. Amin, M. Reisslein, and N. Shah, “Hybrid SDN networks: A survey of existing approaches,” IEEE Commun. Surveys Tuts., vol. 20, no. 4, pp. 3259–3306, 4th Quart., 2018.
11. R. Amin, M. Reisslein, and N. Shah, “Hybrid SDN networks: A survey of existing approaches,” IEEE Commun. Surveys Tuts., vol. 20, no. 4, pp. 3259–3306, 4th Quart., 2018.