I.J. Wireless and Microwave Technologies, **2022, 3, 54-60**
Published Online June 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2022.03.04

Routing in Hybrid Software Defined Networking using Hop-Count Approach

Rohitaksha K
Assistant Professor/CSE/JSS Academy of Technical Education/Bengaluru,India,560060
E-mail: rohithaksha.k@gmail.com

A B Rajendra
Professor/ISE/Vidyavardhaka College of Engineering/Mysuru,India,570002
E-mail: abrajendra@vvce.ac.in

J Mohan
CSE/JSS Academy of Technical Education/Bengaluru,India,560060
E-mail: jmohanjagadeesh@gmail.com

Received: 26 November 2021; Revised: 28 January 2022; Accepted: 10 March 2022; Published: 08 June 2022

Abstract: Software-Defined Networking is a network framework that involves in separating a network's control functions from its data forwarding functions, centralizing its intelligence, and abstraction its underlying architecture from applications and services. Hybrid Software-Define Networking is a framework which supports both the legacy protocols and the Software Defined Networking protocols to operate in the same environment. In this paper, we present an efficient technique for finding the shortest path between any two nodes in Hybrid SDN by reducing the complicated network topology to a basic topology while taking hop count into account. In the suggested proposal, a subgraph is generated from a complicated network, and the best path is found using a modified Bellman-Ford method, which saves time. The emulation test bed in Mininet was established for various numbers of nodes, and the results were analysed. The modified Bellman-Ford algorithm selects the optimal path between the two nodes. For networks with different numbers of nodes, the proposed approach offered a significant increase in the TCP throughput when compared with that of Spanning Tree Protocol. The number of packets received by the SDN controller is less with the proposed work. This increases TCP throughput by reducing the broadcast packets.

Index Terms: Software Defined Networks (SDN), OpenFlow protocol, Hybrid Software Defined Networks (Hy SDN), Hop count, Bellman-Ford algorithm.

1. Introduction

The Software-defined networking (SDN) technology is a network management approach that supports monitoring dynamically and configure networks programmatically to improve performance. SDN is a paradigm where a controller is just like a human brain that controls the behavior of the entire network. In SDN, the switches are responsible for only forwarding the packets while the "brain" or control logic is implemented in the controller. This technology has several benefits compared to conventional methods. First, it is easier to deploy new protocols and test the network without any new devices. Second, because of the centralized approach administrators can configure the network by installing the policies at the controller end without configuring the network devices individually [1].

SDN networks overcome the limitation of legacy networks, e.g., configuring the network is easy and enforcement of security policies. But SDN has several disadvantages 1) to replace all the existing switches with the SDN switches the deployment cost is too costlier, 2) traditional routing protocols are efficient for connection between forwarding plane and control plane, 3) SDN switches can handle only a few thousands of forwarding entries, etc. A network with SDN and convolutional network devices is commonly referred to as a hybrid SDN network [3].

In the existing scenario, we have Spanning Tree Protocol to eliminate the loops in the network. The drawback of STP is that whenever we have multiple paths with the same cost between one node to another node, all the traffic will be routed through the same path. Since the network overhead increase, this will reduce the network performance. In this paper first, we consider complex network topology. Furthermore, STP is intended for dispersed networks and is not applicable to Hybrid SDN. We utilised a global view of the SDN controller in this study to solve the loop problem by preventing broadcast storm instead of using the STP protocol at layer 2 of the network.

In comparison to the STP protocol, the following contributions of this work can be summarised:

- Because fewer switch ports are blocked, more switches with higher capacity may be employed in the network.
- In contrast to STP, the suggested technique employs several pathways.
- In the controller and network, the overhead of broadcast packets is reduced.
- The controller's CPU utilisation is reduced.

The rest of this paper is laid out as follows. In Section 2, we look at some related papers on STP issues and over Hybrid SDN. Sect. 3 explains the suggested strategy for loop prevention based on SDN. The proposed approach is examined in Section 4. Finally, Sect. 5 discusses the findings.

2. Related Work

The author in [4] has explored that in traditional routing the legacy switches store the details of the complete network topology and the packets are forwarded based on the forwarding table entries in the switch. In the SDN architecture, OpenFlow switches examine the header of the new packets and the details are sent to the controller which in turn compute the path and share the forwarding details to all the switches. So in this paper, the author has overcome the drawback of the existing technique and has proposed segment routing in which the packet contains only the partial forwarding information instead of storing everything in the switch reducing the overhead in the network.

In [5] authors have addressed the problem of routing stability. The paper focuses on three different approaches namely global optimization, greedy and local search to obtain routes that remain unchanged for similar traffic patterns. Here the centralized controller is used to obtain the feasible paths and deploy them on the switches. In the second case, based on the propriety classpaths are precomputed and deployed.

By deploying the SDN switches in the traditional network, legacy switches need to be upgraded to SDN-enabled ones. To migrate the conventional switches to SDN compliant ones, authors in [6] have proposed a Fully Polynomial Time Approximation scheme to obtain maximum flow. In this approach traffic is classified into controllable and non-controllable, only allowing the SDN controllable traffic in the network.

Traffic management is one of the import QoS parameters in the network. Since the SDN controllers are heavily loaded packet flows experience long propagation delay. In [7] authors have proposed techniques to find the locations of the upgraded switches and the controllers to minimize the propagation delay.

To address the increasing traffic demand, the authors in [8] addresses the smart open shortest path first technique which reduces the congestion by distributing the traffic at only the edge nodes. This is achieving by introducing a filter for the port number and IP to distribute the traffic over the edge nodes.

When an enormous amount of packets is pumped into the network it creates broadcast storms which in turn leads to network failure. To address this problem in [9] authors have proposed the technique to eliminate the loops in the network. In this approach using the candidate selection algorithm, switches are classified into a contender and non-contender switches, non-contender switches are eliminated from the network. After obtaining the subgraph STP protocol is execute to obtain the optimal path so that even the broadcast storm occurs network will not fail.

The performance of a multi-controller network architecture running OpenFlow and the Open Shortest Path First protocol is demonstrated in [15]. On a multi-controller-based network running OpenFlow versus OSPF on a mobile core network, data traffic for Packet data ratio and Jitter, as well as their related consequences was analysed. The experiment produced two topologies: a multi controller-based network and an Open Shortest Path First network, both of which were investigated. The production of traffic from User Equipment to the network-based server in the data centre and back, with traffic metrics captured on an inbuilt integrated development environment, is used to evaluate video and ping traffic. An OpenFlow controller, HyperFlow algorithm, OpenFlow switches, and Open Shortest Path First routers were used in the simulation.

The method considers a performance evaluation tool for SDN controllers that uses OpenFlow as the communication protocol [16]. This tool allows users to evaluate the throughput, latency, memory consumption, CPU usage, and consumption in kB for input/output interfaces of SDN controllers. The authors compared four well-known SDN controllers: OpenDaylight, Ryu, Floodlight, and OpenMUL, using a tool. Experiments have shown that OpenMUL and Floodlight perform similarly. Ryu was the one who got the worst outcomes. We also tested our tool against Cbench, a well-known open-source application for evaluating SDN controllers.

3. Proposed Work

Hybrid SDN architecture is shown in Fig 1. In the proposed method first, we create the subgraph for the complex network. In the subgraph, the nodes that are not essential are eliminated and thereby reducing the complexity of the network topology. The subgraph is then used as an input to the controller and by running the modified Bellman-Ford algorithm the optimal paths are created between the given source and the destination hosts.

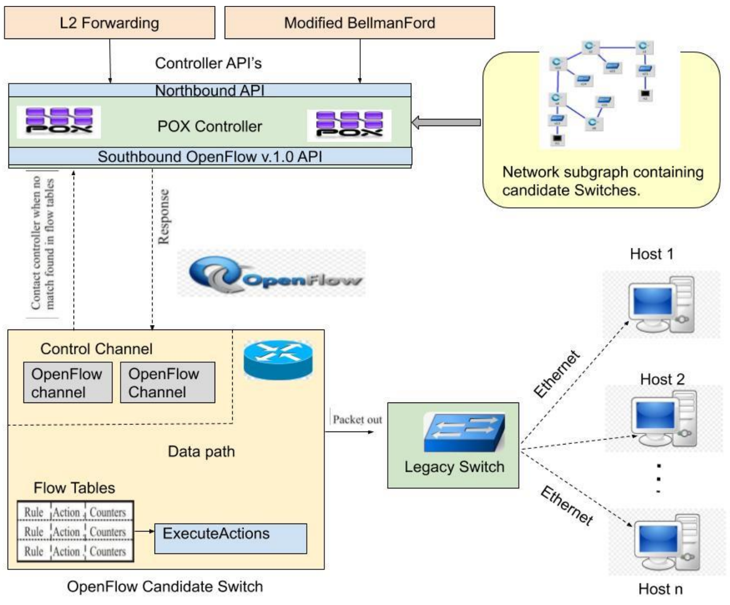


Fig.1. Proposed Hybrid SDN Model.

Table 1. Algorithm for creating subgraph from complex networks

Algorithm 1: Algorithm for creating subgraph from complex network

```
Input: The set of switches (S), N x N matrix where row (r) → S, column (c) → S, Network graph.

Output: subgraph with switches participating in routing.
1: consider the hop counts(hc→1,2,3,4) between the switches in the network
2: find the average (avg) → hc/ number of hc
3: initialize max = avg
4: for all S in the matrix:
6:        if the hc value in the corresponding r of matrix is <= max
7:        mark S has the candidate switch
8:        break
9: end
10: if S is not marked remove it from the candidate list
11: if any node is not reachable in the subgraph, we consider the intermediate node
        nearest to the isolated node to make it has a complete subgraph
12: H→ subgraph from the network graph containing the candidate switches
```

In table 1, algoritm creates a subgraph from the complex network topology. In the algorithm, we construct an N x N matrix in which the row and column representing the switches in the network. Here we consider the hop counts to create the subgraph. Initially, we consider the variable max which is used to select the node participating in the subgraph.

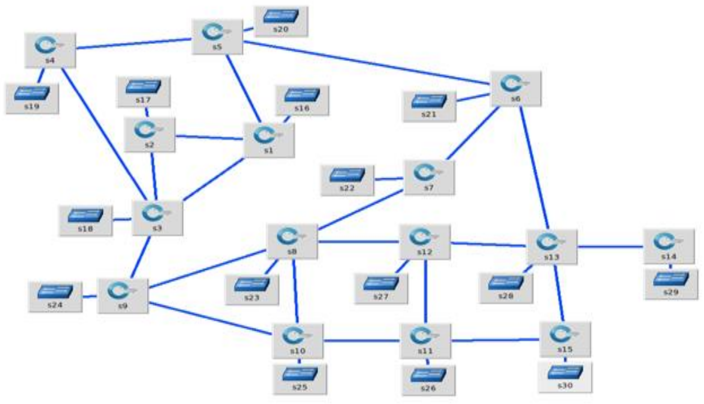


Fig.2. Hybrid SDN Network with 15 Open Flow Sitches.

Consider the network topology shown in figure 2 for the simulation. In this network, we have considered 15 OpenFlow switches connected. After applying algorithm 1 we will get the subgraph consisting of candidate switches as shown in figure 3. Similarly, for the simulation purpose, we have considered a network with 11 and 20 nodes.

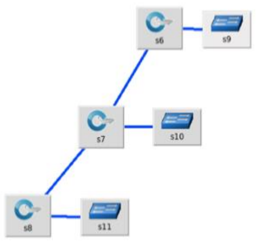
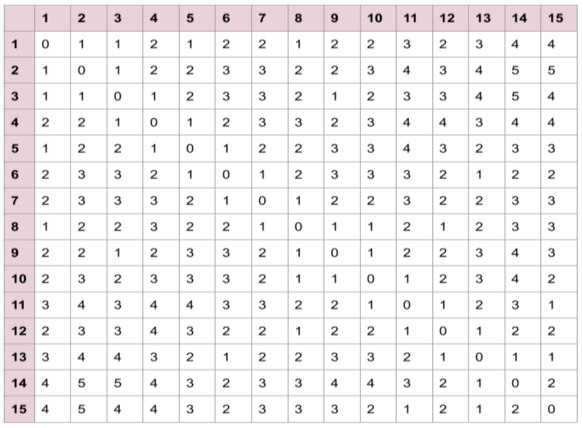


Fig.3. Network created after applying algorithm 1.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	2	1	2	2	1	2	2	3	2	3	4	4
2	1	0	1	2	2	3	3	2	2	3	4	3	4	5	5
3	1	1	0	1	2	3	3	2	1	2	3	3	4	5	4
4	2	2	1	0	1	2	3	3	2	3	4	4	3	4	4
5	1	2	2	1	0	1	2	2	3	3	4	3	2	3	3
6	2	3	3	2	1	0	1	2	3	3	3	2	1	2	2
7	2	3	3	3	2	1	0	1	2	2	3	2	2	3	3
8	1	2	2	3	2	2	1	0	1	1	2	1	2	3	3
9	2	2	1	2	3	3	2	1	0	1	2	2	3	4	3
10	2	3	2	3	3	3	2	1	1	0	1	2	3	4	2
11	3	4	3	4	4	3	3	2	2	1	0	1	2	3	1
12	2	3	3	4	3	2	2	1	2	2	1	0	1	2	2
13	3	4	4	3	2	1	2	2	3	3	2	1	0	1	1
14	4	5	5	4	3	2	3	3	4	4	3	2	1	0	2
15	4	5	4	4	3	2	3	3	3	2	1	2	1	2	0

Fig.4. Matrix with 15 x 15 constructed for the network with 15 nodes.

Figure 4 shows the matrix where row and column represent the nodes in the network. Each cell in the corresponding row represents the number of hops from each node to other nodes. Candidate nodes are selected from the given matrix only if the hop count value in the corresponding row is less than the max value. Once the candidate nodes are selected based on algorithm 1 we will obtain the subgraph as shown in figure 3.

The shortest path from one OpenFlow switch to all the further OpenFlow switches in network topology is obtained from the Bellman-Ford algorithm [11]. In this work, we have used the modified Bellman-Ford algorithm [11] to find the optimal path by eliminating the acyclic graphs.

Table 2. Modified Bellman-Ford algorithm for selecting optimal path.

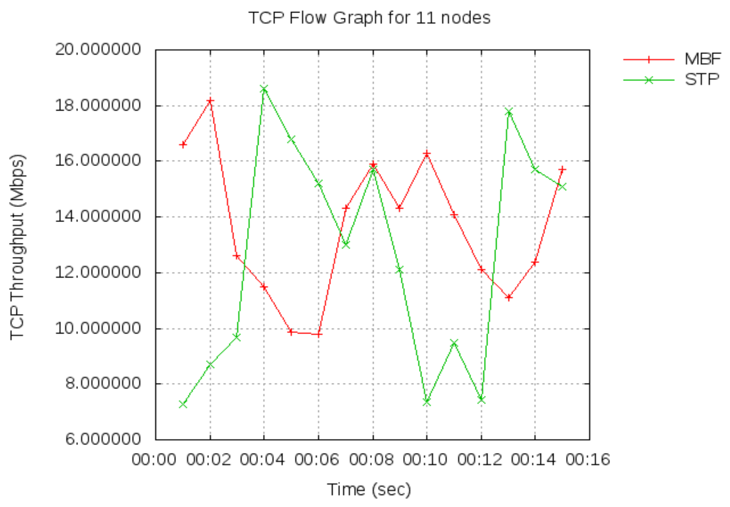
Algorithm 2: shortest path selection using Modified Bellman-Ford algorithm

```
1. d[s] ← 0
2. for each v £V - {s}
        do d[v] ← ∞
        for i ← 1 to |V| -1
                for i ← 1 to |V|-1
                        do for each edge(u,v) £E
                                do if d[v] > d[u]+w(u,v)
                                then d[v] ← d[u]+w(u,v)
3: for each edge (u,v) £E
        do if d[v]>d[u]+w(u,v)
        then report that a negative weight cycle exists.
4: At the end, d[v]= §(s,v), if no negative weight cycles.
```

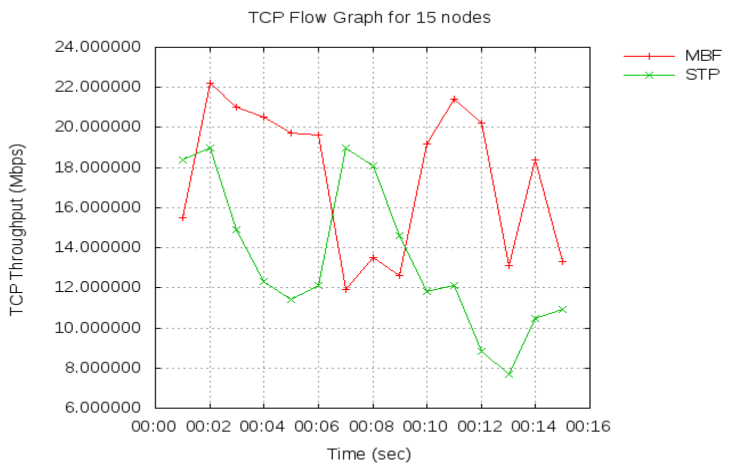
4. Result Analysis

We considered the POX controller [10] as the network controller and Mininet [16] to create the network topology. The results were analyzed by considering a different number of nodes. For the experiment, we created a network with 11, 15, and 20 nodes. After applying our proposed method network with 11,15, and 20 nodes will have proximately 8,3, and 4 nodes selected as candidate nodes.

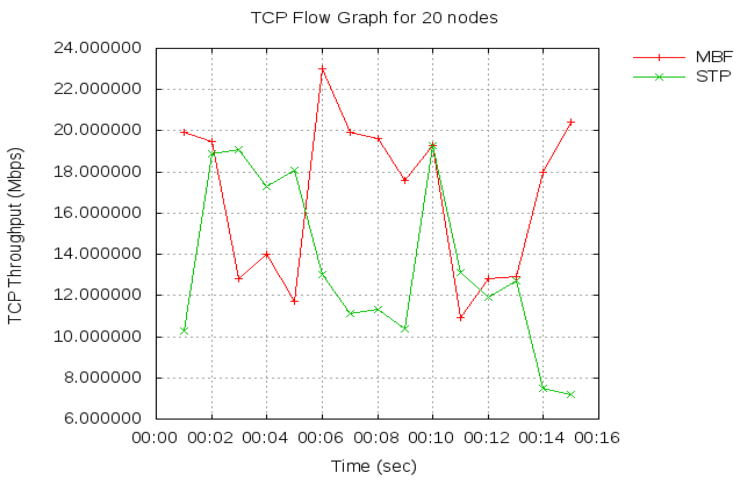
We measured the TCP throughput by considering the complex network with 11, 15, and 20 nodes. Initially, we created the network and applied our proposed hop count approach in Mininet to reduce the complexity of the network. Once the subgraph is created we execute the Pox controller by providing our algorithm 2 as an input to find the optimal path between the source and destination hosts. From figure 5, we can analyze that our approach with the STP and modified Bellman-Ford algorithm, TCP throughput increases as the time increases compared to STP.



(a)



(b)



(c)

Fig.5. TCP convergence concerning time for STP and algorithm 1 in the Hybrid SDN for 11, 15, and 20 nodes.

5. Conclusion

Hybrid SDN (Software Defined Networking) is a new networking architecture in which the control plane and forwarding plane are separated. It claims to make network management easier while also allowing for network innovation. The control plane in Hybrid SDN is logically centralised , and network devices are only responsible for packet forwarding. STP is used in outdated networks to prevent layer 2 loops. STP has a number of issues, including scalability for large networks such as datacenters. It is also not ideal for SDN because it is built for scattered networks. To tackle the loop problem in this work, we employed the global view of the SDN controller. As a consequence, we were able to get greater performance than STP by employing the global view of the SDN controller. The proposed mechanism combines the Modified Bellman-Ford algorithm with the subgraph algorithm to select the influential nodes. Hybrid SDN eliminates the initial deployment cost required to set up the new network with only the Open Flow switches. In the proposed mechanism we can obtain better performance by reducing the complexity of the complete network. Due to less complexity, the number of broadcast packets is reduced, increasing the performance of the network and improving the convergence time. The proposed work can be extended to reduce link failure and congestion in the network.

References

[1] Jain R, Paul S. Network virtualization and software-defined networking for cloud computing: a survey. IEEE Communications Magazine. 2013 Nov 11;51(11):24-31. DOI: 10.1109/MCOM.2013.6658648.

[2] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM computer communication review. 2008 Mar 31;38(2):69-74. DOI: 10.1145/1355734.1355746.

[3] Amin R, Reisslein M, Shah N. Hybrid SDN networks: A survey of existing approaches. IEEE Communications Surveys & Tutorials. 2018 May 17;20(4):3259-306. DOI: 10.1109/COMST.2018.2837161.

[4] Li Z, Huang L, Xu H, Zhao G. Segment routing in hybrid software-defined networking. In2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN) 2017 May 6 (pp. 160-165). IEEE. DOI: 10.1109/ICCSN.2017.8230098

[5] Tseng SH, Tang A, Choudhury GL, Tse S. Routing stability in hybrid software-defined networks. IEEE/ACM Transactions On Networking. 2019 Mar 11;27(2):790-804. DOI: 10.1109/TNET.2019.2900199

[6] Hu Y, Wang W, Gong X, Que X, Ma Y, Cheng S. Maximizing network utilization in hybrid software-defined networks. In2015 IEEE Global Communications Conference (GLOBECOM) 2015 Dec 6 (pp. 1-6). IEEE. DOI: 10.1109/GLOCOM.2015.7417144

[7] Guo Z, Chen W, Liu YF, Xu Y, Zhang ZL. Joint switch upgrade and controller deployment in hybrid software-defined networks. IEEE Journal on Selected Areas in Communications. 2019 Mar 21;37(5):1012-28.DOI: 10.1109/JSAC.2019.2906743.

[8] Nakahodo Y, Naito T, Oki E. Implementation of smart-OSPF in a hybrid software-defined network. In2014 4th IEEE International Conference on Network Infrastructure and Digital Content 2014 Sep 19 (pp. 374-378). IEEE. DOI: 10.1109/ICNIDC.2014.7000328

[9] Amiri E, Javidan R. A new method for layer 2 loop prevention in software-defined networks. Telecommunication Systems. 2020 Jan;73(1):47-57.

[10] Kaur S, Singh J, Ghumman NS. Network programmability using POX controller. InProc. Int. Conf. Commun., Comput. Syst.(ICCCS) 2014 (Vol. 138, pp. 134-138.

[11] Bellman, Richard (1958). "On a routing problem". Quarterly of Applied Mathematics. 16: 87– 90. DOI: https://doi.org/10.1090/qam/102435

[12] MOHAN J, DEVANAGAONKAR SM, GARUR VV, KARTHIK A, ROHITAKSHA K, RAJENDRA A. An Improved Approach for Eliminating the Loop in Hybrid Software Defined Network (SDN) Using Modified Bellman-Ford Algorithm. Journal of Interconnection Networks. 2020 Dec;20(04):2150001.

[13] Qin K, Fu B, Chen P, Huang J. MCRA: Multicost Rerouting Algorithm in SDN. Journal of Advanced Computational Intelligence and Intelligent Informatics. 2020 Nov 20;24(6):728-37.

[14] MiniNet. http://minin et.org/. Accessed 17 January 2018

[15] Omumbo, N., Muhambe, T.M. and Ratemo, C.M., 2021. Evaluation of Routing Performance using OSPF and Multi-Controller Based Network Architecture. International Journal of Computer Network & Information Security, 13(4).

[16] Gamess, E., Tovar, D. and Cavadia, A., 2018. Design and implementation of a benchmarking tool for OpenFlow controllers. Int. J. Inf. Technol. Comput. Sci, 10(11), pp.1-13.

Authors’ Profiles

Rohitaksha K received the MTech degree from the Department of Electronics and Communication Engineering, Visveswaraya Technological University in 2012. His areas of intrest are Software Defined Networks, Machine Learning, Database Managament Systems, Python. He has Published papers in peer reviewed journals and International conferences.

Dr. A B Rajendra received his PhD in the year 2015 from Visveswaraya Technological University , Belagavi. His areas of intrest Cryptography, Network Security, Internet of Things, Machine Learning. He has published papers in peer reviewed journals, International conferences, National Conferences.He has received several National awards like Bharath Vikas Award winner in 2017. He is member of IEEE, ISTE, IETE, CRSI.

How to cite this paper: Rohitaksha K, A B Rajendra, J Mohan, " Routing in Hybrid Software Defined Networking using Hop-Count Approach", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.12, No.3, pp. 54-60, 2022.DOI: 10.5815/ijwmt.2022.03.04