An Efficient Routing Algorithm for Software Defined Networking Using Bellman Ford Algorithm

https://doi.org/10.3991/ijoe.v15i14.11546

Kavikondala Praveen Kumar Rao (✉), Tamilarasan Senthil Murugan
Vel Tech Rangarajan Dr Sagunthala R & D Institute of Technology and Science, Chennai, India
praveenkumarrao.k@gmail.com

Abstract—Software-Defined Networking (SDN) is the developing technology and has the advantages of handling dynamic nodes in the network with improved performance. SDN has the problem of allocating the resources to the user with high latency and this affects the overall system performance. To solve this problem, the routing method based on Bellman Ford Algorithm (BFA) is proposed in the SDN. The Bellman-Ford has less computation time in identifying the shortest path in the nodes of SDN graph. The BFA is applied to identify the optimal path for the nodes to the user with low latency. The BFA is compared with Dijkstra’s algorithm to analyze its performance. The experimental outcome shows that the BFA has lower latency compared to the Dijkstra’s algorithm. The lower computation time is achieved due to BFA has a lower magnitude time in vertices and edges compared Dijkstra's algorithm. The Dijkstra’s algorithm has the latency of 10.8 ms, while proposed BFA has the latency of 2.97ms.

Keywords—Bellman Ford Algorithm, Dijkstra's Algorithm, Latency, Shortest Path, Software Defined Networking.

1 Introduction

The internet has been utilized by many users; this creates the large traffic on the internet, which is maintained by the emerging services such as cloud services, server virtualization, service for mobile devices and big data analytics etc. [1]. The OpenFlow method is difficult to scale due to the configuring fine grained paths that include the strict rules to the forwarding table of switches [2]. The number of user increases in the network affects the quality of services required are significantly different [3]. Traffic Engineering technique is widely used to increase the network performance and avoid network congestion, and has been applied in both industries and academic [4]. The controller needs to know the accurate network traffic to maintain the better control decision in network and this also important to acquire the traffic matrix for SDN [5].

As the system develops, the new configuration has been update in the network equipment to optimize the network performance based on traffic variation [6]. The SDN forwarding capacity helps to control the unprecedented fine-grained on the traffic occurs in the network [7]. Updating the schemes requires high cost, since any modification in
the individual field or modify the rule of 35 may break the existing dependencies, and this cause to redistribute the system [8]. Decoupling of the control layer from the data forward layer is involved in major role in large scale, high speed computing system [9]. All paths between the source and destination node pair is needed to reduce the communication cost and improve the interaction among the users in a smart community [10].

To optimize the large SDN network for traffic demands, fast and efficient routing protocol is required. The objective of the proposed method is to analyze the network routing in SDNs and apply BFA for dividing the network into many subnets and identify the shortest path in every subnet and between subnets. The proposal of a routing algorithm for large-scale SDNs based on the BFA for embedding the proposed algorithm code in the SDN controller and the testbed design for the proposed routing algorithm.

The paper is organized as follows, literature survey is given in the section 2, and the proposed Bellman Ford method in SDN explained in section 3, experimental result of proposed method is illustrated in section 4. The conclusion of this research work is made in section 5.

2 Literature Survey

C. Zhang, et al., [11] developed a Neural Network-based Intelligent Routing Schema for SDN (NNIRSS) for control the transmission pattern of data flow based on NN and replace the flow table with well-trained NN packets. With a combination of APC-III and k-means algorithm, the APCK-means method to estimate the centers of radial basis functions. The simulation result shows that the developed NNIRSS method has high feasible and effectiveness in the system. The NNIRSS reduce the storage space and communication time as well as increase routing efficiency. The route prediction success rate is low and cannot adapt to the large-scale SDN system.

Y. Wang, et al., [12] developed Sleep Scheduling algorithm for the energy saving based on Energy Consumed Uniformly-Connected K-Neighborhood (ECCKN) method and also applied to manage the energy in the network. The developed method completed every computation in the controller and there was no broadcasting takes place between two nodes, which is the important features in the traditional ECCKN technique. The experimental analysis shows that the developed method has a higher performance in terms of energy management, network lifetime, managing the live nodes and solo nodes in the network. The developed ECCKN is working well on the static WSN only.

R. Maaloul, et al., [13] identifies the problem in the energy-aware routing in carrier-grade Ethernet networks SDN. The method is based on the turn-off the unused nodes and links to minimize the energy consumption without affecting the Openflow switch rule and maintaining the allowable maximum link utilization. First, presented a method based on an Integer Linear Programming Formulation. A set of first-fit heuristic method is convenient for large-sized networks. The experimental result shows that the developed method has the efficiency on both exact and heuristic method. The method has a low flow table size, which is one of the main constraints in this model.
H. Ghafoor, and Insoo Koo, [14] developed the routing protocol for the SDN in the vehicle cognitive network to find the suitable route between the source and destination. As the method involves in the cognitive routing protocol, the primary task of the algorithm is spectrum sensing to increase network stability. A belief propagation method is applied for channel selection. Experimental analysis shows that the developed method has higher performance in the manner of end-to-end delay, high delivery ratio and low overhead. The message exchange rate among vehicles increases to identify an idle channel.

C. Bu, et al., [15] developed an Adaptive Routing Service Customization (ARSC) method based on the Network Function Virtualization (NFV). In this method, the routing services are customized based on the application with user utility and profit. The simulation result shows that the ARSC was feasible and effective. With only one single central controller, this inevitably brings scalability problem.

3 Proposed Method

The SDN is the developing method in the network architecture, which have advantages over the conventional network. The SDN has lower performance in finding the nodes to the user request with considerable performance. Segment Routing is the path of information in the network. In this research, the routing method based on BFA is applied in the SDN to solve the scalability problem. The SDN is represented in the graph with the set of nodes and edges. The bellman ford finds the optimal path for the user based on the nodes in the graph with less computation time. The block diagram of the proposed BFA in SDN is shown in Fig. (1).

![Diagram](image-url)

**Fig. 1.** The block diagram of the proposed method
3.1 Software defined networking topology

An SDN network is represented in weighted graph $G = (V, E)$, where $V$ is the set of nodes, and $E$ is the set of edges. Each vertex in $V$ and each edge in $E$ represents the switch and a switch link in SDN, respectively. For each edge $e \in E$, let $b(e)$ represent the currently available bandwidth of a link that connect with pair of nodes. The weight function is defined for each edge $e \in E$, which named as link weight $w(e)$. Consider $w(e) > 0$ and this is used to calculate the optimal route, which has the minimum weight path for the pair of nodes. As minimum weight path involves in the longer distance, longer time, more consumption of bandwidth and high overhead.

Traffic Matrix (TM) is defined in this method, which records the traffic demands in a time interval $P$. The TM consist of set of nodes in the time interval $P$, which is considered as the potential traffic demand in the future. The demand of traffic is represented as the request from the source $s \in V$ to destination $d \in V$, with respected bandwidth $B_{sd}$. In addition, the minimum hop count $H_{sd}$ is defined between the source node $s$ and destination node $d$. Assume $MAX_{sd}$ be the considerable maximum hop count path between source node and destination node, which is the sum of $H_{sd}$ and extra hop count $E_{sd}$. The extra hop count is applied to reduce the maximum path length for the solution. In this research, BFA is proposed to identify the optimal path to balance the link loads of switches.

3.2 Bellman-Ford algorithm

The basic BFA process is relaxation loop on the graph edges, in any order [16]. The BFA can be applied for the graphs with negative edges. The BFA identify the shortest path cost to the vertices and shortest path tree in $r + 1$ rounds, where $r$ is the minimal integer such that for any vertex reachable from $s$, shortest path is present from $s$ to it containing at most $r$ edges. The input graph has a negative cycle reachable from $s$, shortest path to some vertices do not exist, and Bellman-Ford identify that in $|V|$ rounds. Otherwise, $r$ as above is at most $|V| - 1$. The single round cost is $O(|E|)$, the implied running time bound is $O(|V||E|)$.

BFA is the shortest path algorithm that measures the shortest path from the source vertices to all other vertices in a weighted directed graph. The SDN graph is given as input to the BFA to identify the shortest path of the switches. The number of vertices $V$ and edges $E$ are taken from the graph. The relaxation method is used in this method in which approximate distance to each vertex is always greater than or equal to true distance. If the new value is lower than old value, new value is replaced. In this method, all the edges are relaxed for $|V| - 1$ times. After processing the whole network, the shortest path from the source node will be identified. This algorithm usually takes a large amount of time $O(VE)$. 

http://www.i-joe.org
Algorithm: Bellman Ford

**Input:** list vertices, list edges, vertex source

**Output:** distance [], predecessor []

// Step 1: initialize graph
for each vertex v in vertices:
    distance[v] := inf  // Initialize the distance to all vertices to infinity
    predecessor[v] := null  // And having a null predecessor
    distance[source] := 0  // The distance from the source to itself is, of course, zero

// Step 2: relax edges repeatedly
for i from 1 to size(verticess)-1:
    for each edge (u, v) with weight w in edges:
        if distance[u] + w < distance[v]:
            distance[v] := distance[u] + w
            predecessor[v] := u

// Step 3: check for negative-weight cycles
for each edge (u, v) with weight w in edges:
    if distance[u] + w < distance[v]:
        error "Graph contains a negative-weight cycle"
return distance[], predecessor[]

4 Experimental Result

The SDN technology has more efficiency than a conventional method in dynamic and programmable network. The SDN allocate the user to the nodes, which are having higher latency that affects the system performance. To solve this problem, the routing algorithm based on BFA is proposed. The BFA is applied to find the optimal nodes to user based on the vertices and edges. This technique will reduce the latency in the system and increase the scalability. The Bellman Ford has the advantages of low computation time in finding the shortest path. The metrics such as latency, throughput, and rejection rate is calculated to analyze the performance of the proposed BFA in SDN. The BFA is compared with Dijkstra's algorithm in finding shortest path in the SDN graph. This section will provide a brief explanation about the performance of the proposed BFA in SDN.
The performance of the Dijkstra algorithm and BFA are analyzed in the SDN graph, as shown in Table 1. The metrics such as Bandwidth, latency and rejection ration are measured in the graph. The result shows that the proposed BFA in the SDN graph has the higher performance compared to Dijkstra method. The rejection ratio of the bellman ford method is low compared to the Dijkstra algorithm.

| Methods       | Bandwidth (MB/sec) | Latency (ms) | Rejection Ratio (%) |
|---------------|--------------------|--------------|--------------------|
| Dijkstra Algorithm | 0.95               | 10.807       | 4.9                |
| BFA           | 1.05               | 2.979        | 0.075              |

The proposed Bellman ford and dijkstra algorithm are tested on the different path and analyze the processing time, as shown in Table 2. The paths are randomly chosen to test the performance of the existing and proposed method. This shows that the proposed method has the low processing time compared to the dijkstra method.

| Methods       | Processing Time (ms) | Paths                      | Processing Time (ms) | Paths                      |
|---------------|----------------------|----------------------------|----------------------|----------------------------|
| Dijkstra Algorithm | 1205.545            | {S7,S6,S2,S3,S4}           | 1207.489            | {S8,S6,S2,S3,S4}           |
| BFA           | 1204.712             | {S7,S6,S2,S3,S4}           | 1205.278            | {S8,S6,S2,S3,S4}           |

Figure 2 shows that the bandwidth for the Dijkstra and BFA. The BFA has the higher bandwidth compared to the dijkstra method. The bandwidth of the BFA is 1.05 MB/s, while the Dijkstra method has the bandwidth of 0.95 MB/s.
The latency is measured for the different method such as Dijkstra and BFA, as shown in Fig. (3). The proposed Bellman Ford method has lower latency compared to the Dijkstra algorithm. The Dijsktra’s method has the latency of 10.807 ms in SDN graph and BFA has the latency of 2.97 ms.

The experimental result shows that the proposed bellman ford method has higher performance compared to the Dijsktra’s method in SDN graph. The proposed Bellman ford technique can be applied in the SDN graph to find the optimal path.

5 Conclusion

The major issue in the SDN is that the user is allocated with the resources of the higher latency. This research aims to propose the Bellman Ford based routing algorithm to find the optimal path for the nodes allocate to the user. The SDN data are represented in the graph with nodes and vertices. The BFA is applied in the SDN graph to find the optimal path for the user. The Bellman Ford uses the vertices and edges to find the optimal path for the user. The Bellman Ford is compared with the Dijkstra’s algorithm to analyze its efficiency in the SDN system. This shows that the proposed BFA has higher performance compared to the Dijkstra’s algorithm. The proposed BFA has a bandwidth of 1.05 MB/s, while Dijkstra’s algorithm has the bandwidth of 0.95 MB/s. The proposed BFA has higher performance due to that Bellman Ford has a lower magnitude in the time complexity compared to the Dijkstra’s algorithm. The future scope of this method involves increasing the Quality of Services (QoS) in the SDN.
6 References

[1] Anithaashri, T.P., Ravichandran, G. and Baskaran, R. (2019). Security enhancement for software defined network using game theoretical approach. Computer Networks, 157: 112-121. https://doi.org/10.1016/j.comnet.2019.04.014

[2] Long, L., Yu, H., Luo, S., Ye, Z., Du, X. and Guizani, M. (2019). Scalable explicit path control in software-defined networks. Journal of Network and Computer Applications. https://doi.org/10.1016/j.jnca.2019.05.014

[3] Liu, T., Qin, P., Li, L. and Tang, Y. (2019). Software-defined converged access network with cross-layer intelligent control architecture. Optical Fiber Technology, 50: 242-249. https://doi.org/10.1016/j.yofte.2019.04.001

[4] Guo, Y., Wang, Z., Liu, Z., Yin, X., Shi, X., Wu, J., Xu, Y. and Chao, H.J. (2019). SOTE: Traffic Engineering in Hybrid Software Defined Networks. Computer Networks. 8: 154:60-72. https://doi.org/10.1016/j.comnet.2019.03.008

[5] Li, D., Xing, C., Zhang, G., Cao, H. and Xu, B. (2019). An online dynamic traffic matrix completion method in software defined networks. Computer Communications. https://doi.org/10.1016/j.comcom.2019.05.023

[6] Sanvito, D., Filippini, I., Capone, A., Paris, S. and Leguay, J. (2019). Clustered robust routing for traffic engineering in software-defined networks. Computer Communications. https://doi.org/10.1016/j.comcom.2019.06.002

[7] Wang, X., Zhang, Q., Ren, J., Xu, S., Wang, S. and Yu, S. (2018). Toward efficient parallel routing optimization for large-scale SDN networks using GPGPU. Journal of Network and Computer Applications, 113: 1-13. https://doi.org/10.1016/j.jnca.2018.03.031

[8] Wang, L., Li, Q., Sinnott, R., Jiang, Y. and Wu, J. (2018). An intelligent rule management scheme for Software Defined Networking. Computer Networks, 144: 77-88. https://doi.org/10.1016/j.comnet.2018.07.027

[9] Imran, M., Durad, M.H., Khan, F.A. and Derhab, A. (2019). Toward an optimal solution against Denial of Service attacks in Software Defined Networks. Future Generation Computer Systems, vol. 92, 444-453. https://doi.org/10.1016/j.future.2019.09.022

[10] Attia, M.B., Nguyen, K.K., and Cheriet, M., (2018). QoS-aware software-defined routing in smart community network. Computer Networks, vol. 147, pp. 221-235. https://doi.org/10.1016/j.comcom.2018.10.015

[11] Zhang, C., Wang, X., Li, F., and Huang, M. (2018). NNIRSS: neural network-based intelligent routing scheme for SDN. Neural Computing and Applications, 1-17. https://doi.org/10.1007/s00521-018-4327-z

[12] Wang, Y., Chen, H., Wu, X., and Shu, L. (2016). An energy-efficient SDN based sleep scheduling algorithm for WSNs, Journal of Network and Computer Applications, 59, 39-45. https://doi.org/10.1016/j.comcom.2015.05.002

[13] Maaloul, R., Taktak, R., Chaari, L. and Cousin, B. (2018). Energy-Aware Routing in Carrier-Grade Ethernet using SDN Approach, IEEE Transactions on Green Communications and Networking, 2018. https://doi.org/10.1109/tgcn.2018.2832658

[14] Ghafoor, H., and Insoo, K. (2018). CR-SDVN: A cognitive routing protocol for Software-Defined vehicular networks, IEEE Sensors Journal, 18: 1761-1772. https://doi.org/10.1109/717.2788014

[15] Bu, C., X. Wang, H. Cheng, M. Huang, K. Li, and Das, S.K. (2017). Enabling adaptive routing service customization via the integration of SDN and NFV, Journal of Network and Computer Applications, 93: 123-136. https://doi.org/10.1016/j.jnca.2017.05.010

[16] Dinitz, Y. and Izhak, R. (2017). Hybrid Bellman–Ford–Dijkstra algorithm. Journal of Discrete Algorithms, 42: 35-44. https://doi.org/10.1016/j.jda.2017.01.001
Short Paper—An Efficient Routing Algorithm for Software Defined Networking Using BFA

7 Authors

Kavikondala Praveen Kumar Rao completed BE in Computer Technology and M. Tech in Software Engineering from Nagpur University and Kakatiya University in 1997 and 2006 respectively. He published 19 papers in various International Journals. He is working in Department of Computer Science & Engineering, Vel Tech Rangarajan Dr Sagunthala R & D Institute of Technology and Science, Avadi, Chennai, India.

Dr. Tamilarasan Senthil Murugan is working in Department of Computer Science & Engineering, Vel Tech Rangarajan Dr Sagunthala R & D Institute of Technology and Science, Avadi, Chennai. senthilmuruganme@gmail.com

Article submitted 2019-08-19. Resubmitted 2019-09-23. Final acceptance 2019-09-25. Final version published as submitted by the authors.