British Journal of Mathematics & Computer Science
21(3): 1-8, 2017; Article no.BJMCS.30609
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Simulation of Minimum Path Estimation in Software Defined Networking Using Mininet Emulator

S. M. Shamim[1*], Mohammad Badrul Alam Miah[1], Angona Sarker[1], Ali Newaz Bahar[1] and Ananya Sarker[2]

[1]Department of Information, Communication and Technology, Mawlana Bhashani Science and Technology University, Bangladesh.
[2]Department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Bangladesh.

Authors' contributions

This work was carried out in collaboration between all authors. Authors SMS and MBAM designed the study, performed the statistical analysis. Author AS managed the analyses of the study. Authors ANB and AS managed the literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJMCS/2017/30609
Editor(s):
(1) Doina Bein, Applied Research Laboratory, The Pennsylvania State University, USA.
(2) Junjie Chen, Department of Electrical Engineering, University of Texas at Arlington, USA.
(3) Paul Bracken, Department of Mathematics, The University of Texas-Pan American Edinburg, TX 78539, USA.
Reviewers:
(1) Zheng Zhao, Zhengzhou Science and Technology Institute, China.
(2) Yang Hui, Beijing University of Posts and Telecommunications, China.
Complete Peer review History: http://www.sciencedomain.org/review-history/18458

Received: 21st November 2016
Accepted: 1st March 2017
Published: 1st April 2017

Short Research Article

Abstract

Software-Defined Networking (SDN) has become a significant topic of discussion among the network service providers, operators, and equipment vendors where control planes are separated from the data plane in networking devices. This paper implements Bellman-Ford algorithm for computing the shortest path in Software-Defined Networking using Mininet emulator. Bellman–Ford algorithm computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph. This algorithm is versatile, as it is capable of handling graphs in which some of the edge weights are negative numbers. All the simulation has been done using POX as an OpenFlow controller, OpenvSwitch (OVS) as a forwarding function and Mininet which installed on Ubuntu Virtual Machine (VM). The result of this paper shows that the simulation of SDN with OpenvSwitch (OVS) and POX controller runs Bellman-Ford algorithm for finding the minimum path among the designed network topology.

**Corresponding author: E-mail: ictshamim@yahoo.com;*

Keywords: Software-defined networking; Bellman-Ford algorithm; OpenFlow switch; POX controller; virtual machine; Mininet emulator.

1 Introduction

Future internet is required to be more secure, flexible, reliable, and having other advanced features. In traditional network control plane that provides information used to construct a forwarding table and data plane that consults the forwarding table are combined together. Network device conducts forwarding table to make a decision on where to send packets or frames penetrating the device. Since control plane and forward plane exist directly on the networking device, conventional networks become more complicated.

Software Defined Networking (SDN) is an emerging scheme of networking that enables network programmability, faster innovation and simplified network management. SDN fulfills most of the task to achieve require functioning. SDN is revealing a new approach for developing new network services. Furthermore SDN can simplify network monitoring and network management. The control plane is separate from the forwarding plane in SDN [1]. OpenFlow [2,3] technology is the first standard for SDN which capable for controlling multiple hardware or software switches from a single controller. The control plane communicates with the data plane through OpenFlow mechanism. A single control plane controls several forwarding devices [4]. Control plane makes the decision what to do with packets (E.g. sets up forwarding plane rules) i.e. how and where to deliver the packets. SDN uses a centrally managed controller to form flow tables that set up the forwarding table responsible for delivering packets in the network [5-7]. A simplified view of this architecture is shown in Fig. 1.

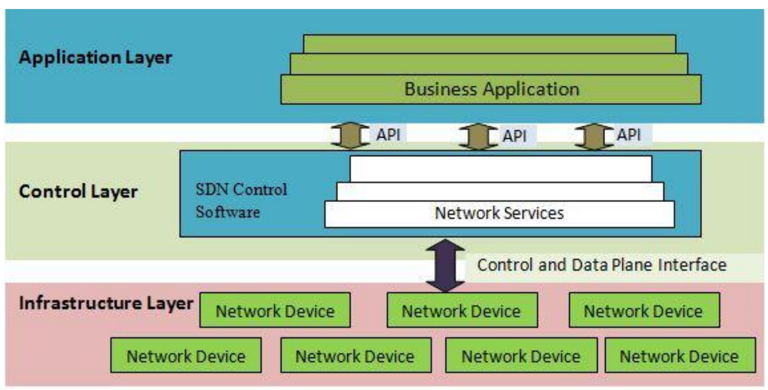

Fig. 1. SDN architecture

SDN separate the vertical integration by decoupling the network's control plane from the data plane which underlying routers and switches and forward the traffic. Second, Network switches become forwarding devices and the control logic is implemented in a logically centralized controller with decouple of the control and data planes. Logically centralized programmatic model does not postulate a physically centralized system [8] and production-level SDN network designs resort to physically distributed control planes [9].

Moreover, everything is maintained centrally at the control plane. Consequently there is no need to configure forwarding plane device manually. The aspect of SDN is to centralize the network control plane. Centralized network control plane leads to innovative approaches to traffic engineering, reducing network energy consumption, and data center network management [10-12]. SDN architectural components include SDN Application (SDN App), SDN Controller, SDN Datapath, SDN Control to Data-Plane Interface (CDPI), SDN Northbound Interfaces (NBI) [13]. SDN architecture has three layers, an infrastructure layer, a control layer and an application layer. Infrastructure layer consists of network devices switches, routers, virtual switches, wireless access point. Control layer consists of SDN controller Floodlight [14], Beacon [15], POX

[16], NOX [17], Open Daylight [18] etc. Application layer includes the applications for configuring the SDN Access control, traffic/security monitoring, energy-efficient networking, management of the network.

The structure of this paper is as follows. Initially, related work has been discussed in section II. In Section III, materials and methods has been described. Section IV, experiment result describes in details. Finally conclusion and future work are mentioned in Section V.

2 Related Works

Software Defined Networking has become powerful technologies which are capable to program network flow paths into flow-table in switches for network control. Data plane and control plane are decoupled in SDN architecture where network is controlled by manipulating flow-table on control plane. Effective use of SDNs for traffic engineering has been described in [19] especially when SDNs are incrementally introduced into an existing network. In [20] provide a profound understanding of the problems related to satisfying global network objectives, such as maximum flow, in environments where the size of the forwarding table in network devices is limited. Dijkstra's algorithm and the modified Floyd-Warshall algorithm have been implemented in order to find shortest path in OpenFlow [21]. In [22] propose an approach to reduce to the rule generation computation cost in networking application by excluding duplicated paths during rule generation time. Network performance between Double Constrained Shortest Path (DCSP) and Dijkstra algorithms in a smart grid communication network with different link bandwidths has been implemented in [23]. In [24] authors implement the extended Dijkstra's algorithm and compare it with the original Dijkstra's algorithm under the Abilene network using Mininet tool.

A study on multi-dimensional resources integration (MDRI) for service provisioning in cloud radio over fiber network (C-RoFN) has been described in [25]. Author's present performance of RIP scheme under heavy traffic load scenario quantitatively evaluated to demonstrate the efficiency of the proposal based on MDRI architecture. In [26] present a novel cross stratum optimization (CSO) architecture in elastic data center optical interconnection. Overall feasibility and efficiency of the proposed architecture experimentally demonstrated on OaaS testbed with four OpenFlow-enabled elastic optical nodes. Authors also compare the result with MFA, ALB, and CSO-DGLB service provisioning schemes in terms of path setup/release/adjustment latency, blocking probability, and resource occupation rate.

3 Materials and Methods

3.1 Research methodology

Proposed architecture has been implemented by using Mininet [27,28] which is inexpensive and quickly configurable network emulator. Mininet is standard Linux based networking emulator where virtual topologies like virtual host, switch and link can be created. It also supports OpenFlow protocol which can be used for computer network based SDN simulation. Mininet is also great way to enhance, share, and experiment with OpenFlow and Software-Defined Networking systems. By single command Mininet creates realistic virtual network, runs collection of end-hosts, switches, routers, and links on a single machine (VM, cloud or native) [29]. Mininet released under a permissive BSD Open Source license which is actively developed and supported.

3.2 Experiment setup

All the simulation has been done by using POX which is Python based open source OpenFlow/Software Defined Networking (SDN) Controller. POX controller provides an efficient way to implement the OpenFlow protocol and used for faster development and prototyping of new network application. POX controller runs different applications like hub, switch, load balancer, and firewall.

Designed network topology consists of seven OpenFlow switch, an OpenFlow POX controller and fourteen hosts where two hosts is connected each of the switch. Host h1, h2 is connected to switch S1 and host h11, h12 is connected to switch S6. In addition, host h3, h4 is connected to switch S2 and host h13, h14 connected with switch S7. Fig. 2 shows designed network topology.

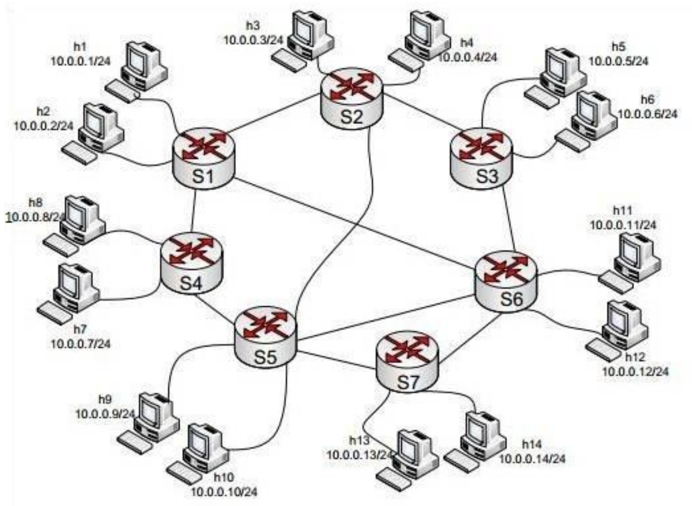

Fig. 2. Network topology

4 Experiment Results

Packet Internet Group (PING) operates by sending Internet Control Message Protocol (ICMP) echo request packet to the target host and wait for reply to check the IP connectivity between defined hosts. If network communication is established, ping tests also determine the connection latency (technical term for delay) between the two computers. Ping can be used for troubleshooting to test connectivity and to determine response time.

In the first evaluation to check the connection is performed on the network ping from host h1 to host h12. The packet from host h1 to host h12 must be analyzed first by the controller before sending a command to install a flow entry to handle the packet. Since at the beginning switch flow tables are empty, first response is always the longest in terms of delay compare to others delay. Ping test result between two host h1 which is connected with switch S1 and host h12 which is connected to switch S6 as illustrated in Fig. 3. Host h1 send 5 echo request packets which are successfully transmitted and received at defined host h12. Transmitted and received echo request packets and the ping statistics are also shown in Fig. 3.

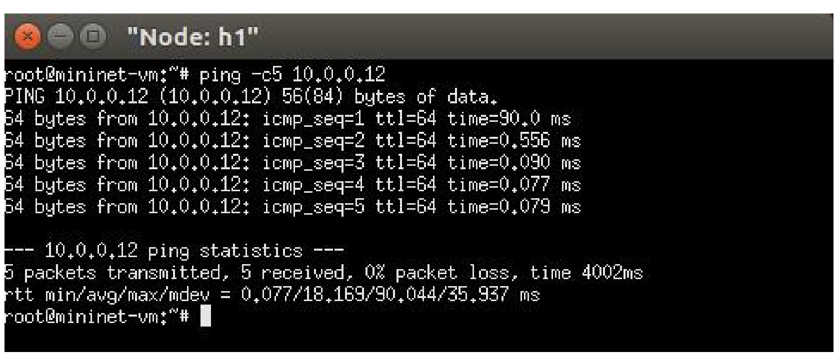

Fig. 3. Ping test result from host h1 to host h12

When the packet reaches to switch S1, it finds the minimum path to reach the destination address host h12 which connected in switch S6. There are several path exists between switch S1 and switch S6. For example, host h1 can communicate via switch S1-S4-S5-S6 and finally reached host h12. In addition, host h1 can also communicate via switch S1-S2-S6 and finally switch h12. Moreover, there is also a direct path from switch S1 to switch S6.

According to Bell-Man Ford algorithm, minimum path use to communicate host h1 and host h12. The shortest path between host h1 to host h12 is the direct path from switch S1 to switch S6. Simulation result illustrated in Fig. 4 shows shortest path (S1=00:00:00:00:00:01 and S6=00:00:00:00:00:06) which is used to communicate between host h1 and host h12.

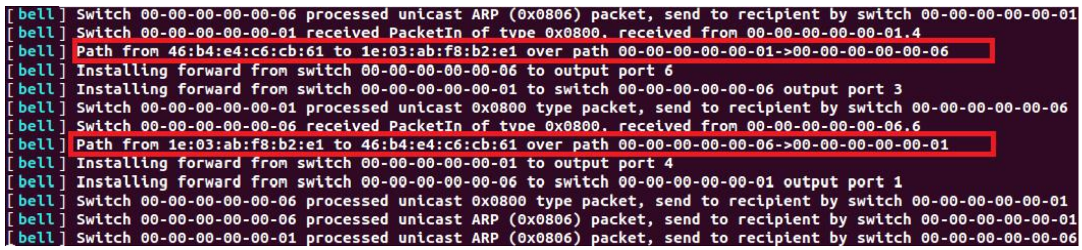

Fig. 4. Computation of Shortest path from host h1 to host 12

In the second evaluation, host h3 which is connected to switch S2 ping to reach the target host h14 which is connected to switch S7. The corresponding ping results illustrated in Fig. 5 shows host h3 sends five echo request packet to the target host h14 and receives corresponding ICMP packet response.

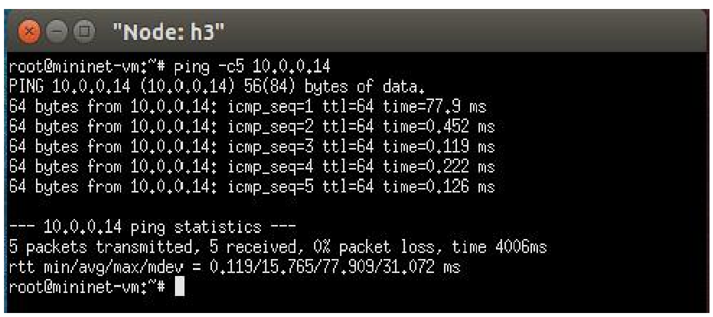

Fig. 5. Ping test result from host h3 to host h14

There are several path exists between switch S2 and S7. Switch S2 may communicate with switch S7 via switch S2-S1-S4-S5-S7 or S2-S2-S6-S7 or S2-S5-S7. Since host h3 and host h14 is connected to switch S2 and S7, according to Bell Man Ford algorithm shortest path use to communicate from switch S2 to switch S7.

Second evaluation result illustrated in Fig. 6 shows the shortest path, which is used to communicate between host h3 and host h14. The shortest path exist between two switch S2 and S7 is used by the Bell Man Ford algorithm where the shortest path is switch S2-S5-S7 (S2=00:00:00:00:00:02, S5=00:00:00:00:00:05, S7=00:00:00:00:00:07).

Network and transportation related analysis have become common practice in many application areas within Geographic Information Systems (GIS) technology. With the advent of GIS technology computation of minimum paths between different locations on a network is one of the major problems in network and transportation analysis. Sometimes this computation has to be done in real time. SDN framework provides a way to solve for the minimum path based on dynamic link statuses through SDN's high network monitoring capability. Our paper implements Bellman Ford Algorithm for computing shortest path over SDN which shows expected outputs.

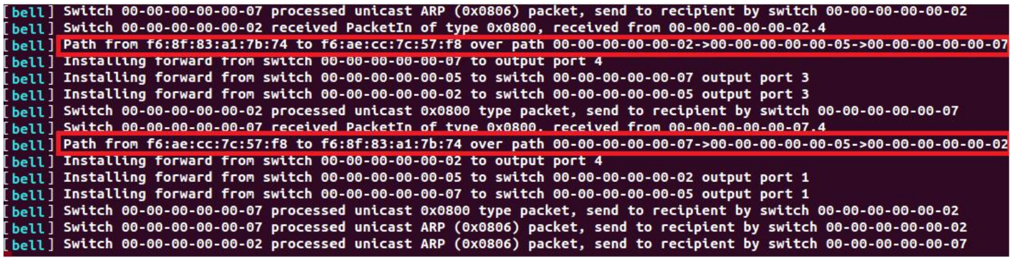

Fig. 6. Computation of shortest path from host h3 to host 14

5 Conclusion

It is well known that computation of shortest paths is an important task in many network and transportation related analysis. A number of different algorithms and a considerable amount of empirical have done to compute least path in existing network. The development, computational testing, and efficient implementation of minimum path algorithms have remained important research topics within related disciplines such as operations research, management science, geography, transportation, and computer science. This paper implements Bellman Ford Algorithm for computing shortest path in Software Defined Networking using POX controller and Mininet emulator. For future works, we will implement other shortest path algorithm in SDN and compare the result with each other to compute the best minimum path finding algorithm.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Diekmann C. Software defined networking.
Available:http://www.net.in.tum.-de/pub/diekmann/sdn2014.pdf
(Accessed: 10-12-2015)

[2] Lara A, Kolasani A, Ramamurthy B. Network innovation using OpenFlow: A survey. Communications Surveys & Tutorials, IEEE. 2014;16(1):493-512.

[3] Kobayashi M, Seetharaman S, Parulkar G, Appenzeller G, Little J, Van Reijendam J, McKeown N. Maturing of OpenFlow and software-defined networking through deployments. Computer Networks. 2014;61:151-175.

[4] Bholebawa IZ, Jha RK, Dalal UD. Performance analysis of proposed OpenFlow-based network architecture using mininet. Wireless Personal Communications. Springer; 2015.

[5] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Turner J. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review. 2008;38(2):69-74.

[6] McKeown N. How SDN will shape networking. Open Networking Summit. Available:http://www.youtube.com/watch?v=c9-K5O qYgA (Accessed: 09-12-2015)

[7] Nunes B, Mendonca M, Nguyen XN, Obraczka K, Turletti T. A survey of software-defined networking: past, present and future of programmable networks. Communications Surveys & Tutorials, IEEE. 2014;16(3):1617-1634.

[8] Koponen T, Casado M, Gude N, Stribling J, Poutievski L, Zhu M, Shenker S. Onix: A distributed control platform for large-scale production networks. Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, In OSDI. 2010;10:1-6.

[9] Jain S, Kumar A, Mandal S, Ong J, Poutievski L, Singh A, Zolla J. B4: Experience with a globally-deployed software defined WAN. In ACM SIGCOMM Computer Communication Review. ACM. 2013;43(4):3-14.

[10] Al-Fares M, Radhakrishnan S, Raghavan B, Huang N, Vahdat A. Hedera: Dynamic flow scheduling for data center networks. In Proceedings of USENIX NSDI '10, 2010. 2010;10:19.

[11] Heller B, Seetharaman S, Mahadevan P, Yiakoumis Y, Sharma P, Banerjee S, McKeown N. ElasticTree: Saving energy in data center networks. In Proceedings of USENIX NSDI. 2010;10(10): 249-264.

[12] Kim H, Feamster N. Improving network management with software defined networking. Communications Magazine, IEEE. 2013;51(2):114-119.

[13] ONF Market Education Committee. Software-defined networking: The new norm for networks. ONF White Paper; 2012.

[14] Floodlight OpenFlow Controller- Floodlight Project. Available:http://www.projectfloodlight.org/floodlight/ (Accessed: 10-11-2015)

[15] Home-Beacon-Confluence. Available:https://openflow.stanford.edu/display /Beacon/Home/ (Accessed: 12-11-2015)

[16] About POX — NOXRepo. Available:http://www.noxrepo.org/pox/about-pox/ (Accessed: 05-11-2015)

[17] NOXRepo. Available:http://noxrepo.org/wp/ (Accessed: 09-03-2016)

[18] The OpenDayLight Platform— OpenDayLight. Available:http://www.opendaylight.org/ (Accessed: 25-11-2015)

[19] Agarwal S, Kodialam M, Lakshman TV. Traffic engineering in software defined networks. In INFOCOM, 2013 Proceedings IEEE. 2013;2211-2219.

[20] Cohen R, Lewin-Eytan L, Naor JS, Raz D. On the effect of forwarding table size on SDN network utilization. In IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE. 2014; 1734-1742.

[21] Furculita AG, Ulinic MV, Rus AB, Dobrota V. Implementation issues for Modified Dijkstra's and Floyd-Warshall algorithms in OpenFlow. In Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition. IEEE. 2013;1-6.

[22] Sim JH, Kim SH, Park MW, Chung TM. Eliminating duplicated paths to reduce computational cost of rule generation by using SDN. In International Conference on Computational Science and Its Applications. Springer International Publishing. 2014;603-613.

[23] Zhao J, Hammad E, Farraj A, Kundur D. Network-Aware QoS routing for smart grids using software defined networks. InSmart City. Springer International Publishing. 2016;360:384-394.

[24] Jiang JR, Huang HW, Liao JH, Chen SY. Extending Dijkstra's shortest path algorithm for software defined networking. In Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific. IEEE. 2014;1-4.

[25] Yang H, Zhang J, Ji Y, He Y, Lee Y. Experimental demonstration of multi-dimensional resources integration for service provisioning in cloud radio over fiber network. Scientific Reports. 2016;6.

[26] Yang H, Zhang J, Zhao Y, Ji Y, Han J, Lin Y, Lee Y. CSO: Cross stratum optimization for optical as a service. IEEE Communications Magazine. 2015;53(8):130-139.

[27] Lantz B, Heller B, McKeown N. A network in a laptop: Rapid prototyping for software-defined networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. ACM, 2010;19.

[28] Mininet – An Instant Virtual Network on your Laptop (or other PC).
Available:http://mininet.org/
(Accessed: 20-13-2016)

[29] Introduction-to-Mininet. mininet/mininet.wiki. GitHub/
Available:https://github.com/mininet/mininet/wiki/Introduction-to-Mininet
(Accessed: 25-02-2016).

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/18458