End-to-End Fault Tolerant Solution using Loop Avoidance Algorithm in SDN

Maheen Iqbal, Syed Waleed, M. Shahbaz Qureshi, Mansoor Ahmed Tahir, M. Irfan Anis

***Abstract*—In this paper, an implementation of an end to end fault tolerant solution for Dynamic traffic recovery has been performed using a centralized controller. SDN develops programmability within a network by providing flexibility towards innovation with variant policy implementation, as vendor specific policy building restricts dynamic application building. The Minimum Spanning Tree (shortest path algorithm) has been implemented providing resolution for loop confrontation and reduction in convergence delays with faster transmission towards the destination. The algorithm not only maintains loop avoidance procedure within a recovery time of 48 to 60 msbut provides fault tolerance at a faster rate for the provision of an alternative route in case of link failure.**

***Keywords*— ONOS; Open V switch; Root bridge; SDN; STP;**

I. INTRODUCTION

INrecent years, a drastic advancement has been observed in the Software defined Networks (SDN), introduced by the concept of decoupled planes [1]. SDN Architecture decoupled the control and data plane by the provision of communication between the forwarding elements and the centralized control unit through open flow protocol. The centralized network controlling mechanism in SDN that is capable of orchestrating traffic management through integrating low-level forwarding policies into switches has replaced the traditional distributed computing [2].

Network operators to provide reliable and real time communication services are considering failure management as a fundamental tool. To overcome the failures of network components the integration of failover mechanism with networks [3] is being considered. The progression within the technology has taken a higher level since the development of tagged packets. The tagged packets involve the creation of detecting nodes for failure recovery and the purpose of retransmission [4]. The larger and complex networks operating through bridging policy building within a software defined network provides the opportunity to work on more efficient and diversified patterns.

The SDN technology have worked in maintaining mechanisms for fault tolerance [5] in case of a link failure by reducing the convergence time and provision of reduction in transmission time from the source towards the destination. The path restoration within a network by end to end path provisioning [6] is the main concern and SDN is capable to provide the platform to bring diverse policies and flow table entry modification within the Open V switches maintained by a virtual environment.

Open Flow technology introduces the implementation of fast recovery dynamic algorithm building as the conventional pattern of loop avoidance bridging policy named as Spanning Tree Protocol (STP) that causes a particular problem to arise. STP operates on the Root Bridge selection mechanism by selecting the bridge id of the oldest switch by default. It introduces the requirement to manually update the Root Bridge and reduces the priority. It creates delay and reduction in the transmission rate of the data towards the destination that is necessary for reliable networks.

The Open Networking Operating System (ONOS) provides the approach for improved performance and low latency rate transmission [7]. ONOS modified the network state into a network view. A brief study has been conducted in [8] for SDN switch platform that it requires to enable users to develop software and service flexibility. The [9] discusses the current alternatives for implementation and testing of SDN-based protocols and services. In [10] central controller architecture has been redesigned in order to make the controllers and network resilient to failures and eliminate fate-sharing relationships. The Optimization has been done in automatizing the failure recovery within the SDN environment [11] where the controller is in observation mode before the occurrence of failure. The [12] presents a starting point towards the availability of SDNs and emphasize upon the high availability during network failures. The [13] provides a service restoration over an end to end SDN for PON and for fast traffic recovery and demonstration of PON services. However, a scalable and broad approach is required to provide fault tolerance at faster rate.

In this paper, a fault tolerant solution for autonomous failure recovery has been implemented with Open Flow protocol capable of recovering from a link failure using an alternative path. This demonstrates the effectiveness of the implemented mechanisms by emulating undirected mesh topology. The MST algorithm through redundant links provides a shortest path with reduced convergence delay and offers more efficient bandwidth with an optimum path without shutting down any links. This gives faster transmission of data with immense fault tolerance creating a better approach of migration within the network.

The paper is structured as follows: section II explains the proposed architecture depicting the flow within the network and its maintenance between the switches and the central control unit with the provision of policy implementation. The Section III shows the implementation of loop avoidance algorithm using Pox controller and the section IV concludes the paper by discussing the simulation results.

II. PROPOSED PATH DECISION SELECTION ARCHITECTURE

The Architecture is composed of de-coupled planes as in [14] comprising of a control and data element. The centralized controller performs the role of monitoring over the programmed Open V switches. The virtual environment is maintained by Mininet Emulator through a 6633 TCP port that is utilized for the Open flow traffic. The architecture is based upon two interfaces known as the upperbound interface and the lower bound interface. We have implemented the lower bound interface by the provision of network connectivity between the switching nodes and the controller through an Open Flow interface.

In Fig. 1 the controller is layered into three major aspects regarding the process that is being followedduring the implementation of within the switches is provided when a request is being decision making for the conduction of end to end data provisioning. In this process the controller is working in a reactive manner as the path along with the flow table entries maintenance generated for a communication path by the switch.

The switching fabric is maintained by developing a communication link with the controller and all the flow modification is provided on the basis of the encapsulated request sent by the data forwarding elements. Their further actions are established within the network flow on this basis. The network has been formulated in a mesh topology with the provision of redundant links for failure recovery and to ensure fault tolerance in the network. The decoupled plane architecture has introduced the creation of parallel integrated systems by providing a centralized view for higher level configuration. The unequal path formation of the implemented topology has been maintained to represent efficient decision making and policy building within the switching nodes with lesser amount of computation being performed at the nodes.

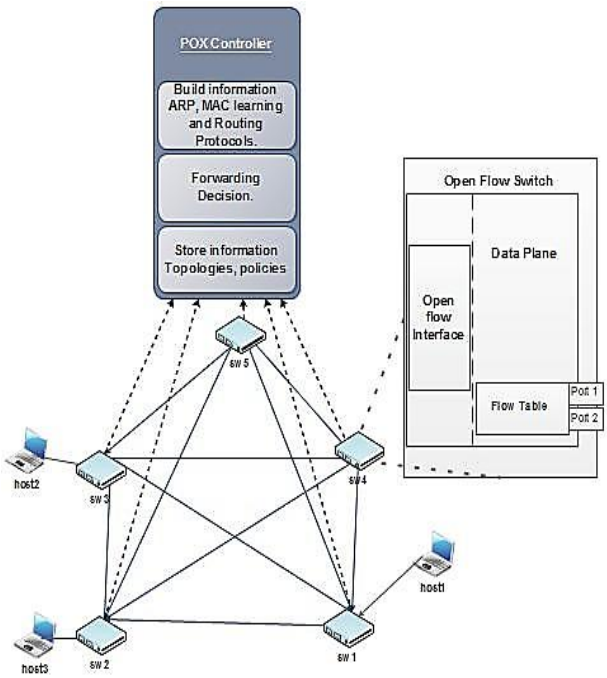

Fig. 1.Controlled switch conceptual diagram.

A. Path Restoration:

The provision of alternate redundant paths within a network as further mentioned in [15] is formulated to overcome the problem of disconnection of communication between end to end switching nodes. Therefore, for efficient path restoration an optimum algorithm building within the network has been initialized so that latency and convergence delays could be minimized. The communication between network interface controller and the switches has been shown in Fig. 2 in order to explain the signaling of Open V switches in case of a failure. The proposed Architecture provides the platform for faster recovery in time duration of within one or two seconds. The implementation of proposed MST Protocol maintains a reliable restoration of links in case of a failure.

B. Shortest Path Routing:

The routes are being maintained on the basis of the minimum path cost towards the destination. The calculation of MST commonly known as spanning tree of minimal cost is utilized as a failure recovery algorithm. The topology that has been maintained within a data center in the formation of a switching fabric requires faster deployment of policy building along with minimum delay routes to create an efficient controlled environment. The path is not being maintained by the controller after the topology discovery of the network links but is defined on the encapsulated request being generated by the switching node towards the destination. The readings accumulated in Fig. 3 represent the calculated throughput along with the bandwidth that is being utilized between two end hosts after a minimum path has been discovered by the controller.

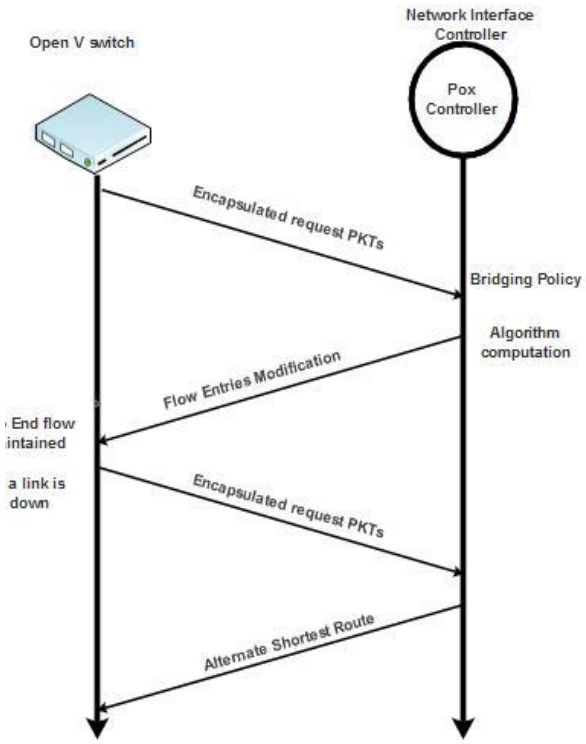

Fig. 2.Signaling of port down event.

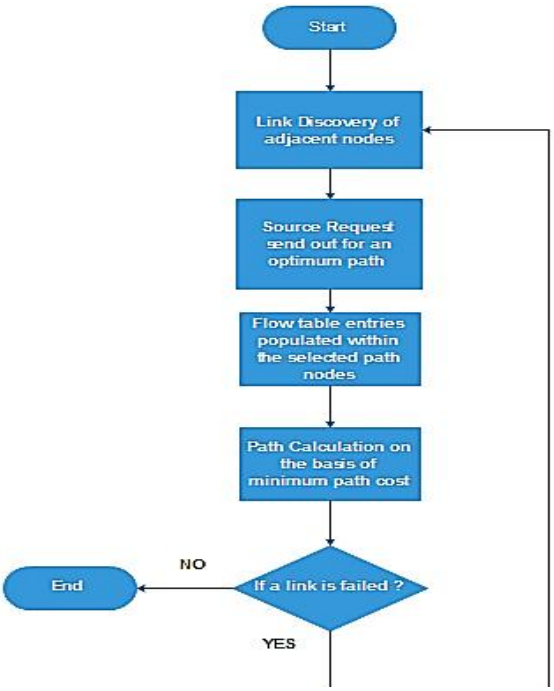

Fig. 3. Flow chart for optimal path selection

III. MINIMUM SPANNING TREE PATH SELECTION

The implemented mesh topology along with the formulated loop structure is a problem that usually arises within the data centers but the introduction of an efficient algorithm is the purpose of Software Defined Networking. The Pox controller that has been integrated with the network is programmed in python scripts for building intelligent path selection procedures.

The MST algorithm is based on the link metrics between two end nodes and provides the shorter and higher bandwidth route towards the destination. The algorithm is represented within the Mininet Emulator Program with the default standardization link metric of 10 Mbps.

The MST is different from the other various shortest path algorithms as in Fig. 4 it calculates the minimum path cost keeping in consideration all possible combinations of the switching nodes. Focusing on the shortest path from the source towards the destination without keeping in thought the individual minimum link cost between two adjacent switching nodes.

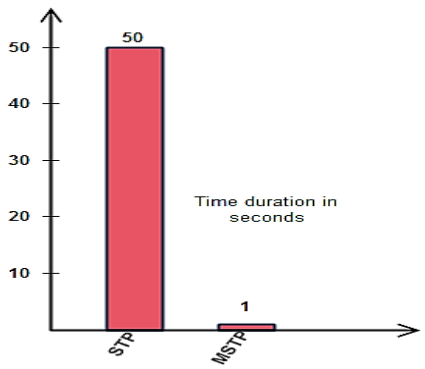

Fig. 4 Path restoration algorithm time duration

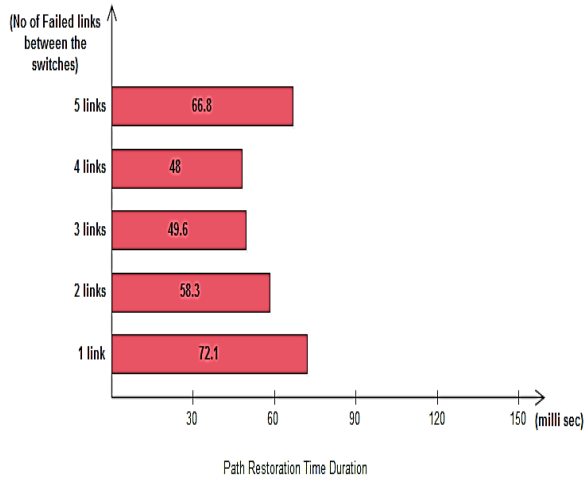

Fig. 5 Time duration statistics for link failure

During the Process if the path discovery introduces two similar cost paths so the first discovered path would be considered automatically until and unless that link is failed so the other link would be brought up to maintain a hindrance free end to end path connectivity.

The emulated topology structure maintained within the Mininet emulator provides the flexibility for the deployment of real time scenario policies with lesser complexity as compared to physical switching node network. The controller within this topology guides the path and maintains the avoidance strategy. The Source switching node initializes the request towards the destination which is populated within the controller and flow table entries are being initiated only along the selected path. The controller calculates the optimum path within few milliseconds with lesser amount of delays. The flow chart for optimum path selection has been shown in Fig. 5 creating a clear view of the entire process.

A. MST (Minimum Spanning Tree) Algorithm:

1. For each u in G: D[u] = 0; P[u] = ∞
2. D[source] = 0
3. While (Topology. Nodes > 0):
4. For each v in Topology. Nodes: remove (v)
5. For each u having adjacent v nodes:
6. If D[u] + weight < D[v];
7. D[v] = D[u] + weight;
8. P[v] = u;

The convergence time of an STP can take up to 50 seconds to move from blocking, listening, learning and then to forwarding accordingly but in Fig. 6 the MST algorithm provides the shortest route towards the destination in duration of 48 to 60 milliseconds which would be approximately 1 to 2 seconds. The algorithm first selects the path and then compares it with the next one between the other adjacent switching nodes and from them both the shortest and the one with the higher bandwidth is selected. In case of paths with same bandwidths, it selects the first measured path as the best optimum route. This makes it more suitable enough for the practical deployment of faster switching network data centers.

Fig. 6 Initialized switching node connections

The principle of least hop count is being applied for all the adjacent nodes starting from the source node. If there is minimum distance from that node to the adjacent one then this node would be stored in the database and like this all the nodes to the destination having the least adjacent distance would be considered and the entries would be populated within that mentioned path that is being discovered.

The path restoration in case of multiple link failures and their time duration by the implementation of MST algorithm is shown in Fig. 7. This represents that the controller takes the maximum time when all the shortest path routes are failed and it needs to calculate an alternate path that is considered as the longest amongst all so path calculation totally depends on the route that needs to be calculated after the link failure and not on the number of links that are failed within the network.

IV. PERFORMANCE EVALUATION

Undirected graph topology having redundant paths introduces a loop within the network. Switch connections in the formation of a web have been presented along with end to end data communication. The traffic recovery mechanism utilizes the route for efficient path selection.

The simulated results provide a clear view that within 10 Mbit/sec links that are established within the network we are having an estimate delay of 0.3 seconds for end to end packet forwarding within a loop structured network of 10 switching nodes.

The simulated outcomes provide the open flow discovery of the switches within the network and the connectivity of the port ids creating a link between both of the switching nodes. In Fig. 8 All the connected switches within the topology are being discovered by the open flow discovery protocol.

TABLE 1 SIMULATION RESULTS

Parameters	Readings
Throughput Results	13.8 MBytes
Time Duration	10 sec
Bandwidth on end Links	9.42 Mbits/sec
Number of Switching Nodes	10
Number of Hosts	3
End to End Delay	0.246 ms

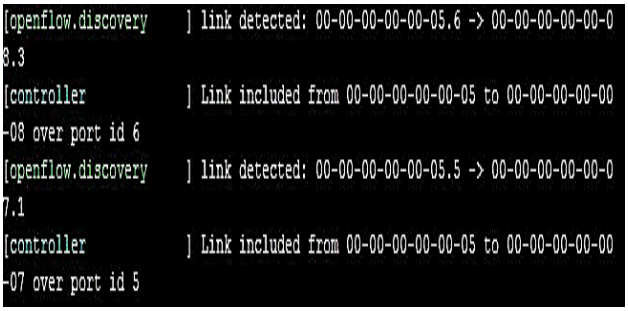

Fig. 7 Link discovery protocol analysis

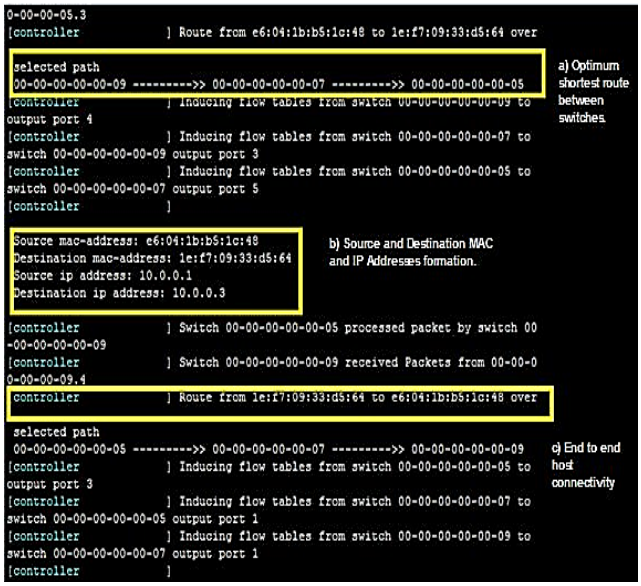

Fig. 8 (a) Provision of shortest route between the end nodes, (b) representation of source and destination. Mac and IP addresses, (c) Host communication through MST algorithm

In Fig. 7 the links between both the adjacent switches are being detected along with the port that is connecting both of the switches while maintaining a link. The link detection is followed by the link inclusion which is maintained every time a new switch or a link is being initiated in the network.

In Fig.8 (a) the selected minimum cost path has been displayed by the controller after inducing flow table entries within the nodes selected for the route. Below are also shown that the IP addresses of the destination hosts are being populated within the controller by flooding the packets on all the edge switches.

The packet description in Fig.8 (b) that is being transferred from one end to another has also been provided with source and destination mac addresses and IP addresses. The host communication in Fig.8(c) provides end to end connectivity. The ARP packets have been utilized in case of not having the required mac address of the destination by the destination host itself instead of flooding the packets within the entire network, but all these policy building have been conducted by the controller. In case of any link failover between the switching nodes, traffic recovery and path restoration is maintained for end to end data provisioning.

The path performance is estimated on the basis of bandwidth utilization and faster link selection between the nodes while not letting the nodes other than the selected

path in the network to be occupied for any kind of service and forwarding tasks.

V. CONCLUSION

In this paper we have demonstrated the decoupled architecture of the Software Defined Networking by implementing a shortest path bridging policy. The policymaintains a route for faster communication taking convergence time within a second as compared to the conventional STP implementation. The implementation of logical Open V switches within an emulator in a network abstracts the switches into a logical unit, creating mapping of the switches less complex than the physical switch. The proposed controller not only transfers the traffic to the calculated path but also performs the processing of packets being conducted by the switches. The technology of programmability within the network is evolving at a faster pace. The networks have been enhancing along with their complexity so a single controller would be considered as a single point of failure and for the network failure preservation the concept of a backup controller would be considered. The proposed algorithm in SDN architecture provides the implementation of failure detection presenting a complete recovery solution from multiple fault domains.

REFERENCES

[1] Sachin Sharma, Dimitri Staessens, Didier Colle, Mario Pickavet and Piet Demeester," A Demonstration of Fast Failure Recovery in Software Defined Networking,"TridentCom, 2012.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Mar. 2008.

[3] Antonio Capone, Carmelo Cascone, Alessandro Q.T. Nguyen, BrunildeSans`o," Detour Planning for Fast and Reliable Failure Recovery in SDN with OpenState," arXiv:1411.7711v2 [cs.NI], Sep 2015.

[4] Berde, Pankaj, et al. "ONOS: towards an open, distributed SDN OS."Proceedings of the third workshop on Hot topics in softwaredefined networking. ACM, 2014.

[5] Kuźniar, Maciej, et al. "Automatic failure recovery for software-defined networks." Proceedings of the second ACM SIGCOMM workshop onHot topics in software defined networking. ACM, 2013.

[6] Ma Wei Zhe1, Jiang Yun Dou1, Dong Hong Yu2 , Xue Xiao Ming1 , Lin Zhong Qiu3 , Zou, Yu4," SDN-based Solution of Path Restoration for Smart Grids," 4th International Conference on Computer, Mechatronics, Control and Electronic Engineering , ICCMCEE, 2015.

[7] Pankaj Berdey, Matteo Gerolaz, Jonathan Harty, Yuta Higuchi, Masayoshi Kobayashi, Toshio Koide§, Bob Lantzy, Brian O'Connory, PavlinRadoslavovy, William Snowy, Guru Parulkar," ONOS: Towards an Open, Distributed SDN OS, HotSDN'14, August 22, 2014.

[8] Hata, Hiroki. "A study of requirements for SDN switch platform." Intelligent Signal Processing and Communications Systems(ISPACS), 2013 International Symposium on.IEEE, 2013.

[9] Nunes, Bruno, et al. "A survey of software-defined networking: Past, present, and future of programmable networks." CommunicationsSurveys& Tutorials, IEEE 16.3 (2014): 1617-1634.

[10] BalakrishnanChandrasekaran, Theophilus Benson," Tolerating SDN Application Failures with LegoSDN," HotNets-XIII, October 27–28, 2014.

[11] MaciejKu´zniar, Peter Perešíni, NedeljkoVasi´c, Marco Canini, DejanKosti," Automatic Failure Recovery for Software-Defined Networks," HotSDN'13, August 16, 2013.

[12] Aditya Akella, Arvind Krishnamurthy," A Highly Available Software Defined Fabric," HotNets-XIII, October 27–28, 2014.

[13] SeamasMcGettrick; Frank Slyne; NattapongKitsuwan; David B. Payne, "Experimental End-to-End Demonstration of Shared N: 1 Dual Homed Protection in Long Reach PON and SDN-Controlled Core", Tu2E.5, in Proc. OFC'15.

[14] Shahid, Adnan, Jinan Fiaidhi, and Sabah Mohammed. "Implementing Innovative Routing Using Software Defined Networking (SDN)." International Journal of Multimedia and Ubiquitous Engineering 11.2 (2016): 159-172.

[15] Jiang, Jehn-Ruey, et al. "Extending Dijkstra's shortest path algorithm for software defined networking." Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific. IEEE, 2014.

AUTHORS

Maheen Iqbal
High Performance Optical Research Group,
FEST Department, Iqra University

Syed Waleed
High Performance Optical Research Group,
FEST Department, Iqra University

M. Shahbaz Qureshi
High Performance Optical Research Group,
FEST Department, Iqra University

Mansoor Ahmed Tahir, mansoor.tahir@iqra.edu.pk
High Performance Optical Research Group,
FEST Department, Iqra University

M. Irfan Anis, mirfananis@iqra.edu.pk
High Performance Optical Research Group,
FEST Department, Iqra University