

# Enhancing Recovery Scheme in Software-Defined Network using Segment Routing

Ohmmar Min Mon, Tha Pyay Win, Myat Thida Mon

*University of Information Technology, Yangon, Myanmar*  
*ommm@uit.edu.mm, thapyaywin@uit.edu.mm, myattmon@uit.edu.mm*

## Abstract

*Software-Defined Networking (SDN) based architectures, such as OpenFlow, offer a chance to design such a control plane. The SDN controller performs all complex functions, including routing and security checks. Segment Routing is a new emerging technology that enables source routing. Segment Routing (SR) technology can enable networks to achieve scalable SDN and traffic engineering solutions. The problem of computing bandwidth guaranteed paths for given flow requests is a challenge. This proposed system presents fast rerouting algorithm exploiting the link or node failure to compute the optimal backup path (active backup path). In this system, failure in network link or node is detected by using the OpenFlow switches and recomputed the new optimal path using fast reroute with segment routing. The goal of this algorithm is to reduce the failure recovery time with low complexity and decreases the packet overhead in case of failure.*

## 1. Introduction

In recent years, it takes great interests in SDN as a way to evolve today's best effort oriented networks to achieve full service, flexible and scalable networks. Software-Defined Networking (SDN) is an emerging architecture that separates control plane from the data plane. Control plane functionalities are handled by an entity called controller which can centralized and distributed. Controller communicates networking devices to

collect information from them and also to push configuration information to them.

Segment Routing (SR) is a new emerging routing technique and can be used in MPLS network that enables source routing. In the segment routing domain, nodes and links are assigned Segment Identifiers (SIDs). It allows the network operator to be forwarded a path from ingress node to egress node. To specify a particular path through the network, these Segment Identifiers allow an ingress node to select a path through the network using either a single SID to represent the destination node or using a series of SIDs, called a segment list. These segment lists can be encoded as one or more Multi-protocol Label Switching (MPLS) labels or as one or more IPv6 addresses. A segment can represent an instruction which can be topological or service-based.

The proliferation of the real time and loss sensitive applications such as VoIP services, video services, and stock exchange data services today is far less tolerant to packet loss. Survivability is a requirement in that services interrupted by equipment failures must be recovered as quickly as possible. Despite continuous technological advances, failures cannot be completely avoided even in well-maintained networks. Therefore, developing fast failure recovery schemes has great practical significance. The failure recovery mechanism eliminates the significant delay by computing the backup path and installing the forwarding rules into the switch on the primary and backup paths. Once the failure along the primary path is detected using high quality failure-detection mechanisms such as Bidirectional Forwarding

Detection (BFD), the OpenFlow switch forwards the packets along the backup path.

When SR is combined with Fast Reroute technique, the network is deterministic, recovery is fast and predictable. Rerouting may take as long as seconds to complete. When a failure is detected, the affected packets are immediately forwarded through backup paths to shorten the service disruption. Internet Service Providers demand for enhanced network performance, increase the need for network resources. The main challenge is how to achieve fast recovery with low complexity and resource usage. The second challenge is to find the optimal path for large networks when failure occurs. This system presents a new design for a path computation that makes use of segment routing to achieve more flexibility for backup path computation. It also demonstrates that the path computation can respond to path queries with acceptable latency. Finally, this system describes how to enhance the failure recovery in SDN with existing Internet applications.

The rest of the paper is organized as follows. Section II describes the related works of segment routing and path calculation. Section III presents the research background. Section IV proposes the overview of rerouting and proposed algorithms to perform path computation. Section V concludes the paper.

## 2. Related Work

Failure management in SDN is a topic that has been already explored by the research community. In [1], Clarence Filsfls, Nagendra Kumar Nainar, Carlos Pignataro, Juan Camilo Cardona, Pierre Francois present an introduction to the SR architecture, highlighting its simplicity, and scaling properties. The author discussed use cases such as Traffic Engineering, showing that SR gives fine-grained control over paths without increasing control-plane overhead at transit nodes. Service Function Chaining has been illustrated using SR as a way to execute a service chain without impacting data-plane resource availability. Finally, networking features can be

made resilient by relying on the basic building blocks of the architecture.

A scheme to find a label encoding algorithm for path and segment list computation in Segment Routing networks is presented in [2] by F. Lazzeri, G. Bruno, J. Nijhof, A. Giorgetti, and C. P. The algorithm guarantees optimal path computation and minimum label stack depth. The algorithm also accounts for equal-cost multiple paths and multiple constraints.

Hyunhun Cho, Jinhyung Park, Joon-Min Gil, Young-Sik Jeong and Jong Hyuk Park [3] construct an application that calculates optimal paths for data transmission and real-time audio and video transmission, which is a major issue of the network service providers. The SDN routing computation (SRC) application is efficiently designed and applied in the multi-domain network for efficient use of the resources, selection of the optimal path between the domains and optimal establishment of end-to-end connections.

In [4], Minglei Fu, Wen Dong, Zichun Le, Xingshu Sun proposed a BFD-triggered failure detection mechanism combining with Fast Reroute (FRR) in Optical burst switching (OBS) networks. BFD protocol can set the transmitting period as well as the detection period according to requirement of OBS networks and node failure (or link failure) can be detected in less than 30ms. Besides, the fast reroute technique is used as the restoration mechanism when the failure has been detected by BFD protocol.

In [5], Nor Masri Sahri, and Koji Okamura discussed fast and efficient failover mechanism for redirecting traffic flows to more optimal backup path when there is a link failure or congestion problem. This system introduces a switch flow entry expiry mechanism to reroute traffic to backup path to reduce the network restoration time.

A novel routing scheme called Accumulative-Load Aware Routing for Software-Defined Networks was presented by Trong-Tien Nguyen and Dong-Seong Kim [6]. In SDN, ALAR makes routing decisions based on a specific link cost function which has the key factor being accumulative load on a link. Leveraging this information, ALAR makes

routing decisions based on a specific link cost function which has the key factor being accumulative load on a link. This approach enhances the utilization of network resources, minimizes the probability of congestion and through which the average end to end delay is improved.

Seungbeom Song, Jaiyong Lee, Kyuho Son, Hangyong Jung and Jihoon Lee presented that link utilization is calculated in SDN controller and recalculated rerouting algorithm is applied to switches which would be configured by using OpenFlow configuration protocol [7].

Yilan Liu, Yun Pan, Muxi Yang, Wenqing Wang, Chi Fang, Ruijuan Jiang described an algorithm called Bandwidth-Constrained Multi-Path Optimization (BCMPO)[8] with Genetic Algorithm and Dijkstra's algorithm performing the optimization of routing paths and more flexible control and better allocation of resources.

Cihat Cetinkaya, Erdem Karayer, Muge Sayit, Cornelius Hellge [9] proposed an optimization model aiming to obtain maximum throughput for DASH services by selecting the optimal paths for video packet flows over SDN. It finds the most suitable path by considering the available bandwidth, bitrates of the current segment, competing flows and the path length.

In [10] the author, Diyar Jamal Hamad, Khirata Gorgees Yalda, and Ibrahim Tanner Okumus introduced how to get traffic measurement statistics from network devices in an SDN environment.

In practice, this system may need to find the optimal paths in SDN. The results show that fast rerouting algorithm in the SDN can optimize the routing paths exploiting the failure risk (link, node) and enables flexible management and scalability.

### 3. Research Background

In this section, this system presents an example of how segment routing works. Segment Routing is a new emerging routing architecture developed within the Internet Engineering Task Force (IETF). It enables to use

non-shortest paths by specifying alternative routes. It is typically associated with a centralized control plane implementation, where the SR controller is in charge of all the decisions in the network. The controller is in charge of setting up the edge-to-edge services, by configuring the ingress and egress PE nodes for a given flow.

In segment routing, packets are forwarded through the shortest path from the source to destination. Nodes and links are assigned Segment Identifiers called segments. In the case of a link (i.e. adjacency) segment, the shortest path to the upstream node is taken and then that link is crossed. In the case of Equal Cost Multipath (ECMP) between two segments, all the ECMP paths are used (for different packets).

SR can be integrated with SDN. The SDN controller can learn the network topology and the real time state information. Using this information, the SDN controller can calculate the best network path based on a set of predefined criteria. To implement SR, the SDN controller needs only to send an ordered list of labels to the source router which should be inserted in the header when the packet is locally forwarded. This provides a much more scalable and simple solution for traffic engineering. Millions of applications or flows can have millions of different network paths.

Optimization of recovery paths can be achieved as the current network state is considered before recovery path selection. The major contribution of this system is to present a new fast rerouting algorithm to perform optimal path computation in SDN. The algorithm also discovers the backup path for carrying traffic of failed link or node. This study uses reserved backup paths and a fast rerouting solution using SR and SDN to find a new optimal path with low message overhead, fast recovery time and sufficient bandwidth. When the failure is detected, the controller calculates the backup path or alternative path against failure of link or node. Unlike the conventional network, it collects information of the switches. With this information, it selects an active backup path. The active backup path with sufficient bandwidth really operates to the recovery procedure to

enhance the backup path computation. The inactive backup path does not operate and reroute the traffic of the primary path.

## 4. Research Methodology

The two strategies are proposed in the system that can provide fast rerouting. The first solution tries to minimize the number of segments used. When failure occurs, it prefers to forward packets to switches closer to the source. This allows packets to traverse the network while causing less overhead. The overhead is the cumulative number of segments used at each hop along the path. The second solution is to compute the best path with minimum bandwidth constraints.

SR uses IGP (IS-IS or OSPF) extensions to distribute segments without a separate protocol such as LDP or RSVP-TE. Thus, scalability of transit nodes is greatly improved, since MPLS LSPs state information is not required. Unlike MPLS, there is no need to maintain path state in segment routing except on the ingress node, since packets are routed based on the list of segments that they carry. In MPLS, the first LSR does a routing lookup, inserts the label for the next hop. Then at each LSR, it swaps the label for the next hop. In SR only the first router inserts the segment that will be maintained until it reaches its destination.

The SDN controller knows the network topology and the current state information. Using this information, the SDN controller can pre-calculate multiple paths for load balancing. This provides a much more scalable and simple solution for traffic engineering. In this study, each time the users request the flows, switches inform the controller about the request. The main function of the controller is to provide the construction of the network topology. It downloads a configuration to the switches in order to enable the routing functions on each of them. In order to get switch statistics, controller sends an `OFPortStatsRequest` to relevant switch and the switch will reply with `OFPortStatsReply` its statistics. Port statistics

give more information about the state (both transmitted and received) such as number of dropped packets, bytes, errors and collisions. The controller obtains information for every flow that follows the same path.

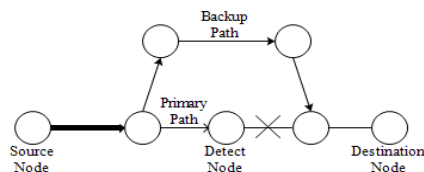
### 4.1. Recovery Scheme

The rerouting model establishes the recovery path only after a failure has occurred. It can be divided into two: establish-on-demand and pre-qualified. The first one, establish-on-demand calculates and establishes a recovery path only after a fault is detected. The second one, pre-qualified provides that a recovery path is already calculated but is only established when a fault occurs. It has the advantages of efficient resource utilization as bandwidth is not reserved and speed up recovery as path selection is completed before failure occurs. Table I gives a comparison of the recovery models.

**Table 1. Recovery model**

Model	Path Type	Path Setup Point	Time	Utilization Optimization
Re-routing	Pre-Qualified	After fault	**	***
	Establish-On-Demand		*	****

When a link or node fails, the affected network traffic must be rerouted via other alternative paths in order to reach its destination. In figure 1, when the middle node fails, the node on its left has to propagate back the error message to the source and then switches to the backup path.



**Figure 1. Example of failure recovery**

Firstly, for rerouting, the network should be able to detect a failure. Node or link failure is usually reported at both upstream and downstream nodes of the failure. Once the failure along the primary path is detected using high quality failure-detection mechanisms such as Bidirectional Forwarding Detection (BFD), the OpenFlow switch forwards the packets along the backup path in SDN.

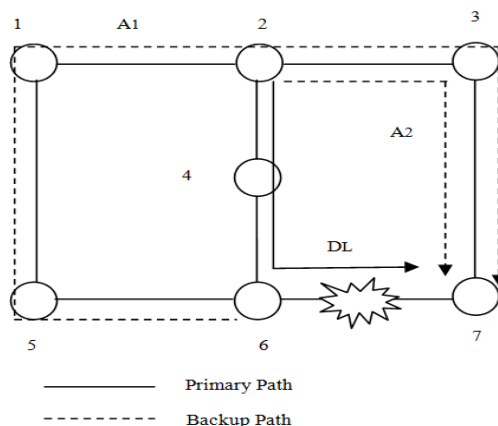
Current OpenFlow implementation on Ethernet was not designed with high requirements on availability and failure detection. On the data-link layer, multiple failure detection protocols exist, such as the Spanning Tree Protocol (STP). In this system, Bidirectional Forwarding Detection (BFD) [4], which has proven to detect the failure, will be used. The (BFD) protocol implements a control and echo message mechanism to detect liveness of links or paths between preconfigured end-points. Each node transmits control messages with the current state of the monitored link or path. A node receiving a control message, replies with an echo message containing its respective session status. A session is built up with a three-way handshake, after which frequent control messages confirm absence of a failure between the session end-points. The protocol can be used over any transport protocol to deliver or improve failure detection on that path.

For fast rerouting, each node detecting a failure on its interface activates all the backup paths that protect the primary paths. Although some are active, some backup paths (inactive backup paths) do not participate to the recovery of the affected communications because the traffic was already redirected by upstream routers onto other backup paths (active backup paths).

In data communication, when a link or node in the path from a sender to a receiver fails, the users at the receiving host would have experienced service interruption until the path is repaired. The duration of the interruption is called the fault recovery time, the time interval between the arrival of the last bit at the receiver before the failure and the arrival of the first bit at the receiver after the path is repaired. The total fault recovery time is defined by equation (1).

$$t_{recovery} = t_{detect} + t_{comp} + t_{switchover} + t_{processing} \quad (1)$$

where,  $t_{detect}$  is the time taken by a node (say D) to detect failure,  $t_{comp}$  is the interval time when the controller receives the notification message when a new backup path has been found successfully,  $t_{switchover}$  is the interval time when the controller has switched the data traffic from the failing path to the new path and  $t_{processing}$  is the processing time need to send data between source and destination.



**Figure 2. Proposed topology**

Let us consider an example. In figure 1, a primary path DL (2→4→6→7). To protect this primary path against the failure of link 6-7, this system can use the backup path A1 (6→5→1→2→3→7) and backup path A2 (6→4→2→3→7) in figure (2). Both the backup paths A1 and A2 become active after the recovery. Figure (4) shows the flow chart of the algorithm. If requested bandwidth is greater than bandwidth constraint then select path. If not, remove all links that have less than the bandwidth constraint. Then it changes all the routes passing through that switch into new route which bypasses that switch. So it can block any new data inserted into that link or node which failure occurs. The controller checks the bandwidth and then modifies flow entry in all switches and replaces with new backup path.

## 4.2. Proposed System

This section presents a new Bandwidth Guaranteed rerouting Algorithm for SDN Networks. To explain Bandwidth Guaranteed rerouting Algorithm, consider a network with  $n$  nodes. When a new traffic flow has to be established, a request is issued to the controller that computes the path. For the computation of the path, the controller needs to know the current network topology and links reserved bandwidth. Consider a SDN network to be undirected graph  $G(V,E)$  where  $V$  and  $E$  are set of nodes and links respectively. Let a request  $SRP = [s,d,b]$  which specifies the source, the destination and the amount of bandwidth required respectively, where  $s,d \in V$ .

The proposed system uses the following notation:

BP= backup path

SRP= segment routing path

PP= primary path

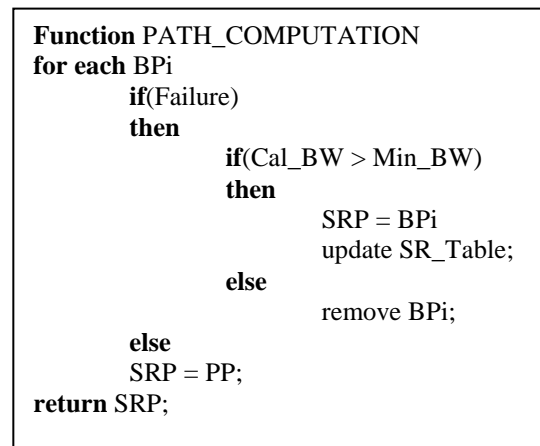
Cal\_BW= calculated bandwidth

Min\_BW= minimum bandwidth

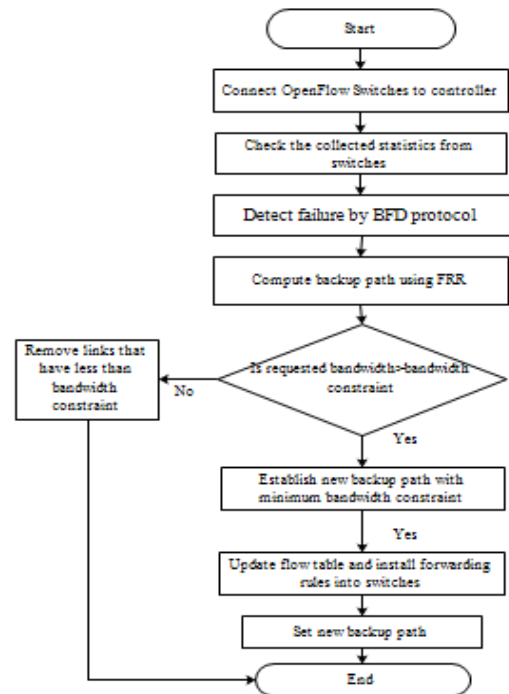
The algorithm is reported in Figure (3):

Input: A set of feasible paths  $SRP = [s,d,b]$  to each affected destination for each protected link and network topology

Output: New best path with minimum bandwidth constraint



**Figure 3. Path computation algorithm**



**Figure 4. Flow Chart of Algorithm**

## 4.3. Expected result

The proposed algorithm attempts to guarantee packet delivery after link or node failure. By implementing the fast reroute algorithm on SDN and SR, the proposed system will minimize packet overhead and number of segments used. This technique can obtain 100% network coverage after link failure or node failure with a small cost of packet overhead.

## 5. Conclusion

Software-Defined Networking is predicted as a means of using more efficiently network resources and dynamically adapting the routing configuration over time. In this paper we have presented a new failure management framework for SDN. This system introduces a design for

more flexible path computation which can compute the paths for end systems. In this paper, this fast rerouting algorithm will be more effective because it reduces the failure recovery time with low complexity by using segment routing and decreases the message overhead by taking advantage of backup paths in OpenFlow Software-Defined Networks. This system will show the performance optimization of network resources efficiently and evaluate the performance of this failure rerouting mechanism. And this failure rerouting mechanism will be faster recovery time, lower message overhead for all QoS constraints. Finally, this system also implements an open-source implementation of the MPLS version of Segment Routing for SDN for new optimal paths with sufficient bandwidth.

## References

- [1] Clarence Filsfls, Nagendra Kumar Nainar, Carlos Pignataro, Juan Camilo Cardona, Pierre Francois, "The Segment Routing Architecture, 2015 IEEE.
- [2] F. Lazzeri, G. Bruno, J. Nijhof, A. Giorgetti, and C. P., "Efficient label encoding in segment-routing enabled optical networks," in Proc. ONDM, May 2015.
- [3] Hyunhun Cho, Jinhyung Park, Joon-Min Gil, Young-Sik Jeong and Jong Hyuk Park, "An Optimal Path Computation Architecture for the Cloud Network on Software-Defined Networking", May 2015, [www.mdpi.com/journal/sustainability](http://www.mdpi.com/journal/sustainability),
- [4] Minglei Fu, Wen Dong, Zichun Le, Xingshu Sun, "BFD-triggered failure detection and fast reroute for OBS networks", 2009 SPIE-OSA-IEEE
- [5] Nor Masri Sahri, and Koji Okamura, "Openflow Path Fast Failover Fast Convergence Mechanism," Proceedings of the Asia-Pacific Advanced Network 2014 v. 38
- [6] Trong-Tien Nguyen and Dong-Seong Kim, "Accumulative-Load Aware Routing in Software-Defined Networks," 2015 IEEE
- [7] Seungbeom Song, Jaiyong Lee, Kyuho Son, Hangyong Jung and Jihoon Lee. "A Congestion Avoidance Algorithm in SDN Environment", 2016 IEEE.
- [8] Yilan Liu, Yun Pan, Muxi Yang, Wenqing Wang, Chi Fang, Ruijuan Jiang, "The Multi-Path Routing Problem in the Software Defined Network" 2015, 11th International Conference on Natural Computation (ICNC)
- [9] Cihat Cetinkaya, Erdem Karayer, Muge Sayit, Cornelius Hellge. "SDN for Segment based Flow Routing of DASH", 2014 IEEE.
- [10] Diyar Jamal Hamad, Khirata Gorgees Yalda, and Ibrahim Tanner Okumus. "Getting traffic statistics from network devices in an SDN environment using OpenFlow", 2014.