SPRING Use Cases for Software-defined Networking
draft-kim-source-routing-use-cases-01.txt

Abstract

In the Software-Defined Networking (SDN) architecture, an SDN controller establishes flow paths. An SDN controller provides global optimization of paths by controlling all managed switches. However, long flow setup time and failure recovery problems arise due to control overhead of an SDN controller. Segment routing provides source routing function by designating an explicit path. SDN and segment routing can be combined to provide both global optimization and performance enhancement. In this document, we illustrate SDN segment routing use cases of flow setup, failure recovery, Content Distribution Network (CDN), and dual homing.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Software-Defined Networking (SDN) architecture provides a network control mechanism through one or more SDN controllers. The SDN controller communicates with one or more SDN switches to calculate and/or establishes flow paths. It provides a global optimization of the managed network, but performance issues can emerge for instance with the flow setup and failure recovery processes. In order to establish new flow or change flow path, the controller should communicate with every switch along the path to send actions. The overhead can be reduced by applying source routing.

Segment routing has a source routing feature that designates an explicit path which consists of a list of segments. By applying the source routing feature of segment routing to SDN, global optimization and low control overhead can be achieved at the same time.

In this document, we will describe a combination of SDN and segment routing use cases. For fundamental communication, we will show flow setup and failure recovery use cases. We also describe use cases for some applications, such as Content Distribution Network (CDN) and dual homing.
2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following terms are defined:

Software-defined Networking:  SDN (Software-defined Networking) provides network programmability by separating control plane and data plane of network infrastructure.

Source Routing:  Source routing allows a sender of a packet to partially or completely specify the route the packet takes through the network.

Segment:  A segment that identifies an instruction

SID:  A 32-bit identification for a segment

Path Protection:  Path protection in telecommunications is an end-to-end protection scheme used in connection oriented circuits in different network architectures to protect against inevitable failures on service providers’ network that might affect the services offered to end customers. Any failure occurred at any point along the path of a circuit will cause the end nodes to move/pick the traffic to/from a new route.

3. Fast Flow Setup

Source routing offers benefits to flow setup in SDN as it minimizes communications between an SDN controller and switches. In the SDN architecture, an SDN Controller decides paths for packet delivery and sets flow entries to switches along the paths to establish the flow paths. The flow setup mechanism of the SDN architecture provides controllability to network administrators, but it takes relatively long time to establish flow paths. By applying source routing to SDN flow setup, faster flow setup can be achieved. In this section, we illustrate an example case of flow setup with source routing. Figure 1 is an example SDN topology to explain source routing use cases.
Assumptions in the above network diagram: An SDN controller is connected to the network and is able to retrieve the topology and traffic information. Based on the information and their policies, an SDN controller calculates path from a source to a destination. In an SDN architecture, Datapath ID (DPID) and port number are used to specify a path. In Figure 1, Nodes A, B, C, D, E, F, and G are switches. A number between a node and a link represents a port number.

Assume that A wants to send a packet to F, A first sends a packet to B, which is directly connected to A. When B receives the packet, it requests the SDN controller to handle the packet. The SDN controller extracts a flow from the packet by inspecting packet header, such as MAC, IP addresses, protocol, etc. Then the SDN controller calculates a path based on its own policy.

For example, path (B, E, D, F) is the shortest path and chosen by the SDN controller. The SDN Controller encodes output port numbers (2, 3) of nodes in path (E, D) and insert them into the flow entry of B. SDN Controller adds a flow entry to B. The flow entry specifies the output port 3 for the flow from A to F and includes the encoded path information to establish the path to F. After the SDN controller adds the flow entry to B, B inserts the encoded path to the packet header and forwards the packet to the port 3 as instructed in its flow entry. E receives the packet and gets its output port number 2 in the packet header. Then, E removes the port number in the encoded path it consumes. E adds the flow entry which specifies output port as 3 to its flow table and forwards the packet to the port 2. D receives the packet, adds its flow entry, and forwards the packet to the port 3. The packet is sent from A to F and the path (A, B, E, D, F) is successfully set.
Segment routing simplifies the task of controller. The controller does not have to encode all switch IDs and port number to specify paths. Assume that the operator provisioned the Node-SIDs 61, 62, 63, 64, 65, 66, and 67 respectively at nodes A, B, C, D, E, F, and G. Path (B, E, D, F) is represented as {{62, 65, 64, 66} by using SIDs. In the example above, the controller specifies the path {62, 66, 65, 67} and attaches corresponding segment routing actions.

4. Failure Recovery

Source routing offers benefits to failure recovery in SDN in terms of recovery speed. If failures occur, such as link or port failures, some flows have been delivered through them will be failed. These flows should be redirected to healthy paths as fast as possible.

A path protection mechanism prepares backup paths in case of failure and provides fast failure recovery as it provides a backup path in advance. However, if backup paths are not set, an SDN controller redirects all the affected flows sequentially, which is called restoration. In this section, we illustrate how source routing can be applied to protection and restoration for fast failure recovery in SDN.

4.1. Path Protection

In order to recover from failures fast, path protection can be implemented in SDN using the source routing mechanism. An SDN controller inserts backup path information, as well as a working path. If a failure occurs in the working path, then the affected flows is redirected to the backup path.

For example, in Figure 1, when a flow path from A to F is set, the SDN controller extracts a flow of the packet. Then, it calculates the working path (B, E, D, F) and the backup path (B, G, D, F). The working path (B, E, D, F) is established using the existing flow setup mechanism or our method shown in section 3. The SDN controller extracts output port numbers of nodes along the path. The backup path (B, G, D, F) is encoded to (2, 3) which includes output port numbers of node G and D and inserted to the flow entry of B. During the network is healthy, packets from A to F are delivered through (B, E, D, F). Assume that the link B-E fails, B detects that the port 3 is unavailable and redirects the packet to the port 4 after inserts the encoded path to the packet header. G receives the packet and obtains the output port 2 for the packet by inspecting the packet header. G adds a flow entry which instructs G to forward matched flows to the port 2. After G obtains the output port, it removes its output port number in the encoded path (2, 3), so G puts (2) to the packet header and forwards the packet to port 2. D receives the
packet, obtains the output port 3, and sends the packet to the port 3.

Similar to the flow setup use case, segment routing simplifies process of the controller. Backup (B, G, D, F) is represented to \{62, 67, 64, 66\} and controller adds the path to protect the working path.

4.2. Restoration

Unlike the path protection, a restoration mechanism does not prepare for failure cases in advance. No additional actions are required during flow setup. If failures occur, SDN nodes detect failures and notify them to a SDN controller. The SDN controller first identifies flows affected by the failures with the failure notification. Then, the controller redirects the affected flows to alternative paths.

For example, in Figure 1, let us assume that link B-E fails, then B and D detect the port 3 and 1 are unavailable, respectively. B and D send notification messages to the SDN Controller, then it extracts flows from flow tables of B and D, which are directly affected by the failure. The SDN controller then calculates alternative paths for the affected flows and sets alternative paths for the flows. When the SDN controller redirects each affected flow, source routing is used to recover failures fast. Redirection process of each affected flow is same as the flow setup shown in section 3. If 10 flows are affected by the failure, then the SDN controller should redirect these flows sequentially.

5. Content Distribution

The delivery of content in current operator’s networks is commonly implemented by means of a Content Distribution Network (CDN). The CDN consists of a number of nodes playing different roles in the overlay distribution. There are some nodes, named ingest nodes, which are placed at the root of the CDN where the content is pushed. There are some other intermediate nodes, known as delivery nodes, at regional level that can be used for traffic delivery to end users requesting the content in the region, or even for traffic forwarding to nodes deeper on the CDN or end points in other regions to serve some remote users. All these nodes maintain a cache where the content is temporary stored in order to serve subsequent request from end users.

Typically, the end user request arrive to a central control point in the CDN infrastructure which decides which end point will be in charge of serving the content, considering factors like end point load, content availability, etc. Then, the end user request is
redirected to the end point of choice, and the traffic delivery starts.

In an SDN enabled environment, the SDN controller can be in charge of instructing the most appropriate delivery node to forward traffic to the end user location by applying the segment routing logic. This could be performed in a simplistic way, just identifying the segment where the end user is located, or in a more sophisticated way, by avoiding for instance some network segments of high load or by implementing more complicated traffic engineering.

Looking at the topology in Figure 1, let us consider a delivery point deployed in A and an end user requesting some content placed in F. Let us consider also that the link E-D has a high load in this moment. Once the CDN control part resolves that the end point to deliver the traffic to the end user is the one located in A, the SDN controller can instruct the delivery node for sending the traffic to the D-F segment while avoiding the segment in E-D.

Note that in this case some communication between the control entities in the CDN and the SDN is required for collaborating in the segment routing definition.

6. Dual Homing

Typical enterprise services are built as overlay networks (in the form of VPNs). In order to offer redundancy to the connectivity of these enterprises, dual homing is implemented by connecting the router located in customer premises to two separate nodes in the operator’s network.

The decision for delivering traffic on top of one link or the other towards the other sites connected to the VPN or to Internet depends on some pre-defined policies, as load balancing, BGP parameterization tuning, etc.

In an SDN controlled scenario, the SDN controller can take the decisions of the traffic delivery and instruct the router in the customer premises to condition the traffic according to the defined segment routing injected from the controller. The policies are not more configured in the router, but in the controller. The router in the customer premises is not more required to keep any per flow state.
7. Security Considerations

One of the most important security issues of source routing in SDN is that malicious users may change paths. To prevent this, switches in a network domain should ignore encoded path information. By deleting encoded path information in packets from ports connected to external networks, we can prevent malicious users from altering paths.

8. Normative References


Authors’ Addresses

Sungsu Kim
KT
Infra R&D Lab. KT
17 Woomyeon-dong, Seocho-gu
Seoul 137-792
Korea

Phone: +82-2-526-6688
Fax: +82-2-526-5200
Email: sngsu.kim@kt.com

Jaewoo Park
KT
Infra R&D Lab. KT
17 Woomyeon-dong, Seocho-gu
Seoul 137-792
Korea

Phone: +82-2-526-6688
Fax: +82-2-526-5200
Email: jawoo.park@kt.com
EunKyoung Paik  
KT  
Infra R&D Lab. KT  
17 Woomyeon-dong, Seocho-gu  
Seoul 137-792  
Korea  

Phone: +82-2-526-5233  
Fax: +82-2-526-5200  
Email: eun.paik@kt.com  
URI: http://mmlab.snu.ac.kr/~eun/

Luis M. Contreras  
Telefonica I+D  
Ronda de la Comunicacion, s/n  
Sur-3 building, 3rd floor  
Madrid 28050  
Spain  

Email: lmcm@tid.es  
URI: http://people.tid.es/LuisM.Contreras/