Abstract

It has already been confirmed that software-defined networking (SDN) can make the networks more programmable, adaptive and application aware. However, due to the large-scale and geographically-distributed nature of wide-area networks (WAN), the scalability could become a critical issue if we incorporate SDN for WANs (i.e., realizing SD-WANs). In this paper, we design and implement a novel network system that can leverage source routing with the protocol-oblivious forwarding (POF) to facilitate efficient e-Health data transfers with low setup latency. We develop the POF-based source routing protocol to realize a pipeline based packet processing procedure, which can replace the table-lookup based approach in traditional SDN networks and make the forwarding plane more efficient. The proposed scheme is demonstrated experimentally, and the results verify that with it, the flow-tables installed in each core switches in a POF-controlled SD-WAN can be minimized and the path setup latency of traffic flows can be reduced significantly as well.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html
The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

Copyright and License Notice

Copyright (c) 2018 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1 Introduction ............................................. 3
  1.1 Terminology ..................................... 4
2 POF-Based Source Routing in SD-WANs ....................... 5
  2.1 Packet Design for source Routing ................. 6
  2.2 Procedure for Source Routing based Packet Processing ..... 7
3 Example ................................................ 10
4 Security Considerations ................................ 11
5 IANA Considerations .................................... 11
6 Conclusion ............................................. 11
7 References ............................................. 11
  7.1 Normative References ............................. 11
  7.2 Informative References ......................... 11
Authors’ Addresses ...................................... 11
1 Introduction

Nowadays, the fast development of data-intensive applications, such as e-Health, e-Science, e-commerce, etc, has brought us into the Big Data era. It is known that certain Big Data applications may generate huge volumes of data, which needs to be transferred over wide-area networks (WANs) for timely processing. For instance, in a telemedicine network, the health-monitoring devices worn by a large population of subscribers can contribute a fair amount of traffic for real-time processing. As the subscribers usually locate in a geographically dispersed manner, it would be challenging to transfer the data to the processing module(s) with low latency. Hence, both flexible architecture and efficient protocols are required to achieve flexible traffic engineering in WANs, especially when cloud-based data gathering and processing are needed. Today’s WAN architecture uses distributed traffic engineering mechanisms to reduce bandwidth congestion, but it has been proven to be inefficient due to the close coupling of the control and forwarding planes of the network. In order to support Big Data applications more efficiently, WANs also need a more programmable and adaptive architecture with effective network control and management (NC&M), which is similar to the innovation trends in other types of networks. Recently, software-defined networking (SDN) [RFC7149] has been proposed as a breakthrough technology that is promising for the next-generation Internet due to the fact that it decouples control and forwarding planes of a network and leverages centralized NC&M to make it more programmable, adaptive and application-aware. Thanks to the advances on SDN, the Big Data related traffic can be managed more efficiently in WANs. However, due to the large-scale and geographically distributed nature of WANs, the scalability could become a critical issue when incorporating SDN for WANs. Specifically, when the traffic volume increases, more and more flow-tables will be installed in the switches, which can use up their memory, make the table look-up and traffic scheduling increasingly complex, and increase the communication overhead significantly.

Unfortunately, the aforementioned issue cannot be addressed properly with the initial implementation of SDN, i.e., OpenFlow [RFC7426]. This is because OpenFlow defines protocol dependent flow-matching rules, which can lead to repeated flow-table installations and look-ups in the forwarding plane. In an OpenFlow-controlled WAN, the interactions between the switches and OpenFlow controller need to bear a relatively long communication latency, which is due to the physical distance between them and intrinsic. Hence, when setting up a flow by configuring the flow-table on each switch along the path, the hop-by-hop operation can cause a long setup delay. Note that, this is especially unwanted for the delay-sensitive e-Health data transfers that usually operate as mice flows. On the other hand, the
increasing volume of flow-tables could be another killing factor for the software-defined WANs (SD-WANs). Specifically, due to its user population, an OpenFlow-controlled SD-WAN usually needs to deploy a huge number of flows, each of which may require to install a flow-table on all the switches along the routing path. Consequently, the switches’ memory space for flow-tables can vanish quickly. Furthermore, most of the commercially-available OpenFlow-enabled switches cannot achieve a processing throughput of more than 500 FlowMods per second. In order to address the issues of OpenFlow mentioned above, recent researches have considered to enhance the programmability and flexibility of SDN forwarding plane, with the protocol-oblivious forwarding (POF) technology or the programming protocol-independent packet processors (P4). The basic ideas behind POF and P4 are similar, i.e., trying to decouple network protocols from the forwarding procedure in SDN-enabled switches and make the forwarding plane reconfigurable, programmable and future-proof. More specifically, POF develops a protocol-independent instruction set that allows to express much more flexible packet processing than the current OpenFlow specifications, while P4 mainly focuses on designing a high-level network programming language for protocol innovations. Note that both approaches have attracted noticeable interests from the open networking foundation (ONF), and are considered in its project on protocol-independent forwarding (PIF).

The memo describes a novel network system that can leverage source routing with POF to facilitate efficient e-Health data transfers with low setup latency. The memo also contains demonstration examples.

1.1 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 RFC 2119 [RFC2119].

Software-Defined Networking (SDN) - A programmable networks approach that supports the separation of control and forwarding planes via standardized interfaces.

Protocol Oblivious Forwarding (POF) - The protocol that is proposed by Huawei to provide a new way to develop SDN.

Match Fields - POF simply defines the search key of a matching field as a tuple <offset, length>. Here, offset indicates the start-location of the field in a packet, length tells the field’s length in bits.

POF-FIS - All the instructions in POF-FIS utilize the tuple <offset, length> to locate the data that they need to operate on. This provides us the freedom to manipulate any bit(s) in packets at will.
which is far more flexible than the scheme in OpenFlow. For instance, in the latest OpenFlow switch specifications, the push action is still protocol-specific and multiple actions have to be defined for each legacy protocol. However, all these push actions can be realized with one generic instruction in POF, which is add-field. Specifically, by using add-field, we can insert any field at any position in a packet. POF-FIS even includes a calculate-field instruction to provide the support on arithmetic and logical operations.

Flow Tables - The flow tables stored in a POF-enabled switch can be classified into four types, i.e., the maskedmatch (MM) table, the longest-prefix-match (LPM) table, the extract-match (EM) table, and the direct table (DT). These types of tables occupy different memory sizes and can be searched with various table lookup algorithms. Note that, a flow entry in all the tables consists of both matching field(s) and related instruction(s), except for DT, whose flow entries only include instructions. By leveraging these tables, the forwarding procedure in a POF-enabled switch can be abstracted as a data-path pipeline, and hence the network programmability and flexibility can be improved significantly.

Metadata Memory - When a switch needs to handle multiple tables in packet forwarding, it uses metadata memory to store the flow information that the current table processing generated for the next. POF-FIS defines three metadata-related instructions.

2. POF-Based Source Routing in SD-WANs

In this section, the memo explain the proposed POF-based source routing for SD-WANs, and describe both the network architecture and the forwarding procedure used by the POF enabled switches. The following figure shows the network architecture of a POF-based SD-WAN, which consist of a centralized POF controller, and core and edge switches. Note that, the edge switches work as the gateways to peer networks, while the core switches focus on packet forwarding inside the SD-WAN.
2.1 Packet Design for source Routing

Thanks to the protocol-independent nature of POF, there is no need to reuse the header fields in legacy protocols (e.g., VLAN and MPLS). Basically, the packet fields can be tailored specially to enable efficient source routing. Here, the fields are still designed to store the path information, and the following figure describes the proposed packet format for POF-based source routing. Specifically, the source routing related header fields are inserted in between Ethernet header and IP header. Moreover, after inserting the new fields, we modify the type field in Ethernet header to "0x0908" to indicate that the Ethernet frame contains a POF-based source routing packet. The detailed descriptions on the new fields are as follows.

Time-to-Live (TTL): This field occupies 8 bits and indicates the remaining hops for the packet to travel to its destination in the POF-based SD-WAN. Therefore, the value of this field will be set at the ingress edge switch, and in each subsequent switch, its value is decreased by 1. When the packet is about to leave the POF-based SDWAN, the egress edge switch remove it by applying the del-field instruction.
Port: This field occupies 32 bits, and its value identifies an output port on a POF-enabled switch. Note that, a source routing packet can contain multiple Port fields to represent the forwarding path, and all the fields form a Port stack. Each switch pops the first Port field (i.e., with <offset=120 bits, length=32 bits>) from the Port stack to find the output port for the packet. This is done by writing the value of the Port field to the metadata memory and removing the field from the packet, i.e., with the writedata-from-packet and del-field instructions, respectively.

2.2 Procedure for Source Routing based Packet Processing

The following two figures show the principle and detailed procedure of POF-based source routing, respectively.
When the first packet of a flow arrives at an ingress edge switch, it triggers a PacketIn message to be sent to the POF controller, since there is no flow entries to match against. Upon receiving the Packet-In message, the controller calculates a routing path for the flow, and sends a Flow-Mod message to the ingress edge switch, which encodes the designated output port to use on each switch along the path. The ingress edge switch stores the output ports in its metadata memory, and will insert them into every packet of the flow by using POF-FIS. The subsequent core switches use a pre-installed pipeline-like matching rule that consists of multiple flow tables to process the packets. Note that, since the forwarding path is encoded in each packet, the core switches do not need to have any interactions with the controller, and hence the setup latency is reduced significantly.

The following figure shows the proposed procedure used to process the flow tables in a core switch for source routing. We use three flow tables, including two MM tables and one DT table, to realize the overall source routing functionality as follows.

```
+--------V--------+ No
| EtherType=0x0908|---------+
+--------+--------+---------+
| Write PortId to | Other packet |
| Metadata      | processing  |
+--------+--------+---------+
| TTL=1      |---------+
+--------+--------+
| No       |---------+
| Pop port field | Pop source |
| routing header |---------+
+--------+--------+
| Send to output |---------+
| Port     |---------+
Core Switch
```
The source routing packet arrives at the switch first goes to Table 0, which includes an entry to check the type field in its Ethernet header, i.e., with <offset, length> equals <96 bits, 16 bits>. If its value equals 0x0908, we determine that it is a source routing packet and should be sent to Table 1.

Table 1 is a DT, and the entries in it only include instructions, as shown in the figure. Here, the switch executes the write-metadata-from-packet instruction to copy the value at <120 bits, 32 bits> (i.e., the Port field to encode the output port for this hop) to its metadata memory.
Table 2 includes two entries that are used to determine whether the switch is the packet’s last hop. Specifically, it checks the TTL field (i.e., <112 bits, 8 bits>). If the value of the TTL field equals 1, the switch is the last hop of this packet and it invokes the del-field instruction to delete the whole source routing related fields in the packet and restore the type field in the Ethernet header to its original value (e.g., 0x0800 for an IPv4 packet). Otherwise, the switch only removes the Port field for this hop. The output instruction is then used to forward the packet to its designated output port.

3 Example
The following example is to verify the functionality of the proposed POF-based source routing scheme. The figure shows the testbed, it consists of several software-based POF-enabled switches and a controller.

```
+---------------+
+------| POF Controller|------+
|      +-------|-------+      |
|              |              |
|              |              |
|              |              |
+-------+     +--------+     +--------+    +------+
| Host  |-----| Switch |-----| Switch |-----| Switch |----| Host |
+-------+     +--------+     +--------+     +--------+    +------+
```

An ICMP Request packet first arrives at edge switch S1, where it finds that there is no match rules configured for it. Then, S1 sends a Packet-In message to the POF controller to ask for the forwarding policy. The POF controller handles the Packet-In message, calculates a path together with the output ports on each hop along it, and instructs edge switch S1 to convert the packets to source routing ones by encoding the output port information in them.
4 Security Considerations
The mechanism described in this document does not raise any new security issues for the PCEP protocols.

5 IANA Considerations
This document includes no request to IANA.

6 Conclusion

In this memo, designed and implemented a novel network system that could leverage source routing with POF to facilitate efficient e-Health data transfers with low setup latency. We developed the POF-based source routing protocol to realize a pipeline based packet processing procedure, which could replace the table-lookup based approach in traditional SDN networks and make the forwarding plane more efficient. The proposed scheme was demonstrated experimentally, and the results verified that with it, the flow-tables installed in each core switches in a POF-controlled SD-WAN could be minimized and the path setup latency of traffic flows could be reduced significantly as well. convert the packets to source routing ones by encoding the output port information in them.

7 References

7.1 Normative References


7.2 Informative References

Authors’ Addresses

Zuqing Zhu
University of Science and Technology of China,
96 Jinzhai Rd., Hefei, Anhui, 230026, China.