Network Working Group L. Yong
Internet-Draft W. Hao
Intended status: Standard Track Huawei

Expires: January 2017 July 8, 2016

Tunnel Stitching for Network Virtualization Overlay
draft-yong-nvo3-tunnel-stitching-00

Abstract

This document describes a tunnel stitching method for delivering network virtualization overlay traffic that traverses multiple underlay networks.

Status of This Document

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 January 8, 2017.

Copyright Notice

Table of Contents

1. Introduction

Network Virtualization Overlay Traffic often traverses multiple underlay networks from a source to a destination. When using a UDP based tunnel encapsulation to deliver overlay traffic, each underlay network constructs a tunnel, then multiple tunnels chain together to form an end-to-end path that the traffic. Figure 1 shows a use case that overlay traffic traverse from DC1 to WAN, to DC2 by using VXLAN encapsulation [RFC7348]. Overlay traffic from VTEP1 to VTEP3 will be carried by Tunnel1, Tunnel2, and Tunnel3 in sequence.

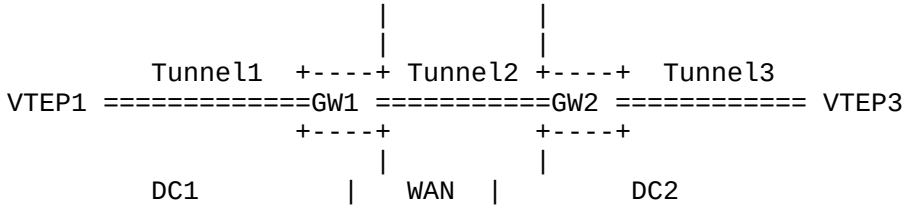

Figure 1 Tunnel Stitching Example

In this example, GW1 and GW2 is the egress for a tunnel and also the ingress for a next tunnel, where two tunnels provide a path for the traffic. Such piggybacked tunnels are referred to as tunnel stitching. A common practice for a node to perform tunnel stitching is for the node to decapsulate a received packet sent from the tunnel ingress, perform payload destination address lookup to determine the next tunnel endpoint, and then encapsulate the packet again with the next tunnel end point address and forward the packet upon.

This draft proposes a tunnel stitching method that avoids the payload lookup. The method can significantly reduce the complexity at a tunnel stitching node and shorten overlay traffic delay time from source to destination.

2. Terminology

2.1. Requirements Language

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].

3. Tunnel Stitching Technique

Assumption of this technique is that tunnel encapsulation header is able to encode a next tunnel identifier.

When a tunnel ingress node encapsulates a packet, it adds encapsulation header and outer header and place next tunnel identifier in the encapsulation header and a tunnel egress address in the outer header.

When a tunnel stitching node, i.e. a tunnel egress receives the encapsulated packet, it gets the VN ID and next tunnel identifier from the encapsulation header; and performs a local lookup to get next tunnel egress address and next-next tunnel identifier; it replace the next-next tunnel identifier in the encapsulation header and next tunnel egress address in the outer header on the packet, and forward upon. This process continues until reaching a last tunnel node, where the packet is decapsulated and forwarded upon the destination address on the packet.

This technique avoids tunnel stitching node to perform payload destination lookup that can be an extreme large table due to the address space and unable to aggregate. Comparing the payload address space, the number of next tunnels for overlay traffic is much, much smaller; thus if a tunnel stitching node assigns a next tunnel identifier to identify each next tunnel results a small table lookup at the node.

This solution only requires the first tunnel ingress node to perform original packet destination lookup to get a tunnel egress address and a next tunnel identifier.

4. Next Tunnel Identifier Field

This draft proposes having an optional field for Next Tunnel Identifier in NVO3 encapsulation protocol header and allocate one bit in the header to indicate the field present. When the bit is clear, the optional field does not present or the value in the field is invalid; when the bit is set, the optional field is present and the field contains a next tunnel identifier.

The draft proposes that a optional field for Next Tunnel Identifier contains 24 bits. 24 bits give enough space for next tunnel identifier. How a tunnel stitching node to assign a value to identify a next tunnel is outside scope of this draft.

The proposal applies to an NV03 encapsulation protocol such as VXLAN [RFC7348], GUE[GUE], VXLAN-GPE [GPE], Geneve[GNV], etc.

Note MPLS technology naturally has the capability for an egress node to assign a label that identifies a next tunnel and inform the label to an ingress node.

5. SDN Controller

When using a NVA [RFC7365] to push overlay to underlay mappings to the first tunnel ingress node, the NVA sends <NVID/NVO address, a tunnel end point address, a next tunnel identifier> mappings; the node forms a table per a VN (see Figure 2 as an example); the NVA also sends <NVID/tunnel identifier, tunnel end point address, next tunnel identifier> mappings to each tunnel stitching node. A stitching node forms a lookup table based on tunnel identifier per VN base. (See Figure 3 as an example)

MAC Address	Egress Tunnel IP Address	Next Tunnel ID
00-1B-63-84-45-E6	192.10.10.30	7654321
00-BB-78-48-45-E6	192.10.20.200	2222222
00-AA-11-34-23-D4	192.10.20.200	6666666

Figure 2 VN 100 Table at first tunnel node

Tunnel ID	Egress Tunnel IP Address	Next Tunnel ID
7654321	192.168.20.300	1234567
2222222	192.168.20.200	1111111
6666666	192.168.100.2	6666666

Figure 3 VN 100 Table at the first tunnel stitching node

Note: It is possible that two different overlay address has the same tunnel egress address but different next tunnel identifier.

6. Distributed Control Plane

When a distributed control protocol such as MP-BGP is used to distribute overlay prefix to tunnel endpoint mappings [ROSEN]; it also carries next tunnel identifier. To achieve that, a new extended community is required for the next tunnel identifier.

When a tunnel stitching node receives BGP update; it allocates the tunnel identifier and add an entry in local lookup; to redistribute the route, it inserts its IP address in remote extended community and tunnel identifier in the next tunnel ID extended community.

Each tunnel stitching node can form a local lookup table as shown in Figure 2.

7. IANA Considerations

8. Security Considerations

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC2119, March 1997.

9.2. Informative Reference

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, August 2014,.

[RFC7365] Lasserre, M., et al, "Framework for Data Center (DC) Network Virtualization", RFC7365, October 2014.

[GUE] Herbert, T., Yong, L., Zia, O., "Generic UDP Encapsulation", draft-ietf-nvo3-gue-03, work in progress.

[GPE] Kreeger, L., Elzir, U., "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-02, work in progress.

[GNV] Gross, J., Ganga, I., "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-01, work in progress.

[ROSEN] Ronsen, E., Patel, K., Van de velde, G., "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-01, work in progress.

10. Authors' Addresses

Lucy Yong
Huawei Technologies

Email: lucy.yong@huawei.com

Weiguo Hao
Huawei Technologies

Email: Haoweiguo@huawei.com