Network Working Group H. Zhou
Internet-Draft C. Li
Intended Status: Experimental eBay Inc.
Expires: September 14, 2014 March 13, 2014

Segmentation Offloading Extension for VxLAN
draft-zhou-li-vxlan-soe-00

Abstract

Segmentation offloading is nowadays common in network stack implementation and well supported by para-virtualized network device drivers for virtual machine (VM)s. This draft describes an extension to Virtual eXtensible Local Area Network (VXLAN) so that segmentation can be decoupled from physical/underlay networks and offloaded further to the remote end-point thus improving data-plane performance for VMs running on top of overlay networks.

Status of this Memo

Copyright and License Notice

Table of Contents

1 Introduction

Network virtualization over L3 transport is evolved along with server virtualization in data-centers, and data plane performance is one of the keys to the success of this combination. One of the most critical improvements in OS kernel TCP/IP stack in recent years is segmentation offloading, and now hypervisor providers support same mechanism in para-virtualized Ethernet drivers so that virtual servers can benefit from the same mechanism in virtualized world by offloading segmentation tasks to the lowest layer on hypervisors or NICs (if TSO is supported by the NICs equipped in the hypervisor).

Essentially, overlay networks has its own advantage comparing with physical underlay networks in that it does not have a hard MTU limitation. Therefore, segmentation offloading can be pushed to the remote end-point of the transport tunnel, where segmentation can be completely omitted if this remote end-point is on a hypervisor. However, this advantage is not utilized when the transport of the overlay is based on the Virtual eXtensible Local Area Network [I-D.mahalingam-dutt-dcops-vxlan], which provides a transport mechanism for logically isolated L2 overlay networks between hypervisors. Lacking segmentation information in the VXLAN header, hypervisor implementations have to make pessimistic decisions to always segment the packet in the size specified by VMs before delivering to hypervisors' IP stack, because it does not know whether the remote end-point is bridged to a physical network with hard MTU limitations. It is worth noting that the segmentation here is not the IP fragmentation in terms of the physical network MTU, which may still follow if the segment size resulting from the process above plus the tunnel outer header is bigger than the physical network MTU.

To fulfill the potential of segmentation offloading on overlay, this draft introduces segmentation metadata in VXLAN header. With the capability of carrying segmentation metadata in packets, hypervisors can offload the segmentation decision further to the remote tunnel end-point, thus decoupling the segmentation for overlay from physical limitations of underlay, providing higher flexibility to hyerpervisor implementations to achieve significant performance gains in a major part of VXLAN deployment scenarios.

Although the performance gains can be achieved is affected by the physical network MTU, there is inherently no mandatory requirement to physical layer:

1) When physical network MTU is far bigger than overlay MTU, the offloading reduces the number of packets being transmitted by TX hypervisors and received in RX hypervisors and RX VMs.

2) When physical network MTU is close to overlay MTU, the number of packets being transmitted in physical network (resulted in IP fragmentation) may not be reduced significantly, but on RX side after IP reassembling, the number of packets being delivered from the hypervisor to the receiving VM is largely reduced, thus saving the cost of hypervisor <-> VM interaction and protocol stack of the receiving VM. Furthermore, a minor cost saving is that the bytes being transmitted over physical network is slightly reduced because only one copy of headers (inner L2-L4 header, VXLAN header and outer UDP header) is transmitted for a large overlay packet.

In addition, offloading features support from NIC hardware is NOT a requirement, either, to the performance gains discussed above.

1.1 Requirements Notation

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].

1.2 Definition of Terms

GSO: Generic Segmentation Offload.

TSO: TCP Segmentation Offload.

NIC: Network Interface Card.

VM: Virtual Machine.

TX: Sending side.

RX: Receiving side.

VTEP: Virtual Tunnel End Point

2. Approach

2.1 VXLAN Header Extension

The new VXLAN Segmentation Offloading Extension (VXLAN-soe) header is defined as:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S|C|R|R|I|R|R|R|   Reserved    |         Overlay MTU             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   |                VXLAN Network Identifier (VNI) |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The changes to VXLAN are:

S Bit: Flag bit 0 is defined as the S (Segmentation Offloading Extension) bit.

S = 1 indicates that VXLAN-soe is applied to the encapsulated overlay packet, and the C Bit and Overlay MTU field (see below) are valid.

S = 0 indicates that VXLAN-soe is NOT applied, and the C Bit and Segment Size field MUST be set to 0 in accordance with VXLAN.

C Bit: Flag bit 1 is defined as the C (Checksum) bit. This bit is valid only if the S bit is set to 1.

C = 1 indicates that the checksum to the encapsulated packet is required, and SHALL be re-calculated when the segmentation is being performed.

C = 0 indicates that the checksum to the encapsulated packet is NOT needed.

Overlay MTU: bit 16 - 31 is defined as the MTU desired by TX VM for the segmentation being offloaded.

Its value indicates the max size of an overlay segment including its L3 header, but NOT including Ethernet header. This field is valid only if the S bit is set.

2.2 TX VTEP

VTEP at TX side MUST set the S bit to 1 if the packet to be encapsulated is NOT segmented and it decides to offload the segmentation to the remote end-point. In such case the C bit and Overlay MTU field MUST be set accordingly. This is the typical use case when the TX VTEP is a hypervisor transmitting TCP stream of VMs with large sliding windows.

VTEP at TX side MUST clear the S bit if the packet to be encapsulated is segmented already or does NOT need to be segmented in terms of the overlay MTU. In such case, the encapsulation is in the same format as specified in VXLAN. This is the typical use case when the TX VTEP is a hypervisor transmitting small size overlay packets, or a gateway forwarding overlay packets without

offloading requirements.

2.3 RX VTEP - Hypervisors

When a VTEP at RX side is on a hypervisor, checking of the S bit is OPTIONAL.

2.4 RX VTEP - Gateways

When a VTEP at RX side is on a gateway node that connects overlay networks and physical networks, the S bit MUST be checked and the VTEP MUST ensure the segmentation specified by the header fields is performed by the VTEP itself or offloaded further - it MAY offload the segmentation again to the subsequent transmission mechanisms: such as GSO and TSO, or, if the link to the next hop is also an overlay based on VXLAN-soe (or other tunneling protocols that supports segmentation offloading), pass the segmentation metadata to the next hop.

3 Interoperability

In addition to offload segmentation requests from VMs, VXLAN-soe enabled VTEP is able to offload segmentation requests from STT [I-D.davie-stt] overlay, because the metadata required in VXLAN-soe header is a subset of STT metadata. The additional segmentation offloading information carried in STT metadata such as L4 offset can be obtained by examine inner headers of the packets.

VXLAN-soe defines Overlay MTU at the same position of Protocol Type field in VXLAN-gpe [I-D.quinn-vxlan-gpe], another extension of VXLAN. This is not a problem because VXLAN-soe is introduced for segmentation offloading use cases where Ethernet header is always encapsulated, and it uses different flag bits to be distinguished from VXLAN-gpe.

4 Security Considerations

There is no special security issues introduced by this extension to VXLAN.

5 IANA Considerations

This document creates no new requirements on IANA namespaces [RFC5226].

6 References

6.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2 Informative References

[I-D.mahalingam-dutt-dcops-vxlan]
Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-08 (work in progress), February 2014.

[I-D.davie-stt]
Davie, B. and J. Gross, "A Stateless Transport Tunneling Protocol for Network Virtualization (STT)", draft-davie-stt-05(work in progress), March 2014.

[I-D.quinn-vxlan-gpe]
Agarwal, P., Fernando, R., Kreeger, L., Lewis, D., Maino, F., Quinn, P., Yong, L., Xu, X., Smith, M., Yadav, N., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-quinn-vxlan-gpe-02 (work in progress), December 2013.

Authors' Addresses

Han Zhou
eBay, Inc.

EMail: hzhou8@ebay.com

Chengyuan Li
eBay, Inc.

Email: chengyli@ebay.com