Design and Implementation of Hardware Accelerated VTEP in datacenter networks

Chang-Gyu LIM*, Soo-Myung PAHK*, Tae-Il KIM*, Jong-Hyun LEE*

*ETRI (Electronics and Telecommunications Research Institute), Daejeon, Korea

{human, smpahk, tikim, jlee}@etri.re.kr

***Abstract*— VXLAN (Virtual eXtensible Local Area Network) is an edge-overlay model that uses L2-in-L3 tunneling protocol. It has attracted attentions for multi-tenant datacenter networks. For the deployment of VXLAN in legacy networks, networks can include VXLAN gateways which forward traffic between VXLAN and non-VXLAN environments. This paper proposes the design of VXLAN gateways which are not in servers, but in physical devices. Additionally, we show a hardware accelerated VTEP (VXLAN tunnel end point) that can connect virtual machines to VXLAN segments without software VTEPs, such as OVS (Open vSwitch) VTEPs. The performance result of the hardware accelerated VTEPs is more efficient than software VTEPs' with regard to CPU consumption and traffic throughput of servers.**

***Keywords*— VXLAN, VTEP, datacenter**

I. INTRODUCTION

In these datacenters, resources are dynamically allocated to a tenant based on changing application needs. The virtual L2 network [1] is agnostic to the physical topology of the datacenter network and isolates the tenant's resources from issues concerning physical reachability.

VXLAN (Virtual eXtensible Local Area Network) is an edge-overlay model that uses L2-in-L3 tunneling protocol. It has attracted attentions for multi-tenant datacenter networks. VXLAN is based on pre-standard IETF draft [2].

For the deployment of VXLAN in legacy networks, networks can include VXLAN gateways which forward traffic between VXLAN and non-VXLAN environments. Figure 1 shows an example of using VXLAN in a datacenter network with a TOR switch and a VXLAN gateway (in a physical switch) connecting servers and a non-VXLAN legacy network via an L3 core network.

In Section 2, we propose the design of VXLAN gateways which are not in servers, but in physical devices. Additionally, we show a hardware accelerated VTEP (VXLAN tunnel end point) that can connect VM (Virtual Machine)s to VXLAN segments without software VTEPs, such as OVS (Open vSwitch) [3] VTEPs.

Section 3 shows implementation of the proposed hardware accelerated VTEP.

Section 4 presents experimental results of the hardware accelerated VTEPs and software VTPEPs.

II. REQUIREMENTS AND DESIGN

In this section, we describe some requirements of VXLAN and show the proposed features of the hardware accelerated VTEP.

A. Requirements

VXLAN is a MAC in UDP encapsulation scheme. The lists below are the requirements of the proposed hardware accelerated VTEP.

- VXLAN segments identifying an overlay network by VNI (VXLAN Network Identifier)
- VTEP functionality
- VXLAN gateways
- VXLAN flooding over unicast/multicast tunnels
- Hardware Accelerated VTEP that enables to assign a VM to a VNI

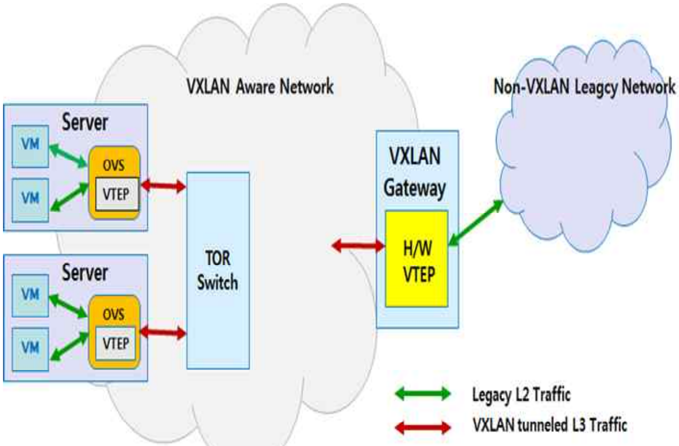

Figure 1. An example of using VXLAN in a datacenter network

B. VXLAN Segments and Packet Format

VXLAN is an L2 overlay scheme over an L3 core network. Each overlay is called a VXLAN segment. VMs can communicate with each other in a same VXLAN segment. Each VXLAN segment is scoped through a 24 bit segment ID, called VNI. Figure 2 shows a VXLAN packet format. The original L2 packet is bridged to and from a virtual L2 network at a VTEP. VXLAN header includes VNI. UDP destination port is a well-known port to identify VXLAN header. UDP source port is a hash of the inner Ethernet packet's header to obtain a level of entropy for load balancing of the VM to VM traffic across the VXLAN overlay.

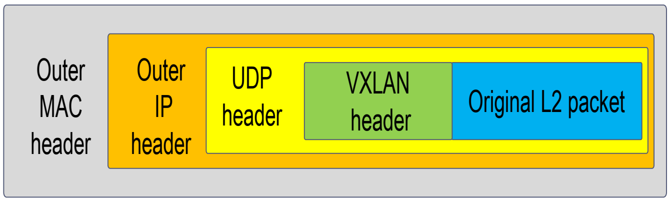

Figure 2. A VXLAN packet format

C. VTEP functionality

The VNI and VXLAN related tunnel/outer header encapsulations are done in VTEP. Thus, VMs never see it.

When packets are transmitting, the original L2 packets are encapsulated in a VXLAN header. The resulting packets are sequentially wrapped in UDPs. For forwarding packets through an L3 core network, the outer destination IP which is the remote VTEP's destination IP and the outer destination MAC which is the next hop destination MAC are also added to the packets.

When the packets are forwarding in an L3 core network, the outer destination MAC is changed as per normal routing rules.

When the packets are receiving in the destination VTEP, if the outer destination IP of receiving packets and the current VTEP's IP are same, the VXLAN tunnel is terminated and VXLAN header is parsed. VNI is obtained from the VXLAN header. The original L2 packets are forwarded to VMs that are connected to the VXLAN segment which is identified by VNI.

D. VXLAN Gateways

VTEPs are typically deployed within hypervisors that are virtualization aware. However, some legacy network nodes are not virtualization aware. Therefore, we need to implement a VTEP in the physical switch itself. This is a VXLAN gateway.

A VXLAN gateway can forward packets between VXLAN and non-VXLAN environments.

E. VXLAN flooding over unicast/multicast tunnels

VLXAN uses flooding and dynamic MAC running to discover remote MAC addresses (VM's MAC addresses) and MAC-VTEP mappings (remote VTEPs and VM's MAC addresses mappings).

We provide VXLAN flooding and dynamic MAC running by using unicast tunnels. After creating a VXLAN segment (VNI), we create unicast tunnels for each remote VTEP. Unicast tunnels enable to make a VXLAN segment without multicast protocols in an L3 core network. It is useful if an L3 core network does not support any multicast protocol.

We also provide VXLAN flooding and dynamic MAC running optionally by multicast tunnels.

F. Hardware Accelerated VTEP

We already mention that we provide a VXLAN gateway function. However, a VXLAN gateway cannot provide a VP (Virtual Port) per VM to a non-VXLAN network. This is because there is no way to map packets which come in the same port together from VMs in the legacy network to the correct VNI. A VXLAN gateway can only provide a VP per AC (Attached Circuit) to a non-VXLAN network.

To support a VP per VM in a non-VXLAN network, we propose a hardware accelerated VTEP that can connect VMs to VXLAN segments without software VTEPs, such as OVS VTEPs. Software VTEPs, in a hypervisor, can assign each VM's MAC address to a VNI. On the contrary, a general VXLAN gateway cannot handle it.

The proposed hardware accelerated VTEP enables to assign a VM to the correct VNI by two different ways.

The one is that a VM MAC address is assigned to a VNI automatically by EVB (Edge Virtual Bridging)-VDP (Virtual Station Interface Discovery and Configuration Protocol) [4]. VDP protocol sends a VM MAC address to a hardware VTEP (in a physical switch). When the hardware VTEP received a VDP message which is a VXLAN service creation request from a VM in a server, the hardware VTEP finds a request VNI and connects VM's MAC address to the VNI with creating a VM-VP (Virtual Machine-Virtual Port). Therefore, VM can be mapped to a correct VXLAN segment even if packets come in the same port together.

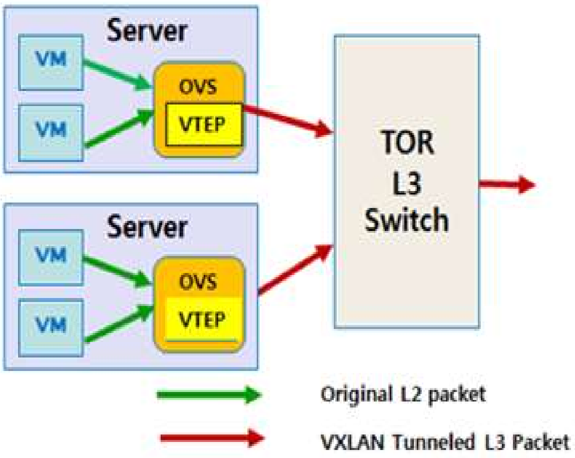

Figure 3. An example of a packet flow on software VTEPs

Figure 3 shows an example of a packet flow on software VTEPs, OVS VTEPs.

Figure 4 shows an example of a packet flow on proposed hardware VTEPs.

The other is that a VM MAC address is assigned to a VNI by an operator. We can type a VM MAC address which needs to connect a VNI to hardware VTEPs directly. This method does not depend on protocols. Thus, it is easier and cheaper than the former method.

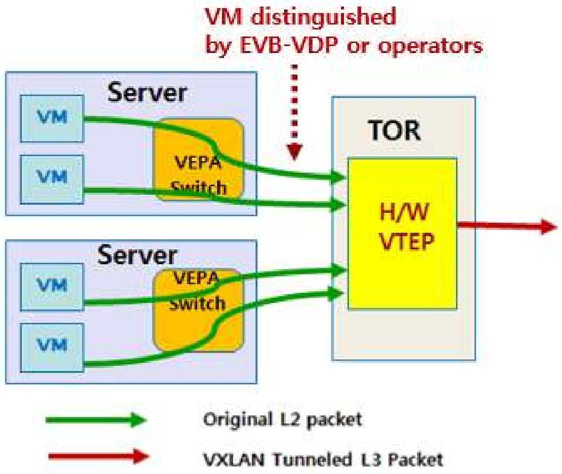

Figure 4. An example of a packet flow on hardware accelerated VTEPs

III. Implementation

We implement the proposed hardware accelerated VTEP using a ZebOS platform.

Figure 5 shows the block diagram of the proposed hardware accelerated VTEP. A VXLAN management module is included in NSM (Network Services Manager) which is the main block of a ZebOS platform. NSM also include the EVB/VXLAN adaptation module that adapts messages between NSM and EVB, NSM and IMI (Integrated Management Interface). CLI (Command Line Interface) commands come to NSM through IMI. There are QoS (Quality of Service)/ACL (Access Control List) databases that have QoS/ACL profiles deploying to VPs that assign to VMs or ACs.

The VXLAN management module provides the lists below.

- VXLAN system configuration
- Bridge mode management
- VNI configuration
- VM-VP creation/deletion
- AC-VP creation/deletion
- Unicast tunnel creation/deletion
- Multicast tunnel creation/deletion
- VNI MAC table management
- QoS/ACL profiles management
- EVB message callback function management
- VXLAN message callback function management

We install this platform to a target system that uses a Broadcom Trident2 chipset and provides 48 ports which support 1Gbps or 10Gbps port speeds. We control the management of a target system by using message communications between HAL (Hardware Abstraction Layer) and HSL (Hardware Service Layer). HSL manages Broadcom SDK (Software Development Kit).

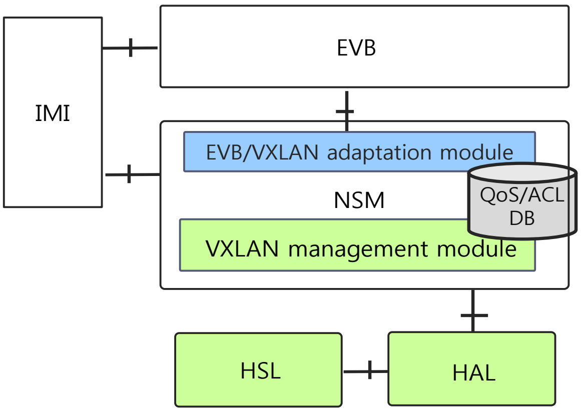

Figure 5. The block diagram of the hardware accelerated VTEP

IV. Result

We compare the proposed hardware accelerated VTEP and a software VTEP, an OVS VTEP.

Figure 6 shows a software VTEP performance measurement environment. Each VTEP has 6 VMs. Two VTEPs are connected to a 10Gbps Ethernet link. In a physical server, OVS-VXLAN is applied for a traffic performance measurement. We need a network performance testing program to generate TCP traffic. We choose a testing program, Netperf [5] that can measure throughput and generate traffic of the variable message size.

On the contrary, figure 7 shows the proposed hardware accelerated VTEP performance measurement environment. Each VTEP has 6 VMs equally. But two VTEPs are connected to a 10Gbps Ethernet link through two physical switches that contain the proposed hardware accelerated VTEP functionality.

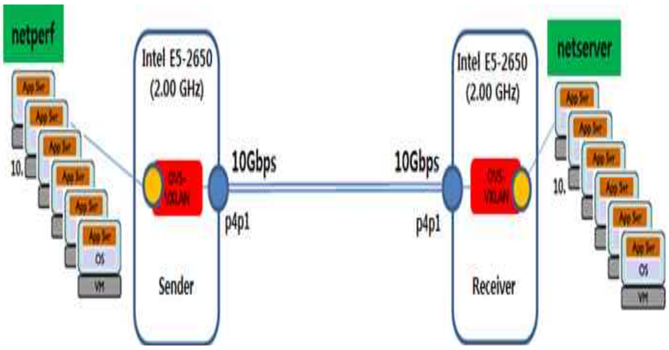

Figure 6. A software VTEP performance measurement environment

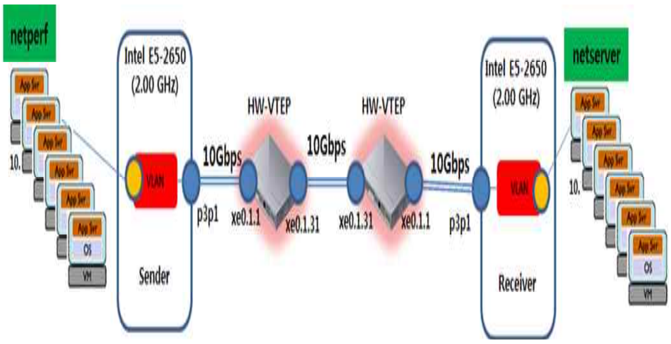

Figure 7. The proposed hardware accelerated VTEP performance measurement environment

The result of TCP throughput comparison between an OVS VTEP and the proposed hardware accelerated VTEP is shown in figure 8.

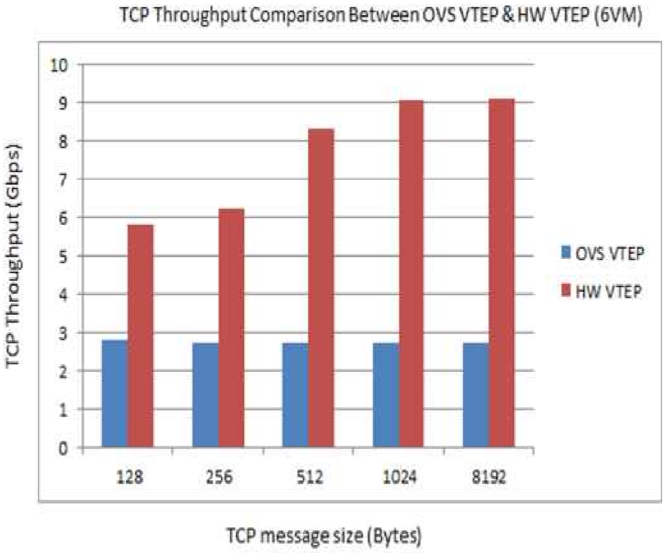

Figure 8. The result of TCP throughput comparison

The throughput of the hardware accelerated VTEP is 2 times (128 bytes TCP message size) or 3 times (1024 bytes TCP message size) more than the OVS VTEP's.

The result of host CPU load comparison between an OVS VTEP and the proposed hardware accelerated VTEP is shown in figure 9.

The host CPU load per 1Gpbs TCP stream transmission of the hardware accelerated VTEP is 2 times (1024 bytes TCP message size) less than the OVS VTEP's.

The proposed VTEP's throughput is depended on I/O port speed limitations in physical servers and the proposed VTEP reduce server's CPU loads.

We suppose that if server's I/O port speed is increased, traffic throughput will be increased by using unused resources of server's CPU. Thus, we can use resources more efficient in datacenter networks by using the proposed hardware accelerated VTEP.

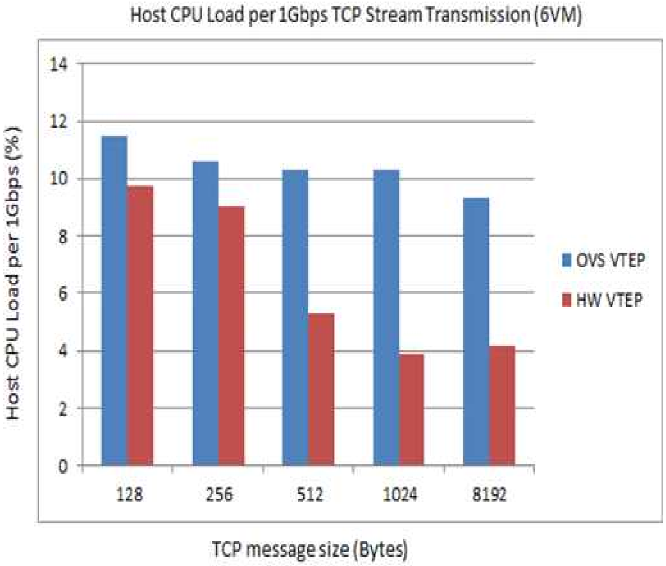

Figure 9. The result of host CPU load per 1Gbps TCP stream trasmission

V. Conclusions

This paper proposes the design of VXLAN gateways which are not in servers, but in physical devices. Additionally, we show a hardware accelerated VTEP that can connect virtual machines to VXLAN segments without software VTEPs, such as OVS VTEPs. The performance result of the hardware accelerated VTEPs is more efficient than a software VTEP with regard to CPU consumption and traffic throughput of servers. The throughput of the hardware accelerated VTEP is 2 times (128 bytes TCP message size) or 3 times (1024 bytes TCP message size) more than the OVS VTEP's. The host CPU load per 1Gpbs TCP stream transmission of the hardware accelerated VTEP is 2 times (1024 bytes TCP message size) less than the OVS VTEP's. We suppose that if server's I/O port speed is increased, traffic throughput will be increased by using unused resources of server's CPU. Thus, we can use resources more efficient in datacenter networks by using the proposed hardware accelerated VTEP.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0101-15-233, Smart Networking Core Technology Development).

References

[1] D. Cai and S. Natarajan, "The Evolution of the Carrier Cloud Networking", IEEE 7th International Symposium on Service-Oriented System Engineering, Mar. 2013.

[2] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, C. Wright, *draft-mahalingam-dutt-dcops-vxlan-03*, IETF, Feb. 2013.

[3] http://www.openvswitch.org

[4] *IEEE Std 802.1Qbg-2012*, IEEE, May. 2012.

[5] http://www.netperf.org