PICA8®
WHITE BOX SDN

WHITEPAPER

Improving Overlay Solutions with Hardware-Based VXLAN Termination

Connections Between the Virtual and Physical World

Abstract

As virtualization and cloud technologies become more prevalent in today's network infrastructure, the legacy network lacks agility and scale. This paper describes how a hardware-accelerated VXLAN (Virtual Extensible LAN) solution using Pica8 software on bare metal switches can solve those issues without performance impact. We will also explain how the growing acceptance of a standard API to control this technology (OVSdb) is helping the deployment of these solutions.

Introduction

Network architectures are undergoing fundamental changes as cloud providers and large data centers address ever-growing mobile workloads. When workloads shift across physical and virtual infrastructure like servers, storage, authentication and load-balancing devices, new networks must reconfigure themselves on the fly based on ever-changing application needs. To meet this requirement, network infrastructures will continue to evolve from being hardware-bound to being software-led. It will be the software that enables the dynamic and flexible control of the network, while the hardware provides performance support.

Operations teams typically build a software layer that defines a typical workflow to initiate a service. But today, hardware vendors control that software and do not share an open interface with the developer community to use its network infrastructure for service differentiation (contrasting to what is possible on the server side).

In addition, traditional network architectures have challenges with logical scale and provisioning. VXLAN tries to solve the Layer-2 scale issue and VLAN limitation on the network. To scale the provisioning of those tunnels, it is common to use the standard OVSdb API used by all the major overlay SDN controllers today (VMware NSX, Midokura MidoNet, OpenContrail).

This white paper examines alternative software-led approaches to controlling the network and specifically looks at the benefits of building VXLAN tunnels as a virtual network overlay.

VXLAN Explained

What Problems VXLAN Are Solving?

Problem 1: Maximum number of VLANs in a data center

VXLAN is a standards-based Layer-2 and overlay technology, defined in RFC 7348. VXLAN provides the same Ethernet Layer-2 network services that a VLAN does, but with greater scalability, extensibility and flexibility. It provides multi-tenancy across the data centers by extending Layer-2 segments over Layer-3 boundaries. With VXLAN, up to 16 million Layer-2 segments are possible in contrast to only 4,000 by VLANs. Therefore, VXLAN is suitable for large-scale deployments. It is also used as an overlay solution to extend Layer-2 connectivity over Layer-3 segments.

Problem 2: Using a routing technology in a data center and keeping Layer-2 connectivity for some applications

Using typical Layer-2 technologies for data centers is not a scalable solution (STP or MLAG). It is becoming increasingly common to use routing protocols (OSPF and BGP) in the data center (see draft-ietf-rtgwg-bgp-routing-large-dc-02 for an example of large data center design). But even for those Layer-3 data centers, it is useful to keep Layer-2 connectivity between hosts or virtual machines. This can be done with a VXLAN tunnel doing Layer-2 frame encapsulation across the Layer-3 data center.

A typical example of such need is VM mobility. Moving VMs between or inside the data center usually needs a Layer-2 segment between the two physical hosts. Creating combined segments over VTEPs and VXLAN enables seamless VM mobility as the combined solution provides virtual Layer-2 across physical boundaries.

VXLAN Defined

VXLAN is an overlay technology. By using UDP for transporting Layer-2 (MAC) frames, it is an MAC-in-UDP encapsulation method. In VXLAN, the original Layer-2 frame is encapsulated inside an IP-UDP packet by adding
VXLAN header as shown in Figure 1.

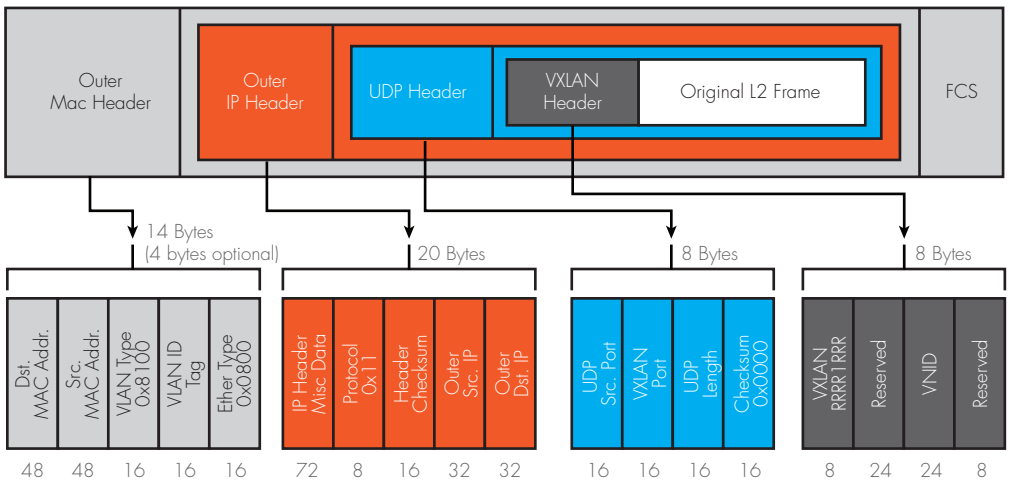


Figure 1. VXLAN Packet Format

The VXLAN header is 8-bytes in length and consists of 24-bit VXLAN network identifier (VNID) and hence, provides up to 16M Layer-2 segments.

VXLAN Tunnel Endpoint (VTEP)

VXLAN uses VTEPs to map end devices or tenants to VXLAN and perform encapsulation and de-encapsulation functions. A VTEP consists of two interfaces: one for local LAN and a second one for an IP interface to connect to other VTEPs across an IP network. The IP interface identifies the VTEP device by unique IP address assigned to this interface. VTEPs use the IP interface to encapsulate Layer-2 frames, and transports the resulting encapsulated packet over the IP network. Additionally, a VTEP can discover remote VTEPs for relevant VXLAN segments, and learns remote MAC address-to-VTEP binding via IP interface. The functional components of VTEPs and corresponding logical topology are shown in Figure 2.

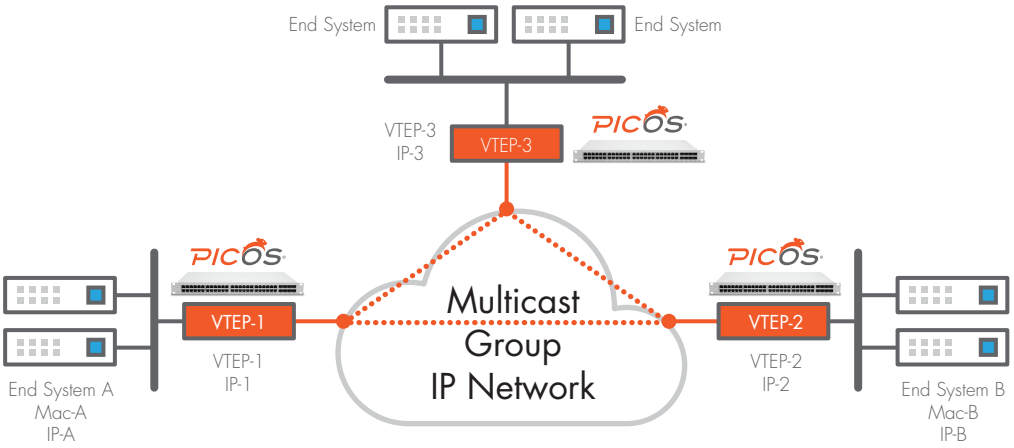


Figure 2. VTEP

VXLAN Tunnel Provisioning

VTEP Termination Provisioning

For a specific VXLAN tunnel, only the two VTEP termination points need to be provisioned. As for the rest of the network, the VXLAN packet is handled as a standard IP packet. A VTEP can be either a hardware VTEP (typically a TOR switch) or a software VTEP (typically a virtual switch on a server like OVS). It is also possible to have a VXLAN tunnel between a hardware VTEP and a software VTEP. The provisioning can be done manually, via CLI like any other networking feature, but it obviously does not scale for a dynamic data center. The standard API for VTEP provisioning is the OVSDB VTEP API.

What is OVSDB VTEP API?

In the past, networking standards were designed by standard bodies like IETF or IEEE. As the networking industry becomes more integrated with the server and software technologies, the open source community will become more and more the source of network standard. OVSDB is a good example of an open source software driven standard.

The Open vSwitch Database Management Protocol (OVSDB) is an OpenFlow configuration protocol designed to manage Open vSwitch implementations. Because OVS has become widely popular, its management protocol has become the de-facto standard API to control VTEP termination. OVSDB is a JSON RPC protocol.

What is an VXLAN Overlay Solution?

A VXLAN overlay solution is a mesh of tunnels built automatically by an SDN controller. Typical examples of this include Midokura MidoNet, VMware NSX or OpenContrail. Those SDN controllers connect all the data center objects like VMs, storage, load balancing or firewalling using common infrastructure while providing secure, seamless connectivity in a multi-tenant environment. This is done by creating an overlay of full-mesh, stateless tunnels between the tenant objects. Those SDN controllers address the connectivity of the virtual switches embedded in VM host operating systems.

Network overlay solutions use a policy-based method to assign network parameters to VMs, rather than use the data plane to learn them as done by legacy equipment. This approach creates an “any-to-any” connectivity between VMs. The virtual switch is

a critical piece of this infrastructure because that is where the tunneling encapsulation and de-capsulation starts. Open vSwitch (OVS) is a good example as it assumes a fully virtualized environment. Deployments are typically a hybrid environment with some non-virtualized servers or service appliances.

To connect a VM to a non-virtualized device, a network gateway is needed as a termination point for the overlay tunnel. A hardware-based VxLAN tunnel solution is preferred because a software-based network gateway can quickly become a performance chokepoint as it has to handle de-capsulation of the tunneling protocol at line-rate speed.

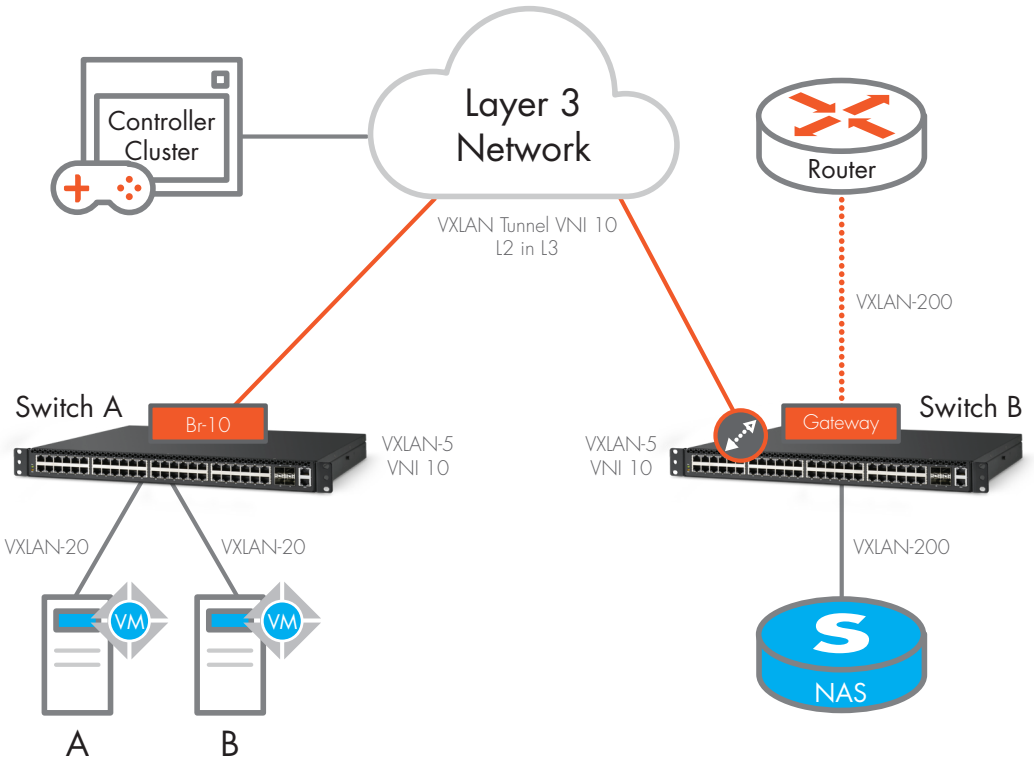
Benefits of Hardware-accelerated VXLAN

Virtual to Physical Bridge

Overlay solutions provide the creation of Layer-2 segments over Layer-3 underlays, via a native VXLAN bridge. However, it can only create these tunnels between virtual switches. Any connectivity to physical bare metal machines or network devices is not covered and requires a gateway.

In every deployment, connectivity beyond the virtual cluster is a must. While connecting many virtual machines is the core need, most data centers require connectivity to physical NAS for storage, external firewalls, routers and cloud management systems. Creating combined solutions using a PicOS-integrated gateway enables a seamless combined solution where the VM is unaware of the translation and seamlessly connects to any physical server and gateways.

For a description of the PicOS architecture, please visit:
https://www.pica8.com/products/



Software VXLAN Encapsulation Performance Impact

VXLAN uses MAC in UDP encapsulation. Since VXLAN is an additional encapsulation mechanism introduced at the hypervisor layer, there are certain performance implications.

VXLAN introduces an additional layer of packet processing at the hypervisor level. For each packet on the
VXLAN network, the hypervisor needs to add protocol headers on the sender side (encapsulation) and remove
them (de-encapsulation) on the receiver side. This causes additional CPU overhead, the amount of which will vary based upon the packet size.

Apart from this CPU overhead, some of the offload capabilities of the NIC cannot be used because the inner packet is no longer accessible. The physical NIC hardware offload capabilities (e.g., checksum offloading, TCP segmentation offload) have been designed for standard (non-encapsulated) packet headers, and some of these capabilities cannot be used for encapsulated packets.

In such a case, a VXLAN-enabled packet will require CPU resources to perform a task that otherwise would have been done more efficiently by physical NIC hardware. There are certain NIC offload capabilities that can be used with VXLAN, but they depend on the physical NIC and the driver being used. As a result, the performance may vary based on the hardware used when VXLAN is configured. Moving the VXLAN VTEP tunnels to ToR switches solve all those performance issues and thus is often a requirement for high performance applications.

Single Management

VLAN configuration has always been a challenge, especially as the configuration file sizes and complexity grow exponentially. Adding more protocols only exacerbates the problem. Managing the mapping between different technologies creates operational nightmares and makes troubleshooting nearly impossible. A single management location for the configuration files can greatly simplify the process of connecting the physical and virtual worlds using VXLAN tunnels.

Documentation

For complete configuration details for PicOS, please visit our support site:
http://www.pica8.com/support/documents/

Pica8 Software Inc.
Corporate Headquarters

1032 Elwell Court, Suite 105
Palo Alto, California 94303 USA
650-614-5838 | www.pica8.com
© Pica8 Software Inc, 2015. All rights reserved.Produced in the United States 05/15.

Pica8 and PicOS are trademarks of Pica8 Software Inc.

Pica8 and PicOS trademarks are intended and authorized for use only in countries and jurisdictions in which Pica8 Software Inc. has obtained the rights to use, market and advertise the brand. Pica8 Software Inc. shall not be liable to third parties for unauthorized use of this document or unauthorized use of its trademarks. References in this publication to Pica8 Software Inc. products or services do not imply that Pica8 Software Inc. intends to make these available in all countries in which it operates. Contact Pica8 Software Inc. for additional information.