Tunnels Configuration Guide, Addendum

February 2019

Supporting AT&T Vyatta Network Operating System

Copyright Statement

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

About This Guide

This addendum describes tunnels functionality that was updated on the AT&T Vyatta vRouter (referred to as a virtual router, vRouter, or router in the guide).

Features	vRouter Release	Rev of Tunnels Configuration Guide 5600 Addendum	Date
VXLAN	1801	1	Jan 2019
VXLAN	1801	2	Feb 2019

Contents

VXLAN Tunnel Overview

Virtual Extensible LAN (VXLAN) tunnels support running an overlay Layer 2 network over an existing IPv4 or IPv6 transport network. It was originally developed to increase scalability in data center and cloud computing environments by increasing the number of isolated L2 networks from 4096 (limited by a 12-bit VLAN ID) to 16 million (using a 24-bit VXLAN ID).

Traffic belonging to the overlay network is encapsulated in a Layer 3 UDP packet that is routed over the underlying transport network.

The entity that performs the encapsulation and de-encapsulation is called a VXLAN tunnel endpoint (VTEP).

A VXLAN network identifier (VNI) uniquely identifies each Layer 2 subnet or segment. Virtual machines on the same VNI can communicate directly with each other, whereas virtual machines on different VNIs need a router to communicate with each other.

Benefits of VXLAN Tunnels

These are some benefits of VXLAN tunnels:

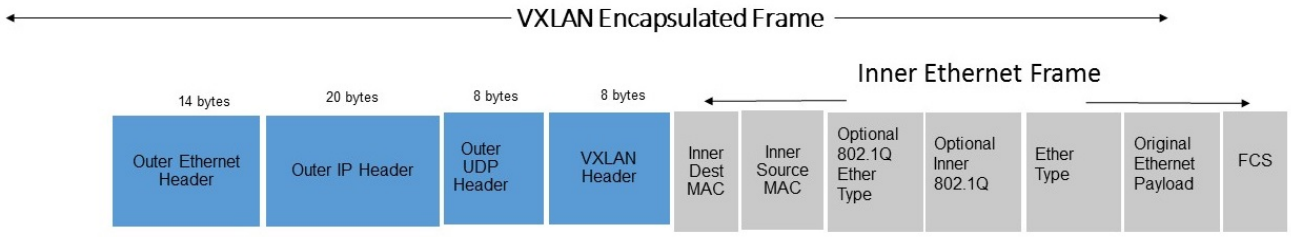
- You can theoretically create as many as 16 million VXLANs
- You can enable migration of virtual machines between servers that exist in separate Layer 2 domains by tunneling the traffic over Layer 3 networks. This functionality allows you to dynamically allocate resources within or between data centers without being constrained by Layer 2 boundaries or being forced to create large or geographically stretched Layer 2 domains.
- Using VXLANs to create smaller Layer 2 domains that are connected over a Layer 3 network means that you do not need to use Spanning Tree Protocol (STP) to converge the topology but can use more robust routing protocols in the Layer 3 network instead. In the absence of STP, none of your links are blocked, which means you can get full value from all the ports.
- Using routing protocols to connect Layer 2 domains allows you to load-balance the traffic to allow you to make the best use of your available bandwidth.

VXLAN as an Overlay Network

VXLAN is often described as an overlay technology because it allows you to stretch Layer 2 connections over an intervening Layer 3 network by encapsulating (tunneling) Ethernet frames in a VXLAN packet that includes IP addresses. Devices that support VXLANs are called *virtual tunnel endpoints (VTEPs)*—they can be end hosts or network switches or routers. VTEPs encapsulate VXLAN traffic and de-encapsulate that traffic when it leaves the VXLAN tunnel. To encapsulate an Ethernet frame, VTEPs add a number of fields, including the following fields:

- Outer media access control (MAC) destination address (MAC address of the tunnel endpoint VTEP)
- Outer MAC source address (MAC address of the tunnel source VTEP)
- Outer IP destination address (IP address of the tunnel endpoint VTEP)
- Outer IP source address (IP address of the tunnel source VTEP)
- Outer UDP header

- A VXLAN header that includes a 24-bit field—called the *VXLAN network identifier (VNI)*—that is used to uniquely identify the VXLAN. The VNI is similar to a VLAN ID, but having 24 bits allows you to create many more VXLANs than VLANs. The VXLAN header also contains 8 bits for VXLAN flags and reserved fields.



Note

Because VXLAN adds 50 to 54 bytes of additional header information to the original Ethernet frame, you might want to increase the MTU of the underlying network. In this case, configure the MTU of the physical interfaces that participate in the VXLAN network, not the MTU of the logical VTEP source interface, which is ignored.

Vyatta vRouter support for VXLAN Tunnels

Vyatta vRouter has added support for both VXLAN and VXLAN Generic Protocol Encapsulation (VXLAN-GPE) tunnels.

The VXLAN configuration must be specified under the existing `tunnel` configuration sub-tree. All existing sub-commands that are relevant to VXLAN tunnels are supported. A new set of sub-commands is added for attributes that are specific to VXLAN tunnels.

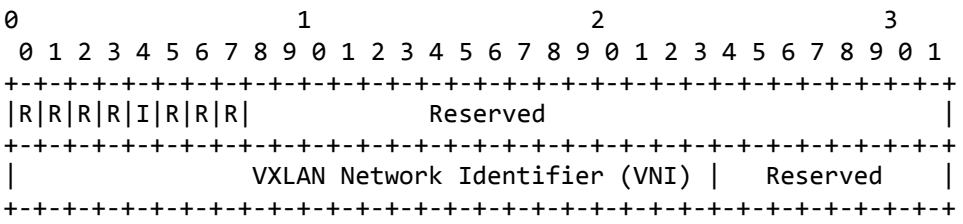
VXLAN compared to VXLAN-GPE

A standard VXLAN frame encapsulates Ethernet frames in an outer UDP/IP transport. The payload within a VXLAN packet is required to be a full Ethernet frame.

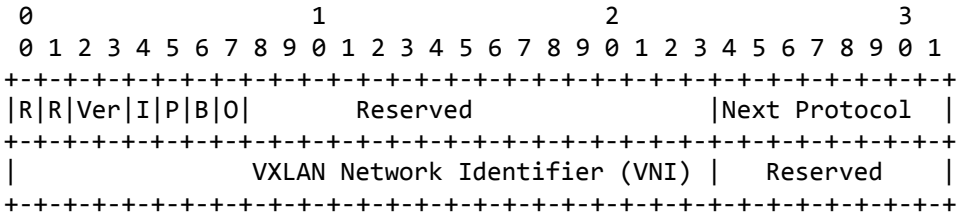
VXLAN-GPE frame adds a GPE header to extend the existing VXLAN protocol to provide support for multiprotocol encapsulation, operations, administration and management (OAM) signaling and explicit versioning. VXLAN-GPE is a backwards-compatible extension to VXLAN to allow overlay traffic of several types (including but not limited to Ethernet) to be transported over the underlying IP network.

Comparing a Standard VXLAN Header with a GPE VXLAN Header

Standard VXLAN Header



VXLAN GPE Header



Fragmentation Considerations for VXLAN GPE

VTEPs must never fragment an encapsulated VXLAN GPE packet, and when the outer IP header is IPv4, VTEPs must set the DF bit in the outer IPv4 header. The underlay network should be configured to carry an MTU large enough to accommodate the added encapsulation headers. It is recommended that VTEPs perform Path MTU discovery to determine if the underlay network can carry the encapsulated payload packet.

Connecting a VXLAN VTEP to a VXLAN GPE VTEP

To connect a VXLAN VTEP to a VXLAN GPE VTEP, use the following guidelines.

- A VXLAN VTEP conforms to VXLAN frame format and uses UDP destination port 4789 when sending traffic to a VXLAN GPE VTEP.
- As per VXLAN, reserved bits 5 and 7, VXLAN GPE P and O-bits respectively must be set to zero.
- The remaining reserved bits must be zero, including the VXLAN GPE version field, bits 2 and 3.
- The encapsulated payload *must* be Ethernet.

Connecting a VXLAN GPE VTEP to a VXLAN VTEP

To connect a VXLAN-GPE VTEP to a VXLAN-VTEP, use the following guidelines.

- Do not encapsulate non-Ethernet frames to a VXLAN VTEP.
- Conform to VXLAN frame format: set the P bit to 0, the Next Protocol to 0 and use UDP destination port 4789.
- A VXLAN GPE VTEP MUST also set O = 0 and Ver = 0 when encapsulating Ethernet frames to VXLAN VTEP.
- The receiving VXLAN VTEP will treat the packet as a VXLAN packet.

NOTE: With VXLAN GPE, issues such as spoofing, flooding, and traffic redirection are dependent on the particular protocol payload encapsulated.

Static VXLAN Tunnels

Static VXLAN tunnels connect two VTEPs in a given environment. Static VXLAN tunnels are the simplest deployment mechanism for small scale environments and are interoperable with other vendors that adhere to VXLAN standards. You must specify which VTEPs are in a particular VNI.

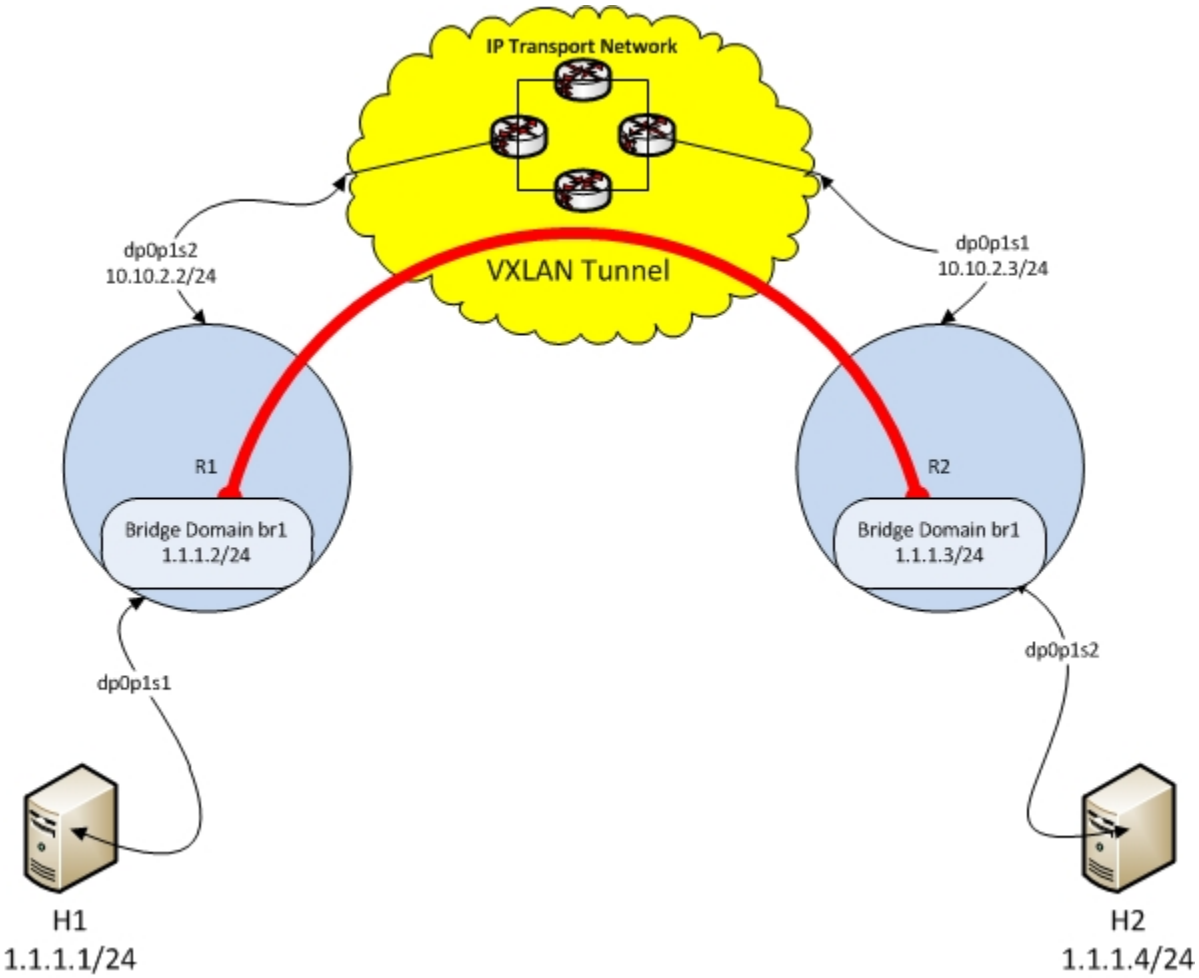
NOTE: AT&T currently supports only static VXLAN tunnels, not dynamic VXLAN tunnels. Dynamic VXLAN tunnels require the use of an IP multicast-listener on the tunnel endpoint nodes to detect multicast traffic. However, IP multicast is not enabled is some networks due to security and performance considerations. The static VXLAN configuration provides most of the same benefits available with the dynamic VXLAN configurations.

Configuring a VXLAN Tunnel

In the following example, Vyatta routers R1 and R2 are configured as VTEPs. H1 and H2 are hosts that are connected to R1 and R2 and the hosts communicate with each other over the VXLAN tunnel.

The VXLAN tunnel is established between the two VTEPs and it runs over a normal IP transport layer.

In this example, the two VTEPs are on the same subnet (10.10.2.x/24 network), but they don't have to be. The VTEPs can be on any subnet as long as they are reachable via normal routing protocols.



The hosts, H1 and H2 are on the overlay network. (1.1.1.0/24)

The VTEP, R1 is directly connected to H1; and the second VTEP, R2, is directly connected to H2. The transport network is 10.10.2.0/24.

The VXLAN tunnel and the interface connecting each host are members of a bridge (br1) on each router.

H1 and H2 can ping each other over the VXLAN tunnel, because the packets are bridged across the VXLAN tunnel.

This is a simple, traditional VXLAN tunnel. The full L2 payload is carried over the tunnel.

Configuring the Interface on Host1

Configure an interface on Host1 to connect to the bridge domain on R1.

Host1: Configuration Parameters Step	Command
Specify the interface that connects Host1 to the bridge domain on R1.	`vyatta@Host1 #set interfaces dataplane dp0p1s1 address 1.1.1.1/24`

Configuring a VXLAN Tunnel on the Router, R1

The following table provides the steps and commands to configure a VXLAN tunnel on R1.

R1 Configuration Parameters Step	Command
Specify the bridge domain address for traffic coming from H1,	`vyatta@R1#set interfaces bridge br1 address 1.1.1.2/24`
Specify that traffic transmitted between R1 and H1 uses interface dp0p1s1 and is associated with bridge br1.	`vyatta@R1#set interfaces dataplane dp0p1s1 bridge-group bridge br1`
Specify that interface dp0p1s2 connects to IP address 10.10.2.2.24 as the source VTEP for R1.	`vyatta@R1#set interfaces dataplane dp0p1s2 address 10.10.2.2/24`
Specifies that tunnel tun0 is to be used for bridge-group br1	`vyatta@R1# set interfaces tunnel tun0 bridge-group bridge br1`
Specifies that the tunnel tun0 is a VXLAN tunnel.	`vyatta@R1#  set interfaces tunnel tun0 encapsulation vxlan`
Specifies the local IP address to be used by the tunnel.	`vyatta@R1# set interfaces tunnel tun0 local-ip 10.10.2.2`
Specifies the remote IP address to be used by the tunnel.	`vyatta@R1# set interfaces tunnel tun0 remote-ip 10.10.2.3`
Specifies the VXLAN ID of the tunnel tun0.	`vyatta@R1#set interfaces tunnel tun0 vxlan-id 5000`

Configuring a VXLAN Tunnel on the Router, R2

The following table provides the steps and commands to configure a VXLAN on R2.

R2 Configuration Parameters Step	Command
Specify the bridge domain address.	`vyatta@R2#set interfaces bridge br1 address 1.1.1.3/24`
Specify that traffic transmitted between R2 and H1 uses interface dp0p1s2 and is associated with bridge br1	`vyatta@R2#set interfaces dataplane dp0p1s2 bridge-group bridge br1`
Specify that interface dp0p1s1 connects to IP address 10.10.2.3.24 as the source VTEP for R2.	`vyatta@R2#set interfaces dataplane dp0p1s1 address 10.10.2.3/24`
Specifies that tunnel tun0 is to be used for bridge-group br1	`vyatta@R2# set interfaces tunnel tun0 bridge-group bridge br1`
Specifies that the tunnel tun0 is a VXLAN tunnel.	`vyatta@R2#  set interfaces tunnel tun0 encapsulation vxlan`
Specifies the local IP address to be used by the tunnel.	`vyatta@R2# set interfaces tunnel tun0 local-ip 10.10.2.3`
Specifies the remote IP address to be used by the tunnel.	`vyatta@R2# set interfaces tunnel tun0 remote-ip 10.10.2.2`
Specifies the VXLAN ID of the tunnel tun0.	`vyatta@R2#set interfaces tunnel tun0 vxlan-id 5000`

Configuring the Interface on Host2

Configure an interface on Host2 to connect to the bridge domain on R2.

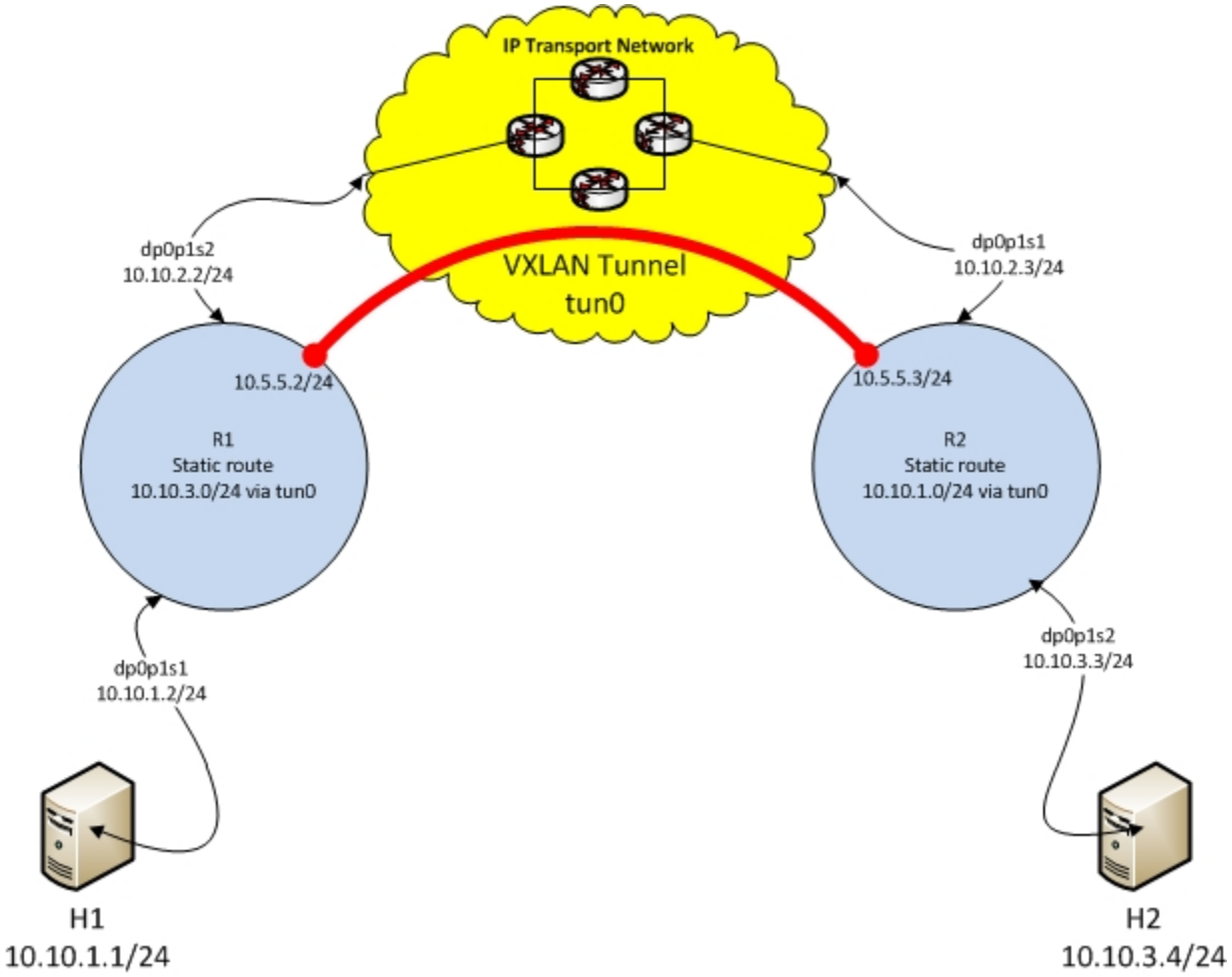
Host2: Configuration Parameters Step	Command
Specify the interface that connects Host2 to the bridge domain on R2	`vyatta@Host2#set interfaces dataplane dp0p1s1 address 1.1.1.4/24`

Configuring a VXLAN-GPE Tunnel

In the following example, Vyatta routers R1 and R2 are configured as VTEPs. H1 and H2 are hosts that are connected to R1 and R2 and the hosts communicate with each other over the VXLAN-GPE tunnel. In this case, the VXLAN-GPE tunnel is an L3 interface with its own IP address.

In this topography, the tunnel is terminated on the VTEP and the traffic is routed in to the tunnel (rather than being bridged into the tunnel as it was for the basic VXLAN).

The VXLAN tunnel is established between the two VTEPs and it runs over a normal IP transport layer.



Configuring the Interface and route on Host1

Configure an interface on Host1 to connect to R1 and a static route.

Host1: Configuration Parameters Step	Command
Specify the interface that connects Host1 to R1.	`vyatta@Host1#set interfaces dataplane dp0p1s1 address 10.10.1.1/24`
Create a static route to Host2 from Host1, R 1 is designated as the next hop.	`vyatta@Host1#set protocols static route 10.10.3.0/24 next-hop 10.10.1.2`

Configuring a VXLAN-GPE Tunnel on the Router, R1

The following table provides the steps and commands to configure a VXLAN-GPE tunnel on R2:

R1 Configuration Parameters Step	Command
Specify the interface for communications from Host1 to R1.	`vyatta@R1# set interfaces dataplane dp0p1s1 address 10.10.1.2/24`
Specify the interface for communications from R1 to the IP transport network.	`vyatta@R1#set interfaces dataplane dp0p1s2 address 10.10.2.2/24`
Specifies the local IP address of the tunnel, tun0.	`vyatta@R1#set interfaces tunnel tun0 address 10.5.5.2/24`
Specifies that the tunnel tun0 is a VXLAN-GPE tunnel.	`vyatta@R1# set interfaces tunnel tun0 encapsulation vxlan-gpe`
Specifies the local IP address of R1 in its connection to the IP transport network.	`vyatta@R1# set interfaces tunnel tun0 local-ip 10.10.2.2`
Specifies the IP address of R2 as its destination.	`vyatta@R1# set interfaces tunnel tun0 remote-ip 10.10.2.3`
Specifies the VXLAN ID of the tunnel tun0.	`vyatta@R1#set interfaces tunnel tun0 vxlan-id 5000`
Specifies that the static route from R1 to R2 uses the tunnel tun0 as its next-hop.	`vyatta@R1#set protocols static interface-route 10.10.3.0/24 next-hop-interface tun0`

Configuring a VXLAN-GPE Tunnel on the Router, R2

The following table provides the steps and commands to configure a VXLAN-GPE tunnel on R2.

R2 Configuration Parameters Step	Command
Specify interface for communications from Host2 to R2.	`vyatta@R2#set interfaces dataplane dp0p1s2 address 10.10.3.3/24`
Specify the interface that connects R2 to the IP transport network.	`vyatta@R2# set interfaces dataplane dp0p1s1 address 10.10.2.3/24`
Specifies the local IP address of the tunnel, tun0.	`vyatta@R2#set interfaces tunnel tun0 address 10.5.5.2/24`
Specifies that the tunnel tun0 is a VXLAN-GPE tunnel.	`vyatta@R2# set interfaces tunnel tun0 encapsulation vxlan-gpe`
Specifies the local IP address of R2 in its connection to the IP transport network.	`vyatta@R2# set interfaces tunnel tun0 local-ip 10.10.2.3`
Specifies the remote IP address of R2.	`vyatta@R2# set interfaces tunnel tun0 remote-ip 10.10.2.2`
Specifies the VXLAN ID of the tunnel tun0.	`vyatta@R2#set interfaces tunnel tun0 vxlan-id 5000`
Specifies that the static route from R2 to R1 uses the tunnel tun0 as its next-hop.	`vyatta@R1#set protocols static interface-route 10.10.1.0/24 next-hop-interface tun0`

Configuring the Interface and Route on Host2

Configure an interface on Host2 to R2 and a static route.

Host2: Configuration Parameters Step	Command
Specify the interface that connects Host2 to R2.	`vyatta@Host2# set interfaces dataplane dp0p1s1 address 10.10.3.4/24`
Specifies that the static route from Host2 to Host1 uses the tunnel tun0 as its next-hop.	`vyatta@Host2#set protocols static route 10.10.1.0/24 next-hop 10.10.3.3`

VXLAN Configuration Commands

This section lists the configuration commands that have been added to support VXLAN tunnels.

set interfaces tunnel <tunx> encapsulation < vxlan | vxlan-gpe>

Specifies whether the tunnel for a specified interface is a VXLAN or a VXLAN GPE tunnel.

Command Parameter	Description
`VXLAN`	Virtual Extensible LAN: An encapsulation developed to support running an overlay L2 network over an existing IPv4/IPv6 network. It was originally developed to increase scalability in data center/cloud computing environments by increasing the number of isolated L2 networks from 4096 (limited by a 12-bit VLAN ID) to 16 million (using a 24-bit VXLAN ID). Traffic belonging to the overlay network is encapsulated in a UDP packet that is routed over the underlying transport network. The payload within a VXLAN packet is required to be a full Ethernet frame.
`VXLAN-GPE`	Generic Protocol Encapsulation: A backwards-compatible extension to VXLAN to allow overlay traffic of several types (including but not limited to Ethernet) to be transported over the underlying IP network. The type of traffic being carried is indicated using a set of bits in the VXLAN header.
`tunx`	The identifier of a tunnel interface. The identifier ranges from tun0 through tunx, where x is a nonnegative integer.

- Use the `set` form of this command to create a VXLAN or a VXLAN-GPE tunnel for a tunnel interface.
- Use the `delete` form of this command to delete a VXLAN or a VXLAN-GPE tunnel for a tunnel interface.
- Use the `show` form of this command to display the type of tunnel encapsulation for a tunnel interface.

set interfaces tunnel <tunx> transport multicast-group <ipv4-address | ipv6-address>

Enable the interface of a specified VXLAN tunnel to send traffic to a multicast-group is using either an IPv4 or and IPv6 address.

Command Parameter	Description
`ipv4`	Specify the IPv4 address for the Broadcast, Unknown unicast and Multicast (BUM) packet.
`ipv6`	Specify the IPv6 address for the BUM packet.
`tunx`	The identifier of a tunnel interface. The identifier ranges from tun0 through tunx, where x is a nonnegative integer.

- Use the `set` form of this command to enable the VXLAN tunnel to handle multicast traffic between VXLANS.
- Use the `delete` form of this command to delete the multicast capability for a specified VXLAN tunnel interface.
- Use the `show` form of this command to display whether this VXLAN multicast group is for IPv4 or IPv6 traffic.

set interfaces tunnel <tunx> transport routing-instance <vrf-name>

Enable the interface of a specified VXLAN tunnel to send traffic using the specified virtual routing and forwarding routing instance.

Command Parameter	Description
`vrf-name`	The name of the virtual routing and forwarding instance that the transport network belongs to.
`tunx`	The identifier of a tunnel interface. The identifier ranges from tun0 through tunx, where x is a nonnegative integer.

- Use the `set` form of this command to direct the specified VXLAN tunnel through the specified VRF.
- Use the `delete` form of this command to delete the specified VRF for a specified VXLAN tunnel interface.
- Use the `show` form of this command to display the VRF for a specified tunnel.

set interfaces tunnel <tunx> vxlan-id <0-16777216>

Specifies the VXLAN ID of the specified tunnel.

Command Parameter	Description
tun*x*	The identifier of a tunnel interface. The identifier ranges from tun0 through tunx, where x is a nonnegative integer.
vxlan-id	The identifier of the VXLAN must be specified as a nonnegative integer from 0 to 16777216.

- Use the `set` form of this command to create the VXLAN tunnel.
- Use the `delete` form of this command to delete the specified VXLAN tunnel.
- Use the `show` form of this command to display whether the VXLAN tunnel.

set protocols static vxlan-mac interface <if-name> mac <mac-addr> remote-ip <ip-address> [vni <vni>]

Creates a static VXLAN tunnel.

Command Parameter	Description
if-name	VXLAN interface for which the static entry applies.
mac-addr	MAC address of a remote node.
ip-address	IPv4 or IPv6 address of remote VTEP
vni	VXLAN network identifier to use to reach the remote node.

VXLAN Operational Mode Commands

This section lists the operational mode commands that have been added to support VXLAN tunnels.

clear vxlan mac [interface <if-name>] [mac <mac-address>]

Removes forwarding entries from the VXLAN forwarding database.

- If no interface is specified, all dynamic entries are cleared.
- If an interface is specified without a MAC address, all dynamic entries learned on the specified interface are cleared.
- If both an interface and a MAC address are specified, only the specified entry is removed

Command Parameter	Description
<if-name>	Name of VXLAN interface on which the clear operation is to be performed.
<macaddr>	MAC address for which forwarding information should be removed.

show vxlan mac [interface <if-name>] [mac <macaddr>]

Displays the VXLAN forwarding database either for all VXLAN interfaces or just for the specified interface/mac.

Command Parameter	Description
<if-name>	VXLAN tunnel interface for which the MAC entries need to be displayed
<macaddr>	MAC address for which forwarding information should be displayed

Example:

```
vyatta@vm-vxlan-2:~$ show vxlan mac

Interface Mac Address       Type  VNI   IP Address
tun2      3e:92:72:c6:64:84  D   51000  10.10.1.1
tun2      42:b5:4:0:1:3      D   51000  10.10.1.1
```

Output Parameters	Meaning
Interface	VXLAN interface on which the MAC entry is learned/programmed.
MAC Address	Ethernet MAC address of remote node
Type	D(ynamic) or S(tatic) or L(ocal) MAC address entry D: Learned from the packet flows between the endpoints. S: Specified in the configuration L: A local made address
VNI	VXLAN Network Identifier used to reach the remote nodes.
IP Address	IPv4 or IPv6 address of the VXLAN Tunnel End Point (VTEP)

show vxlan statistics

Displays packet counters and counts of any errors that have occurred during VXLAN setup or forwarding. Only non-zero counter values are displayed.

Example:

```
vyatta@vm-vxlan-2:~$ show vxlan statistics
Output drops - ARP Failed : 5
Input drops - Bad Header : 3
Input drops - Bad Payload : 1
Input drops - Options : 2
Input drops - No headroom : 4
Input drops - VNI not found : 2
Input packets : 10
Output drops - ND failed : 1
Output drops : 3
Output drops - Encap failed : 2
Output drops - Unknown payload : 1
Output packets : 10
```