BROCADE

SOFTWARE INSTALLATION GUIDE

Brocade Vyatta Network OS Distributed Services Platform Installation and Provisioning Guide, 5.2R1

Supporting Brocade Distributed Services Platform Deployments

53-1004758-02
3 January 2017

Contents

Preface

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade.
Click the **Support** tab and select **Document Library** to access documentation on MyBrocade or www.brocade.com You can locate documentation by product or by operating system.

Release notes are bundled with software downloads on MyBrocade. Links to software downloads are available on the MyBrocade landing page and in the Document Library.

Document feedback

Quality is our first concern at Brocade, and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com
- By sending your feedback to documentation@brocade.com

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers should contact their OEM/solution provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to www.brocade.com and select **Support**.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for non-urgent issues: • Case management through the MyBrocade portal. • Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • Toll-free numbers are available in many countries. • For areas unable to access a toll-free number: +1-408-333-6061	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/solution provider, contact your OEM/solution provider for all of your product support needs.

- OEM/solution providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/solution provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/solution provider.

About this Guide

This guide describes how to install and configure Brocade Distributed Services platform.

Distributed Services Platform Overview

Important Considerations

The following considerations apply to the Brocade Distributed Services platform.

- Avoid restarting the Distributed Services platform control process (vplane-controller service) on the controller VM. Avoid restarting the vPlane controller process, as this can lead to a loss of configuration information from individual vPlane instances. If you need to restart, always reboot the controller VM.
- Virtual bridge commands are supported as described in this guide. These should be used for configuring distributed bridge on the Distributed Services platform. However, do not use the **bridge** commands that are described in vRouter config guide. To add member ports, use the **bridge-group** command. The associated **show** commands use the syntax **show bridge <vbr>**.

Distributed Services platform introduction

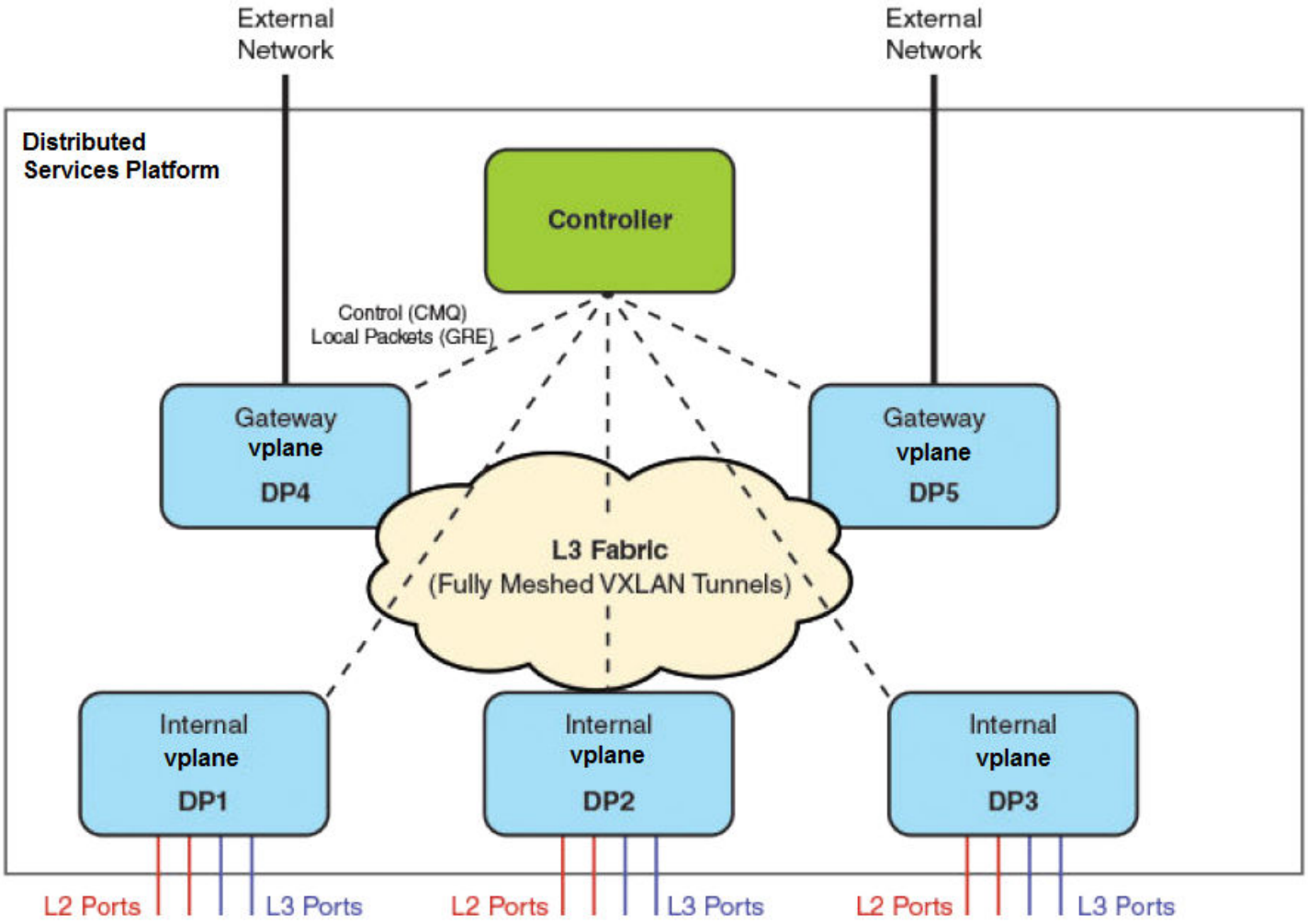
Brocade Distributed Services platform is a large-scale distributed router consisting of a single Distributed Services platform controller and multiple virtual data planes (vPlanes) operating together as a large distributed system across many hypervisors. Distributed Services platform provides a solution for the following:

- Horizontal scaling of VNFs, with numerous tenant-facing interfaces that are distributed across multiple vPlanes for Internet service providers.
- A Layer 2 virtual overlay network for cloud service providers.

The following diagram provides a high level view of the Distributed Services platform architecture, which is based on the following elements:

- Controller—Virtual machine (VM) that provides control and management functions for the Distributed Services platform infrastructure.
- vPlane—VM that forwards data as an instance of the data-forwarding plane. A single Distributed Services platform can include up to 32 vPlanes. Internal vPlanes connect to tenant servers, and gateway vplanes connect to the external network.
- Control network—Network that provides control plane and status communications between the Distributed Services Platform controller and vplanes.
- Fabric network—Full mesh of VXLAN-GPE tunnels between all the vPlanes that is used to forward packets between vPlanes.

FIGURE 1 Distributed Services platform architecture



VXLAN-GPE Usage

VXLAN-GPE is used to encapsulate all traffic flowing on the fabric network. To simplify and minimize user configuration of the fabric network, the VXLAN tunnels used on the fabric network are set up automatically without the need for the user to explicitly configure each tunnel.

VXLAN-GPE is used internally for packet encapsulation; however, VXLAN/VXLAN-GPE tunnels are not currently supported as a user-visible feature.

Distributed Services platform controller

The controller provides control and management functions for the Distributed Services platform infrastructure and is responsible for:

- Configuration and lifecycle management of the entire Distributed Services platform.
- Hosting the control plane for the supported routing protocols.

- Programming the forwarding tables on all vPlanes.
- Aggregating statistics received from the individual vPlanes.

The controller has the following interfaces:

- Management interface for management and diagnostic access
- Control interfaces for communication with the vPlanes.
- Fabric interface used to encapsulate traffic that originates or terminates on the Distributed Services platform system itself (controller VM).

Authentication between controller and vPlanes is optional. When enabled, it is handled by the 0MQ Elliptic Curve security facility. The controller and each vPlane produce two security certificates (keys): public and private. Each public key file must be copied to the peer system - the controller public key (in /etc/vyatta/controller.cert) must be propagated to each vPlane VM and each vPlane public key (in /etc/vyatta/dataplane-certs) must be copied to the controller VM.

vPlanes

A vPlane is an instance of the data forwarding plane. Each vPlane connects to the controller and to tenant hosts or the external network.

vPlanes are configured on separate virtual machines from the Distributed Services platform controller and are analogous to line cards on a distributed router or switch. The virtual machine for the vPlane can be colocated with the controller on the same hypervisor or reside on another hypervisor within the same data center. The configuration and interface topology of each vPlane is shared with all other vPlanes.

> NOTE
> The controller and vPlanes can be deployed in different data centers, but this type of configuration will result in higher latencies for network convergence and packet forwarding.

Each vPlane requires:

- A control interface for communicating control information to the controller.
- A fabric interface for forwarding traffic to and receiving traffic from other vPlanes.
- Multiple tenant interfaces used to connect end systems to their associated network.

The control interface is owned by the vPlane kernel, and the fabric and tenant interfaces are owned by the dataplane process running on the vPlane. Because the fabric interface on a vPlane provides the only communication path to all other vPlanes in the Distributed Services platform, it must be provisioned with adequate bandwidth to minimize over-subscription. For example, the fabric interface can be 10 G or 40 G while the other interfaces are 1 G or lower.

Each vPlane is assigned a forwarding role for internal or external communications.

- Internal – The non-fabric interfaces are connected to tenant hosts. Each tenant host connects to a single vplane.
- Gateway – The non-fabric interfaces serve as gateways from the internal vplanes to the external network.

Most Layer 3 and Layer 2 forwarding features are executed on the internal vplanes. Gateway vplanes provide load balancing functions for incoming traffic and external Layer 3 paths for outbound traffic.

Distributed Services platform fabric network

The Distributed Services platform fabric network is a full mesh of VXLAN-GPE tunnels among all the vPlanes and is used to forward packets between vPlanes. The tunnels carry traffic originating from the controller to the vPlane and terminating traffic from each vPlane to the controller.

The configuration of the fabric assumes an existing IP infrastructure in the data center, with the vPlane configuration layered on top.

All fabric interfaces in Distributed Services platform are configured to belong to a separate routing instance that is designated as the fabric routing instance.

Distributed Services platform control network

The Distributed Services platform control network is used for all control plane and status communications between the controller and vPlanes. The control plane connections are ZMQ running over TCP.

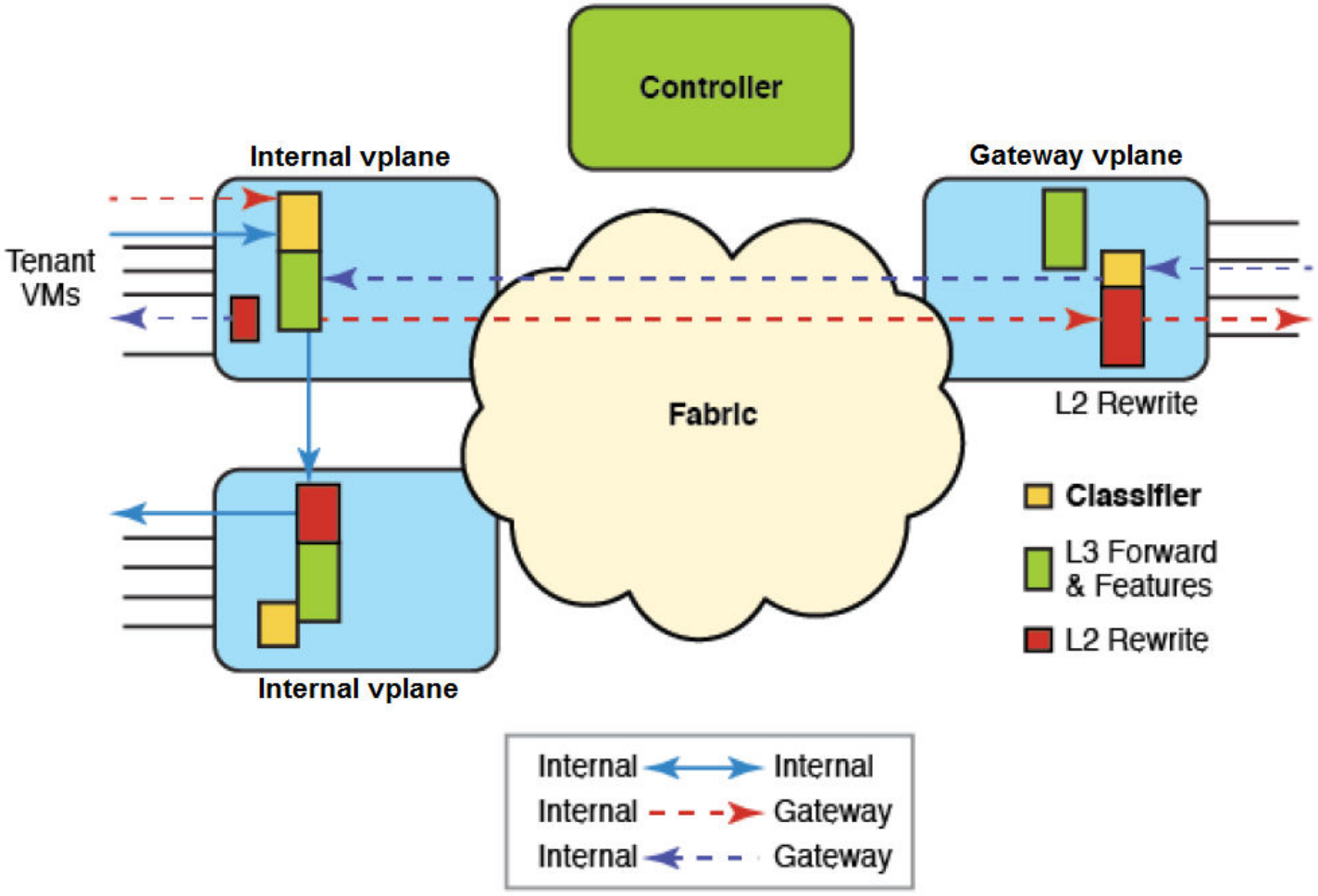
Distributed Services platform feature support

Distributed Services platform supports packet forwarding at the Layer 3 and Layer 2 levels.

Layer 3 forwarding

The following figure shows how Layer 3 forwarding works.

FIGURE 2 Layer 3 forwarding



Distributed Services platform manages these classes of flows:

- Internal – internal. Shown with solid blue lines in the figure. A packet from a tenant host enters an internal vPlane in the Distributed Services platform (solid blue line in the figure). A packet classifier determines where the processing for that packet will occur (on the current or another vPlane). The packet is forwarded to the processing vPlane, which processes the packet and then passes it back out to the internal network.
- Internal-gateway. A packet enters an internal vPlane, and the classifier determines that it will go out a gateway vPlane (to the external network). The classifier assigns an internal vPlane to process the packet, which is then sent out through the gateway vPlane.
- Gateway-internal. A packet enters a gateway vPlane, and the classifier determines that it will go out an internal vPlane (to a tenant host). The classifier assigns an internal vPlane to process the packet, which is then sent out through the internal vPlane.

The following Layer 3 features are supported:

- IPv4/IPv6 unicast forwarding
- ALG - ICMP, RCP, TFTP, PPTP
- Firewall

- SNAT (if multiple SNAT rules specify mapped-to port/address ranges, either the ranges must be non-overlapping or they must be 100% overlapping. Partial overlap in the ranges is not supported)
- PBR
- Point-to-point GRE tunnels (non-encrypted)
- Proxy ARP

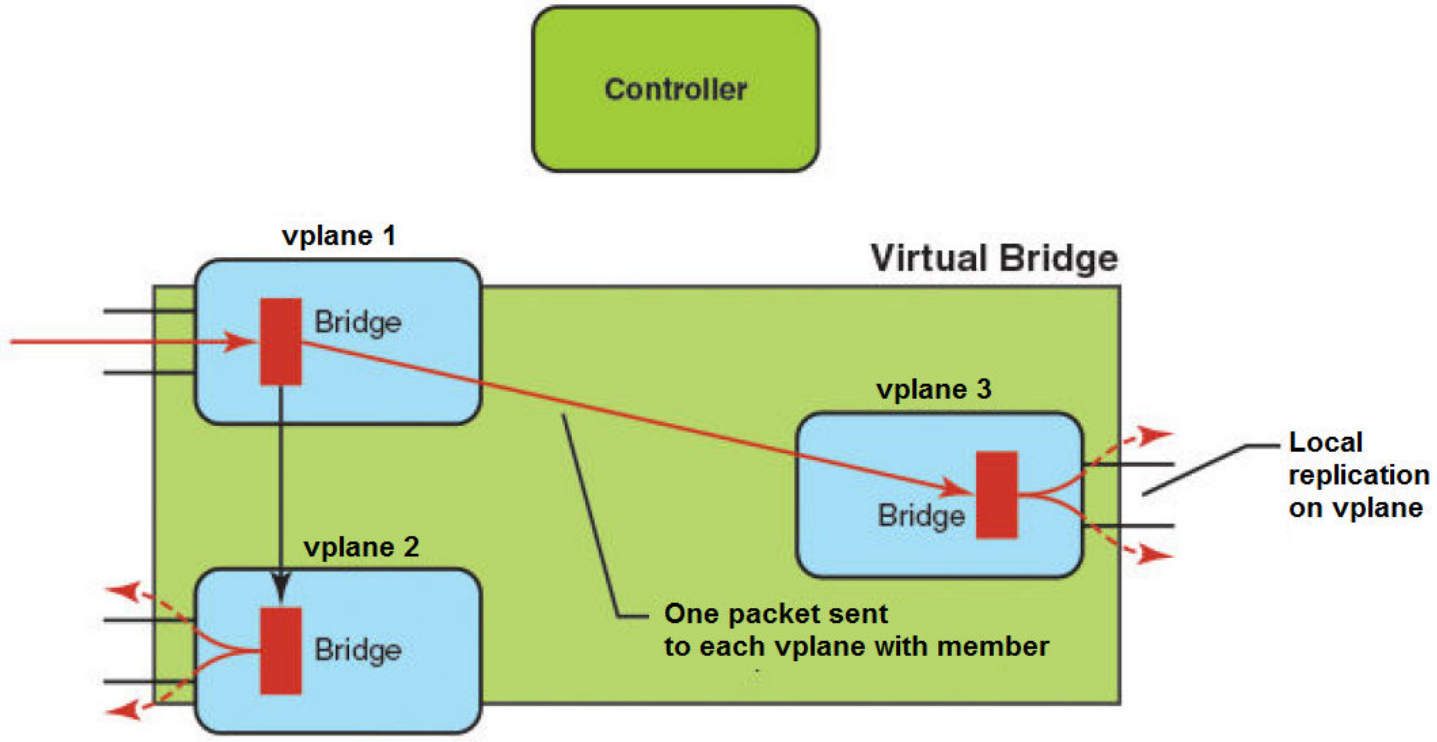
Note the following regarding PBR support:

- The creation of PBR route tables (using the **set protocols static table** command) consumes significant memory. Verify that you have at least 3 GB RAM assigned to each vPlane when provisioning a Distributed Services platform that might use PBR.
- Applying a PBR policy to any interface (tenant or gateway) causes subsequent Layer 3 feature processing to take place on the ingress vPlane. Instead of offloading Layer 3 feature processing to internal vPlanes, a gateway node will fully process incoming traffic before passing it to the internal vPlane for delivery to the tenant.

Layer 2 forwarding

Layer 2 forwarding for the Distributed Services platform occurs in the context of a bridge. When interfaces are designated as members of a bridge group, forwarding decisions for packets received on those interfaces are performed primarily using the bridge forwarding table.

FIGURE 3 Layer 2 Forwarding



The Distributed Services platform introduces a Layer 2 forwarding context called a virtual bridge, which allows a specific Layer 2 domain to span multiple vPlanes. This context allows interfaces from multiple vPlanes to be designated as members of the same virtual bridge. Each virtual bridge must be configured with a VXLAN Network Identifier (VNI), which is used to identify traffic belonging to a particular virtual bridge on the fabric.

As with Layer 3 forwarding, each tenant VM is reachable using exactly one Layer 2 interface on the Distributed Services platform. The topology is required to be loop-free, so STP and RSTP are not supported on the distributed bridge.

Flooding is supported for broadcast and unknown unicast packets in the context of a distributed virtual bridge. Each vPlane receives a single copy of every flooded packet and handles replication of those packets over locally hosted interfaces.

The following forwarding features are supported on virtual bridge interfaces:

- IRB—Each virtual bridge interface can be configured with a Layer 3 address that allows it to attract traffic belonging to the configured subnet. The presence of Layer 3 config also allows Layer 3 transitions for packets originating on member ports of the bridge.
- Layer 2 firewall—Stateless firewall rules can be associated with a virtual bridge interface (in a manner similar to a standalone bridge). The rules apply both to traffic being switched between member ports on the bridge and traffic being routed into/out of the bridge.
- Resolution Proxy—The Distributed Services platform performs ARP broadcast mitigation by mac learning on the fabric side.

MTU defaults and configuration

The Distributed Services platform uses GRE tunnels that terminate on the controller to model interfaces hosted on remote data plane nodes. This approach reduces the effective IP MTU on the interfaces by the size of the encapsulation header and metadata. To allow for other encapsulations that might have higher overheads and support correct operation with default MTUs in the underlying network, the default MTU on all data plane interfaces in the Distributed Services platform is set to 1400 bytes.

The MTU setting has following impact:

- There could be interoperability issues with protocols and applications that expect a 1500 byte IP MTU. For example due to MTU mismatch, OSPF peering will fail with neighbors that use the default configuration.
- Packets larger than 1500 bytes that transit the Distributed Services platform will be fragmented if the system uses the default MTU configuration.

Do either of the following to prevent the MTU issues from occurring:

Explicitly increase the MTU on all the relevant interfaces on the Distributed Services platform as follows:

- Increase the MTU on the control and fabric networks in the underlay to be greater than or equal to 1600.
- Increase the MTU on the control and fabric interfaces by using the **set interfaces system mtu** command.
- Increase the MTU on the dataplane interfaces to 1500 by using the **set interfaces dataplane mtu** command.

OR

Decrease the MTU on the connected routers to match the default MTU value of 1400.

Distributed Services Platform Configuration

Configuration overview

This section presents the tasks to establish a small Distributed Services platform and the configuration steps for each element of the system.

The examples use a KVM hypervisor. It is assumed that the core set of hypervisor networks has already been provisioned. The example networks are named "default" (management), "control" (control plane), "fabric" (fabric network), "net1" and "net2" (tenant networks)

Two software images are provided:

- Controller image, which contains all of the control plane software and the CLI infrastructure required to provision Distributed Services platform (livecd-dsp-ctrl-xxx).
- vPlane image that contains the Linux kernel, dataplane package, and diagnostic packages (livecd-dsp-dp-xxx).

The tasks are numbered and must be done in order.

Task 1: Provision networks for use within Distributed Services platform

1. Create the following network types to support Distributed Services platform:
    - A management network for SSH access to the controller and dataplane VMs. This network can be shared across one or more Distributed Services platforms.
    - A control network for communication between the controller and vPlanes. A separate control network is required for each Distributed Services platform instance. Use a network name that easily identifies the network purpose, such as DSPX_CTRL, where X is a number that identifies the instance.
    - A fabric network for packet traffic between vPlanes. A separate fabric network is required for each Distributed Services platform instance. Use a network name that easily identifies the network purpose, such as DSPX_FAB, where X is a number that identifies the instance.
    - In addition to these required networks, you can create as many networks as needed to provide connections between the vPlanes and other routers and tenant systems.

2. Increase the MTU on the underlying switch/underlay. For bare-metal or physical deployments, set the MTU to the maximum supported value on the switch ports that are used to connect the controller and data planes. For virtual deployments, set the MTU to the maximum supported value on the interfaces that are defined at the level of the hypervisor. Refer to the documentation for the network devices for detailed instructions.

Task 2: Provision the fabric network

Specify a routing instance as the fabric routing instance.

1. Create a routing instance.

    Example:

    ```
    vyatta@vyatta# set routing routing-instance DSPfabric
    ```

2. Designate the routing instance as the fabric routing instance.

    ```
    vyatta@vyatta# set distributed fabric routing-instance DSPfabric
    ```

Task 3: Provision each vPlane

To provision the vPlanes, perform the following steps on each individual vPlane. When installing the dataplane ISO, provision each dataplane VM with at least three interfaces - one for each of the networks described in task 1. Provide additional interfaces as needed to any other Distributed Services platform networks you created.

1. Install the Distributed Services platform dataplane ISO (livecd-dsp-dp-xxx) and reboot.
2. Use the command line on each vPlane to provide the system with SSH access, a host name, and sudo access to the vyatta account.

    ```
    vyatta@vyatta# set service 'ssh'
    vyatta@vyatta# set system login user vyatta level 'superuser'
    vyatta@vyatta# set system host-name 'dsp-vp3'
    ```

3. Log out and log back to cause the changes to take effect.
4. Generate a suitable network configuration for both the management and control interfaces. Use the command **set interfaces system <name> address <IP Address>**.

   Example:

```
vyatta@vyatta# set interfaces system ens0 address dhcp
vyatta@vyatta# set interfaces system ens0 description "Dataplane management network"
vyatta@vyatta# set interfaces system ens1 address 10.10.10.101/24
vyatta@vyatta# set interfaces system ens1 description "Dataplane 1 control network"
vyatta@vyatta# commit
vyatta@vyatta# exit
vyatta@vyatta:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:25:b2:db brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.89/24 brd 192.168.122.255 scope global ens0
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe25:b2db/64 scope link
       valid_lft forever preferred_lft forever
3: ens1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 52:54:00:05:d3:64 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.101/24 brd 10.10.10.255 scope global ens1
       valid_lft forever preferred_lft forever
    inet6 2001::101/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe05:d364/64 scope link
       valid_lft forever preferred_lft forever
vyatta@vyatta:~$
vyatta@vyatta:~$ sudo /usr/sbin/dmidecode -s system-uuid
7190A3C6-8806-D646-A6C9-784F705D8CC9
vyatta@vyatta:~$
```

5. Use the command **sudo /usr/sbin/dmidecode -s system-uuid** or **show version** to determine the system UUID.
6. Create the directory /etc/vyatta to store configuration files and certificates.

```
vyatta@vyatta:~$ sudo mkdir /etc/vyatta
```

Task 4: Install and configure the controller

Follow these steps to install and configure the controller.

1. Install the Distributed Services platform controller ISO (livecd-dsp-ctrl-xxx) and reboot.
2. Use the controller command line in Configuration mode to provide the system with SSH access, a host name, and sudo access to the vyatta account.

```
vyatta@vyatta# set service 'ssh'
vyatta@vyatta# set system login user vyatta level 'superuser'
vyatta@vyatta# set system host-name 'dsp-ctrl0'
```

3. Log out and log back to cause the changes to take effect.
4. You can now configure the Distributed Services platform as a whole: local controller interfaces, routing instance for the fabric network, control plane details, individual vPlanes, and the associated fabric interfaces. For the local management, control, and fabric interfaces, use the configuration command **set interfaces system <name> address <IP Address>**.

Example:

```
vyatta@vyatta#
vyatta@vyatta# set interfaces system ens0 address dhcp
vyatta@vyatta# set interfaces system ens0 description "Controller management network"
vyatta@vyatta# set interfaces system ens1 address 10.10.10.100/24
vyatta@vyatta# set interfaces system ens1 description "Controller control-plane network"
vyatta@vyatta# set interfaces system ens2 address 11.11.11.100/24
vyatta@vyatta# set interfaces system ens2 description "Controller fabric network"
vyatta@vyatta# set routing routing-instance DSPfabric interface ens2
vyatta@vyatta# commit
vyatta@vyatta#
vyatta@vyatta# set distributed fabric network GPE
vyatta@vyatta# set distributed fabric vni 10001
vyatta@vyatta# set distributed fabric routing-instance DSPfabric
vyatta@vyatta# set distributed controller address 10.10.10.100
vyatta@vyatta# set distributed controller fabric network GPE
vyatta@vyatta# set distributed controller fabric address 11.11.11.100
vyatta@vyatta# commit
vyatta@vyatta# exit
 vyatta@vyatta:~$'
```

5. Define each vPlane instance. Use the configuration command **set distributed dataplane < id> interfaces system name <name>** to identify the management and control interfaces on the vPlane. The interfaces named in these commands form an exclusion list in the vPlane configuration file and are ignored by the vPlane process.

   Example:

```
vyatta@vyatta# set distributed dataplane 1 address 10.10.10.101
vyatta@vyatta# set distributed dataplane 1 fabric network GPE
vyatta@vyatta# set distributed dataplane 1 interfaces system name ens0
vyatta@vyatta# set distributed dataplane 1 interfaces system name ens1
vyatta@vyatta# set distributed dataplane 1 uuid '7190A3C6-8806-D646-A6C9-784F705D8CC9'
vyatta@vyatta# set distributed dataplane 3 address '10.10.10.103'
vyatta@vyatta# set distributed dataplane 3 fabric network 'GPE'
vyatta@vyatta# set distributed dataplane 3 interfaces system name en0
vyatta@vyatta# set distributed dataplane 3 interfaces system name ens1
vyatta@vyatta# set distributed dataplane 3 uuid '1090852E-DDCC-0D45-8A75-49F310239D2E'
```

6. Increase the MTU on the system interfaces that will be used for the control and fabric networks.

```
vyatta@vyatta# set interfaces system ens1 mtu 8192
vyatta@vyatta# set interfaces system ens2 mtu 8192
```

7. Commit the configuration. A series of dataplane process configuration files is written to etc/vyatta/, one for each defined vPlane.

```
vyatta@vyatta# commit
vyatta@vyatta# exit
vyatta@vyatta:~$ ls -1 /etc/vyatta/
controller.conf
controller.conf-default
dataplane001.conf
dataplane003.conf
dataplane-certs
dataplane.conf-default
```

8. Verify that the controller VM can use the **ping** command to reach each dataplane VM on the control network.

Task 5: Copy the configuration to target vPlanes and start the dataplane process

1. If authentication is configured, copy the controller certificate to the vPlane and the vPlane certificate to the controller.

2. Copy the dataplane.conf file to the target vPlane VM and restart the dataplane.

   Example:

```
vyatta@vyatta:~$ sudo cat /etc/vyatta/dataplane.conf
 #
 # Configuration for vplane 1
 # auto-generated by controllercfg on Wed Sep 14 18:55:48 2016
 #
[Controller]
 ip = 10..10.10.100
 certificate = /etc/vyatta/controller.cert
 log = 20514

[Authentication]
 method = none

[Dataplane]
 ip = 10.10.10.101
 certificate = /etc/vyatta/dataplane.cert
 uuid = 1090852E-DDCC-0D45-8A75-49F310239D2E
 blacklist = ens0,ens1
 vyatta@vyatta:~$ sudo service vyatta-dataplane restart
```

3. Establish the fabric network by configuring each vPlane fabric interface with its address and moving each interface into the fabric routing instance that has already been defined in Task 4, step 4.

   Example:

```
vyatta@vyatta:~$ config
vyatta@vyatta# set interfaces dataplane dp1s5 address 11.11.11.5/24
vyatta@vyatta# set interfaces dataplane dp3s5 address 11.11.11.9/24
vyatta@vyatta# set routing routing-instance DSPfabric interface dp1s5
vyatta@vyatta# set routing routing-instance DSPfabric interface dp3s5
vyatta@vyatta# commit
```

Task 6: Verify that the controller can detect the vPlane interfaces

After the vyatta-dataplane service is restarted (previous procedure), the vPlane connects to the controller.

1. Use the **show distributed dataplane** command to verify that the vPlane has connected to the controller.

   The value UP in the STATE column indicates that the vPlane has connected.

   Example:

```
vyatta@vyatta:~$ show distributed dataplane
 ID  UUID                                 ADDRESS           #IF  STATE
  1  3E6946E1-669B-4777-A452-4177954DE274  192.168.25.2        6  UP
  2  3E6946E1-669B-4777-A452-4177954DE211  192.168.25.3        6  UP
  3  3E6946E1-669B-4777-A452-4177954DE245  192.168.25.4        0  DOWN
  4  3E6946E1-669B-4777-A452-4177954DE271  192.168.25.5        0  DOWN
```

2. Use the **show interfaces** command to see that the associated interfaces are now available on the controller.

   Example:

```
vyatta@vyatta:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
dp1s5            -                                 A/D
```

```
dp1s6            -                                  A/D
dp1s7            -                                  A/D
dp3s5            -                                  A/D
dp3s6            -                                  A/D
dp3s7            -                                  A/D
ens0             192.168.122.78/24                  u/u
ens1             10.10.10.100/24                    u/u
                 2001::100/64
ens2             -                                  A/D
```

3. Verify the complete configuration for the controller, fabric, and vPlanes.

   Example:

```
vyatta@vyatta:~$ show distributed dataplane
ID  UUID                                   ADDRESS           #IF  STATE
  1  41BDC0DC-42B7-9745-A5F0-E5F60C7893A8  14.14.14.102        2  UP
  2  9BC79DDF-C9C7-634D-8D84-99414C24DA1F  14.14.14.103        2  UP
vyatta@vyatta:~$


vyatta@vyatta:~$ show dataplane 1 fabric status
vyatta@vyatta:~$ show dataplane 1 fabric status
DP Name: dp1
Fabric tunnel (default VRF): NONE
Supported encaps: VXLAN-GPE
------------------------------------------------------------------------------
FTEP Table:
------------------------------------------------------------------------------
DP    IfIndex      IPv4 Addr            IPv6 Addr
------------------------------------------------------------------------------
1    8         11.11.11.5
2    10         11.11.11.9
------------------------------------------------------------------------------
```

Task 7: Configure the fabric endpoint for each vPlane

1. Identify the interface that will serve as the fabric endpoint for each vPlane.

   Example:

```
vyatta@vyatta:~$ config
vyatta@vyatta# set interfaces dataplane dp1s5 address 11.11.11.5/30
vyatta@vyatta# set interfaces dataplane dp3s5 address 11.11.11.9/30
vyatta@vyatta# set routing routing-instance DSPfabric interface dp1s5
vyatta@vyatta# set routing routing-instance DSPfabric interface dp3s5
vyatta@vyatta# commit
vyatta@vyatta# exit
vyatta@vyatta:~$
```

2. Verify the configuration.

   Example:

```
vyatta@vyatta:~$ show dataplane 1 fabric status
```

Distributed Services Platform Commands

clear distributed classifier statistics

Resets the classification counters on all vPlane instances.

Syntax

clear distributed classifier statistics

Modes

Operational mode

Usage Guidelines

Use this command to reset the classification counters on all vPlane instances.

clear distributed fabric health

Resets the total and last counters. Does not cancel any traffic measurements that are in progress.

Syntax

clear distributed fabric health

Modes

Operational mode

Usage Guidelines

Use this command to reset the total and last counters on all vPlane instances.

clear distributed fabric statistics

Resets the fabric counter on all vPlane instances.

Syntax

clear distributed fabric statistics

Modes

Operational mode

Usage Guidelines

Use this command to reset the fabric counter on all vPlane instances.

distributed controller address

Sets the IP address of a Distributed Services Platform controller.

Syntax

set distributed controller address *address*

show distributed controller address

Command Default

None

Parameters

ip-address

An IP address (IPv4 or IPv6).

Modes

Configuration mode

Configuration Statement

```
distributed controller {
    address address
}
```

Usage Guidelines

Use the **set** form of this command to set the address of the controller. You can specify one IPv4 address, one IPv6 address, or one of each.

Use the **show** form of this command to display the address of the controller.

distributed controller authentication method

Sets the authentication method for control connections between the Distributed Services platform controller and vplanes.

Syntax

set distributed controller authentication method { **none** | **elliptic-curve** }

show distributed controller authentication method

Command Default

none

Modes

Configuration mode

Configuration Statement

```
distributed controller {
    authentication {
        method none
        method elliptic-curve
    }
}
```

Usage Guidelines

Use the **set** form of this command on the controller to set the authentication method for control connections between the controller and vplanes:

- none: No authentication
- elliptic-curve: Elliptic curve authentication.

Use the **show** form of this command to display the authentication method for control connections between the controller and vplanes.

distributed controller fabric address

Specifies the Distributed Services platform controller endpoint address for fabric connections.

Syntax

set distributed controller fabric address *address*

show distributed controller fabric address

Command Default

NA

Parameters

address

IPv4 or IPv6 address.

Modes

Configuration mode

Configuration Statement

```
distributed controller {
    fabric {
        address address
    }
}
```

Usage Guidelines

Use the **set** form of this command on the controller to specify the controller endpoint address for fabric connections. You can define up to one address per address family (IPv4 and IPv6).

Use the **show** form of this command to display the controller endpoint address for fabric connections.

distributed controller fabric network

Specifies the fabric network to which the Distributed Services platform controller belongs.

Syntax

set distributed controller fabric network *name*

show distributed controller fabric network

Command Default

GPE

Parameters

name

Text string that identifies the fabric network.

Modes

Configuration mode

Configuration Statement

```
distributed controller {
    fabric {
        network name
    }
}
```

Usage Guidelines

Use the **set** form of this command on the controller to specify the fabric network name.

Use the **show** form of this command to display the fabric network name.

distributed controller logport

Specifies the listener port for rsyslog.

Syntax

set distributed controller logport *port*

show distributed controller logport

Command Default

20514

Parameters

port

The port number (1-65535).

Modes

Configuration mode

Configuration Statement

```
distributed controller {
    logport port
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify a listener port for rsyslog. The remote vPlanes use syslog to propagate logging information to the controller using the rsyslog protocol.

Use the **show** form of this command to display the listener port number for rsyslog.

distributed controller timeout

Sets the timeout for control plane communications between the Distributed Services platform controller and vPlanes.

Syntax

set distributed controller timeout *seconds*

show distributed controller timeout

Command Default

15

Parameters

seconds

The controller timeout (in seconds).

Modes

Configuration mode

Configuration Statement

```
distributed controller {
    timeout seconds
}
```

Usage Guidelines

Use the **set** form of this command on the controller to set the timeout for control plane communication between the controller and vPlanes.

Use the **show** form of this command to display the timeout for control plane communication between the controller and vPlanes.

distributed dataplane address

Sets the IP address of a vPlane.

Syntax

set distributed dataplane *vplane-id* **address** *address*

show distributed dataplane *vplane-id* **address**

Command Default

None

Parameters

vPlane-id

Unique identifier of the vPlane.

ip-address

IP address of the vPlane (ipv4 or ipv6).

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
        address address
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to set the vPlane IP address (IPv4 or IPv6).

Use the **show** form of this command to display the vPlane IP address.

distributed dataplane description

Specifies a text description of a vPlane.

Syntax

set distributed dataplane *vplane-id* **description** *description*

show distributed dataplane *vplane-id* **description**

Command Default

None

Parameters

vPlane-id

The ID of a vPlane.

description

Free-form text describing this vPlane instance.

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
        description description
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify a vPlane description.

Use the **show** form of this command to display a vPlane description.

distributed dataplane fabric network

Specifies the fabric network protocol for the vPlane.

Syntax

set distributed dataplane *vplane-id* **fabric network** *network*

show distributed dataplane *vplane-id* **fabric network**

Command Default

None

Parameters

vPlane-id

Unique identifier of the vPlane.

network

Name specified for the fabric network when it was created using the **set distributed fabric network** commands.

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
        fabric network network
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify the fabric network protocol for a vPlane.

Use the **show** form of this command to display the fabric network protocol for the vPlane.

distributed dataplane features

Specifies IPsec site-to-site VPN as a feature that is bound to a vPlane.

Syntax

set distributed dataplane *vplane-id* **ipsec site-to-site peer** *peer-id*

show distributed dataplane *vplane-id* **ipsec site-to-site peer** *peer-id*

Command Default

None

Parameters

vplane-id

Unique identifier of the vPlane.

peer-id

Identifier of the IPsec site-to-site peer.

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
        features {
              ipsec {
                 site-to-site {
                         peer peer-id
                 }
              }
         }
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify IPsec site-to-site VPN as a feature that is bound to a vPlane.

Use the **show** form of this command to display the feature configuration.

distributed dataplane interfaces dataplane

Provision local kernel interfaces for the Distributed Services platform controller.

Syntax

set distributed dataplane *vplane-id* **interfaces dataplane** { **name** *interface-name* | **type** *role* }

show distributed dataplane *vplane-id* **interfaces dataplane** [**name** *interface-name* | **type** *role*]

Command Default

None

Parameters

vplane-id

Unique identifier of the vPlane.

interface-name

Kernel interface name.

role

Interface role (admin or dataplane).

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
       interfaces {
          dataplane {
             name interface-name
             type type
          }
       }
    }
}
```

Usage Guidelines

Use the **set** form of this command on the controller to specify the kernel interfaces or the roles that the interfaces play.

Use the **show** form of this command to display the kernel interfaces or the roles that the interfaces play.

distributed dataplane interfaces tunnel

Specify tunnel interfaces to be bound to a vPlane.

Syntax

set distributed dataplane *vplane-id* **interfaces tunnel name** *interface-name*

show distributed dataplane *vplane-id* **interfaces**

Command Default

None

Parameters

vplane-id

Unique identifier of the vPlane.

interface-name

Tunnel interface name.

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
       interfaces {
          tunnel {
             name interface-name
          }
       }
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify the tunnel interface for a specified vPlane.

Use the **show** form of this command to display the data plane interfaces.

distributed dataplane interfaces virtual-bridge

Specify virtual bridge interfaces to be bound to a vPlane.

Syntax

set distributed dataplane *vplane-id* **interfaces virtual-bridge name** *interface-name*

show distributed dataplane *vplane-id* **interfaces**

Command Default

None

Parameters

vplane-id

Unique identifier of the vPlane.

interface-name

Virtual bridge interface name.

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
       interfaces {
          virtual-bridge {
             name interface-name
          }
       }
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify the virtual bridge interface for a specified vPlane.

Use the **show** form of this command to display the data plane interfaces.

distributed dataplane role

Specifies the type of vPlane (internal or gateway).

Syntax

set distributed dataplane *vplane-id* **role** { **internal** | **gateway** }

show distributed dataplane *vplane-id* **role**

Command Default

internal

Parameters

vplane-id

Unique identifier of the vPlane.

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
        role internal
        role gateway
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify the type of vPlane:

- Internal: Hosts only tenant or fabric-facing interfaces (does not have direct external connectivity).
- Gateway: Connects to the external network.

Use the **show** form of this command to display the type of vPlane.

distributed dataplane uuid

Sets the universally unique identifier (UUID) for a vPlane.

Syntax

set distributed dataplane *vplane-id* **uuid** *uuid*

show distributed dataplane *vplane-id* **uuid**

Command Default

None

Parameters

vplane-id

Unique identifier of the vPlane.

uuid

String representation of the vPlane UUID. Example: 1F7D1264-B47F-D040-AD7D-06F92FC441F4

Modes

Configuration mode

Configuration Statement

```
distributed {
    dataplane vplane-id {
        uuid uuid
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify a vPlane UUID.

Use the **show** form of this command to display the vPlane UUID.

distributed fabric network description

Specifies a text description of the fabric network.

Syntax

set distributed fabric network *name* **description** *string*

show distributed fabric network *name* **description**

Command Default

None

Parameters

name

Text string to identify the fabric network.

string

Text description of the fabric network.

Modes

Configuration mode

Configuration Statement

```
distributed fabric {
    network name {
        description description
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify a text description of the fabric network.

Use the **show** form of this command to display the text description of the fabric network.

distributed fabric network protocol

Sets the fabric network protocol.

Syntax

set distributed fabric network *name* **protocol** *protocol*

show distributed fabric network *name* **protocol** *protocol*

Command Default

vxlan-gpe

Parameters

name

Text string to identify the fabric network.

protocol

Protocol for the fabric network. The following values are supported:
vxlan: Standard VXLAN encapsulation protocol
vxlan-gpe: VXLAN generic protocol encapsulation

Modes

Configuration mode

Configuration Statement

```
distributed fabric {
    network name {
        protocol protocol
    }
}
```

Usage Guidelines

Use the **set** form of this command to configure on the Distributed Services platform controller to specify the fabric network protocol.

Use the **show** form of this command to display the fabric network protocol.

distributed fabric routing-instance

Specifies the VRF routing instance for the fabric network.

Syntax

set distributed fabric routing-instance *instance-name*

show distributed fabric routing-instance *instance-name*

Command Default

None

Parameters

instance-name

Name of a routing instance.

Modes

Configuration mode

Configuration Statement

```
distributed fabric {
    routing-instance instance-name
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to specify a routing instance for the fabric network.

Use the **show** form of this command to display the VRF routing instance.

distributed fabric vni

Sets the VXLAN Network Identifier (VNI) for the fabric network.

Syntax

set distributed fabric vni *vni*

show distributed fabric vni

Command Default

None

Parameters

vni

The VXLAN Network Identifier. Range is 0-16777215.

Modes

Configuration mode

Configuration Statement

```
distributed fabric {
    vni vni
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to set the fabric network VNI.

Use the **show** form of this command to display the fabric network VNI.

interfaces dataplane fabric

Configures the IP address of the fabric network default gateway.

Syntax

set interfaces dataplane *interface-name* **fabric default-gateway** { **ipv4** *ipv4-address* | **ipv6** *ipv6-address* }

set interfaces dataplane *interface-name* **fabric default-gateway**

Command Default

fabric

Parameters

interface-name
The interface name *(dpNsM)*.

ipv4-address
IPv4 address.

ipv6-address
IPv6 address.

Modes

Configuration mode

Configuration Statement

```
interfaces {
    dataplane interface-name  {
        fabric {
          default-gateway {
            ipv4 ipv4-address
            ipv6 ipv6-address
          }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command on the Distributed Services platform controller to set the IPv4 or IPv6 address of the fabric network default gateway.

Use the **show** form of this command to display the IP address fabric network default gateway.

show dataplane fabric statistics

Displays fabric statistics for the specified vPlane.

Syntax

show dataplane *vplane-id* **fabric statistics**

Parameters

vplane-id

Unique identifer of the vPlane.

Modes

Operational mode

Usage Guidelines

Use this command on the Distributed Services Platform controller to display fabric statistics for a specified vPlane.

Examples

The following example show fabric statistics for vPlane 1. It includes a fabric endpoint table containing the names, indices, and addresses of all fabric interfaces in the system.

```
vyatta@vyatta:~$ show dataplane 1 fabric statistics
DP Name: dp1
Fabric routing-instance: 2
Fabric tunnel (default VRF): def-fab-vxl
Supported encaps: VXLAN-GPE
------------------------------------------------------------------------
FTEP Table:
------------------------------------------------------------------------
DP    IfIndex      IPv4 Addr            IPv6 Addr
------------------------------------------------------------------------
1     10           11.1.1.2
2     0
3     11           11.1.1.3
513   0            11.1.1.1
vyatta@vyatta:~$
```

show dataplane fabric status

Displays the state of the fabric on the specified vPlane.

Syntax

show dataplane *vplane-id* **fabric status**

Parameters

vplane-id

Unique identifer of the vPlane.

Modes

Operational mode

Usage Guidelines

Use this command on the Distributed Services platform controller to display the state of the fabric on the specified vPlane. The output includes the following tables:

- Fabric endpoint table containing the names, indices, and addresses of all fabric interfaces in the system.
- Interface table containing the list of all interfaces instantiated on the specified vPlane and their corresponding local addresses.

Examples

The following example shows how to display the state of the fabric on a vPlane 1.

```
vyatta@vyatta:~$ show dataplane 1 fabric status
DP Name: dp1
Fabric tunnel (default VRF): NONE
Supported encaps: VXLAN-GPE VXLAN
------------------------------------------------------------------------------
FTEP Table:
------------------------------------------------------------------------------
DP    IfIndex      IPv4 Addr            IPv6 Addr
------------------------------------------------------------------------------
1    12         10.90.100.5      2020:2021::2005
2    10         10.90.100.9      2020:2021::2009
------------------------------------------------------------------------------
Intf Table
------------------------------------------------------------------------------
Name           IfIndex         Mac Addr          IPv4/IPv6 Addresses
------------------------------------------------------------------------------
dp1s9          15          72:8f:d3:4a:d0:34     10.20.31.131
                                                 2004:2005::2011
dp1s10         11          5a:67:a0:36:13:78
dp1s11         12           82:2:41:8e:dc:f0     10.90.100.5
                                                 2020:2021::2005
dp2s5          13          6a:32:cd:94:2b:46     10.20.32.181
                                                 2007:2009::2011
vbr500         6           3a:64:34:82:a3:16
dp2s11         10          fe:b9:a9:a9:63:70     10.90.100.9
                                                 2020:2021::2009
vxl-vbr500     7           3a:64:34:82:a3:16
dp2s6          14           62:a9:8:ba:f5:5a
vyatta@vyatta:~$
```

show distributed classifier statistics

Displays the classification result for each packet received by the Distributed Services platform.

Syntax

show distributed classifier statistics

Modes

Operational mode

Usage Guidelines

Use this command on the controller to present the following information about fabric packets:

- L2: Layer 2 packets seen by each vplane.
- IPv4: IPv4 packets seen by each vplane.
- IPv6: IPv6 packets seen by each vplane.

For each packet type, the output shows counts that are OK, dropped, and local (for IPv4 and IPv6).

Examples

The following example shows how to display fabric packet information for the full fabric network.

```
vyatta@vyatta:~$ show distributed classifier statistics
DP      L2 (OK     Drop)  IPv4 (OK    Local     Drop)  IPv6 (OK    Local     Drop)
1            0        0          0        0        0          9        9        0
2            2        0         17        0        0         81       19        0
3            2        0         15        0        0         88       20        0
vyatta@vyatta:~$
```

show distributed dataplane

Displays information about the vPlanes.

Syntax

show distributed dataplane

Modes

Operational mode

Usage Guidelines

Use this command on the Distributed Services platform controller to display information about the vPlanes.

Examples

The following example shows how to display information about the vPlanes.

```
vyatta@vyatta:~$ show distributed dataplane
 ID  UUID                                 ADDRESS          #IF  STATE
  1  3E6946E1-669B-4777-A452-4177954DE274  192.168.25.2       6  UP
  2  3E6946E1-669B-4777-A452-4177954DE211  192.168.25.3       6  UP
  3  3E6946E1-669B-4777-A452-4177954DE245  192.168.25.4       0  DOWN
  4  3E6946E1-669B-4777-A452-4177954DE271  192.168.25.5       0  DOWN
vyatta@vyatta:~$
```

show distributed fabric health

Displays the number of fabric packets transmitted and received by each vPlane within the fabric, providing an indication of the health of the fabric connection between vPlanes.

Syntax

show distributed fabric [*vplane-id*] [**peer***vplane-id2*] **health**

Parameters

vplane-id

Unique identifer of a vPlane.

vplane-id2

Unique identifer of a peer vPlane.

Modes

Operational mode

Usage Guidelines

Use this command on the Distributed Services platform controller to present the following information about fabric packets:

- Total Tx: Number of packets sent to the peer data plane.
- Tx drops: Difference between total Tx and the Rx reported by peer data plane.
- Sample Tx: Number of packets sent to the peer in the last acknowledged measurement window.
- Sample Tx drops: Difference between the sample Tx and the Rx reported by the peer data plane during the last acknowledged traffic measurement window.
- Hops Avg/Min/Max: Hop count of IPv4 and IPv6 routes between the peers.
- Sample seconds: Duration in seconds of the last measurement window.

Examples

The following example shows how to display fabric packet information for the full fabric network.

```
vyatta@vyatta:~$ show distributed fabric health
DP->DP   Total    Tx    Drops    Sample Tx     Drops  Hops Avg/Min/Max  Secs
1->2               76        0            7        0        0/ 0/ 0      10
2->1               78        0            8        0        0/ 0/ 0      10
2->3               28        0            7        0        0/ 0/ 0      10
3->2               28        0            5        0        0/ 0/ 0      10
```

The following example shows how to display fabric packet information for the packets leaving vPlane 2.

```
vyatta@vyatta:~$ show distributed fabric 2 health
DP->DP   Total    Tx    Drops    Sample Tx     Drops  Hops Avg/Min/Max  Secs
2->1               78        0            8        0        0/ 0/ 0      10
2->3              137        0           29        0        0/ 0/ 0      10
vyatta@vyatta:~$
```

The following example shows how to display fabric packet information for the traffic between vPlane 2 and 3.

```
vyatta@vyatta:~$ show distributed fabric 2 peer 3 health
DP->DP   Total    Tx    Drops    Sample Tx     Drops  Hops Avg/Min/Max  Secs
2->3             137        0           29         0        0/ 0/ 0      10
3->2             127       11           21        11        0/ 0/ 0      10
vyatta@vyatta:~$
```

show distributed fabric statistics

Displays information about systemwide activity on the fabric network.

Syntax

show distributed fabric statistics[**detail**]

Modes

Operational mode

Usage Guidelines

Use this command on the Distributed Services platform controller to display information about systemwide activity on the fabric network. The **detail** option includes pairwise transmit and receive counters for each active vPlane.

Examples

The following examples provide fabric packet statistics from each vPlane. The **detail** command option provides pairwise transmit and receive counters for each active vPlane.

```
vyatta@vyatta:~$ show distributed fabric statistics
DP       RxPkts       TxPkts      RxDrops      TxDrops
1            78           76            7            0
2           104          106            6            0
3            28           28            9            0

vyatta@vyatta:~$ show distributed fabric statistics detail

DP      Peer       RxPkts       TxPkts      RxDrops      TxDrops
1       2             78           76            1            0
1       3              0            0            0            0
1       513            0            0            0            0
2       1             76           78            0            0
2       3             28           28            0            0
2       513            0            0            0            0
3       1              0            0            1            0
3       2             28           28            1            0
3       513            0            0            0            0
vyatta@vyatta:~$
```