Managing Network Virtualization and Network Resources in Oracle® Solaris 11.4

Part No: E60989
November 2020

ORACLE®

Managing Network Virtualization and Network Resources in Oracle Solaris 11.4

Part No: E60989

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Référence: E60989

Restrictions de licence/Avis d'exclusion de responsabilité en cas de dommage indirect et/ou consécutif

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Exonération de garantie

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Avis sur la limitation des droits

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Avis sur les applications dangereuses

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Marques

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Inside sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Epyc, et le logo AMD sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Avis d'exclusion de responsabilité concernant les services, produits et contenu tiers

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Date de publication et mention de la version préliminaire de Disponibilité Générale ("Pre-GA")

Version préliminaire de Disponibilité Générale ("Pre-GA") : 15.01.2020

Avis sur la version préliminaire de Disponibilité Générale ("Pre-GA") de la documentation

Si ce document est fourni dans la Version préliminaire de Disponibilité Générale ("Pre-GA") à caractère public ou privé :

Cette documentation est fournie dans la Version préliminaire de Disponibilité Générale ("Pre-GA") et uniquement à des fins de démonstration et d'usage à titre préliminaire de la version finale. Celle-ci n'est pas toujours spécifique du matériel informatique sur lequel vous utilisez ce logiciel. Oracle Corporation et ses affiliés déclinent expressément toute responsabilité ou garantie expresse quant au contenu de cette documentation. Oracle Corporation et ses affiliés ne sauraient en aucun cas être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'utilisation de cette documentation.

Mention sur les informations confidentielles Oracle

INFORMATIONS CONFIDENTIELLES ORACLE. Destinées uniquement à un usage autorisé. Ne pas distribuer à des tiers.

Avis sur la reconnaissance du revenu

Si ce document est fourni dans la Version préliminaire de Disponibilité Générale ("Pre-GA") à caractère privé :

Les informations contenues dans ce document sont fournies à titre informatif uniquement et doivent être prises en compte en votre qualité de membre du customer advisory board ou conformément à votre contrat d'essai de Version préliminaire de Disponibilité Générale ("Pre-GA") uniquement. Ce document ne constitue en aucun cas un engagement à fournir des composants, du code ou des fonctionnalités et ne doit pas être retenu comme base d'une quelconque décision d'achat. Le développement, la commercialisation et la mise à disposition des fonctions ou fonctionnalités décrites restent à la seule discrétion d'Oracle.

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité de la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse : http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info ou le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs si vous êtes malentendant.

Contents

Figures

Tables

Examples

Using This Documentation

- **Overview** – Describes how to configure the Oracle Solaris virtual networking features and monitor network traffic. It also describes the different processes that are used to manage network resources.
- **Audience** – System administrators.
- **Required knowledge** – Basic and some advanced network administration skills.

Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/pls/topic/lookup?ctx=E37838-01.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPTER 1

Introduction to Network Virtualization and Network Resource Management

This chapter provides an overview of network virtualization and network resource management in Oracle Solaris. It contains the following topics:

What's New in Network Virtualization and Network Resources

- **Create VNICs over paravirtualized IPoIB datalinks** – In Oracle Solaris Kernel Zones, you can create an IPoIB VNIC over a paravirtualized IPoIB datalink. The IPoIB VNIC enables you to create a non-global zone inside a kernel zone. See "Creating VNICs Over Paravirtualized IPoIB Datalink" on page 89.
- **Configure offload policy for a NIC** – To utilize NIC resources effectively, you can configure the offload policy for a NIC by using the `hw-flow` property. See "Setting Properties for Flows" on page 194.
- **Configure a group of flow filters for a flow** – You can configure a flow that consist of multiple flow filters. See "Configuring a Flow With a Group of Flow Filters" on page 200
- **Display per-ring statistics and input and output packet drops for an aggregation** – The `dlstat show-aggr` command can display per-ring statistics, and input and output packet drops for an aggregation. See "Displaying Network Traffic Statistics of Link Aggregations" on page 216.
- **Create VNICs with dedicated ring groups to support hardware SLAs of VNICs** – VNICs can have their own dedicated transmit (Tx) and receive (Rx) ring groups. These

VNICs have better resource isolation, hardware SLA enforcement, and hardware anti-spoofing support. See “Creating VNICs With Dedicated Ring Groups” on page 178.

- **Identify the link status of an IPoIB VNIC** – You can use the datalink property, `broadcast-group` to check whether the IPoIB VNIC has failed to join the broadcast multicast group. See “Identifying the Link Status of an IPoIB VNIC” on page 44.

Overview of Network Virtualization

Network virtualization is an OS-provisioned mechanism that enables you to programmatically create and configure virtual networks that are decoupled from the underlying physical network. A virtual network is therefore a pseudo network that uses the physical network only as a packet forwarding backbone. You can virtualize entire network topologies of servers, routers, switches, and firewalls all running on a single platform and requiring no additional investment in networking hardware.

In virtual networks, virtual machines (VMs) are provisioned that run instances of the operating system. The VMs are isolated from one another but communicate with each other within the network.

Note - IP addresses that are used in Oracle Solaris 11 documentation conform to RFC 5737, IPv4 Address Blocks Reserved for Documentation (`https://tools.ietf.org/html/rfc5737`) and RFC 3849, IPv6 Address Prefix Reserved for Documentation (`https://tools.ietf.org/html/rfc3849`). IPv4 addresses used in this documentation are blocks 192.0.2.0/24, 198.51.100.0/24, and 203.0.113.0/24. IPv6 addresses have prefix 2001:DB8::/32.

To show a subnet, the block is divided into multiple subnets by borrowing enough bits from the host to create the required subnet. For example, host address 192.0.2.0 might have subnets 192.0.2.32/27 and 192.0.2.64/27.

Network Virtualization Technologies That Are Supported by Oracle Solaris

Oracle Solaris supports the following network virtualization technologies:

- **Edge Virtual Bridging (EVB)** – Enables a host to exchange information related to virtual links on a system with an external switch. EVB is used to exchange information about all the virtual links behind a port whereas data center bridging (DCB) is used to exchange information about the port. See Chapter 4, “Using Edge Virtual Bridging”.

- **Virtual Extensible Local Area Network (VXLAN)** – VXLAN addresses the 4K limitation of virtual local area network (VLAN) and also reduces the demand of virtualization on physical infrastructure such as switches. It uses physical server resources effectively in a data center that spans multiple L2 networks and provides scalability and network isolation for virtual networks. See Chapter 3, "Configuring Virtual Extensible Local Area Networks".
- **Single Root I/O Virtualization (SR-IOV)** – Enables the creation of a virtual function (VF) based VNIC on a network device that supports SR-IOV. See "Using Single Root I/O Virtualization With VNICs" on page 69.
- **Private Virtual Local Area Network (PVLAN) VNICs** – Enables you to configure PVLAN VNICs that are used for dividing a VLAN into sub-VLANs to isolate the network traffic thereby providing better usage of the limited number of available VLANs.

Network Virtualization Components

This section describes the components of network virtualization.

Virtual Network Interface Card (VNIC)

A *VNIC* is an L2 entity or virtual network device that is configured over a physical datalink. VNICs are created either by an administrator or by the system. The VNICs are then assigned to zones to enable the zones to exchange network traffic.

A VNIC has an automatically generated MAC address. However, you can also assign a different MAC address to a VNIC if preferred.

Virtual Switch

A *virtual switch* is an entity that facilitates communication between virtual machines (VMs) that share the same datalink. The virtual switch loops traffic between virtual machines (inter-VM traffic) within the physical machine and does not send this traffic out on the wire. A virtual switch is implicitly created whenever you create a VNIC on top of an underlying datalink. The VNICs configured with the VMs need to be on the same VLAN or VXLAN for inter-VM communication.

In certain cases, communication between VMs in a system might require the use of a switch. For example, communication between VMs might need to be subjected to access control lists (ACLs) that are configured on the switch. By default, a switch cannot send packets on the same port where the packets are received. Therefore, reflective relay is enabled on the switch for

communication between VMs that use a switch. Reflective relay enables the switch to forward the packets on the same port where the packets are received.

Etherstub Virtual NIC

An *etherstub* is a pseudo Ethernet NIC that is configured at the datalink layer (L2) of the Oracle Solaris network stack. You can create VNICs over an etherstub instead of over a physical NIC. With etherstubs, you can construct a private virtual network that is isolated both from the other virtual networks on the system and from the external network. For example, you can use etherstubs to create a network environment without the external connectivity or resources.

Elastic Virtual Switch

The Oracle Solaris Elastic Virtual Switch (EVS) feature provides virtual networking infrastructure that interconnects virtual machines that reside on multiple systems. EVS enables centralized management of virtual switches on multiple hosts and VNICs connected to the elastic virtual switch. Virtual machines connected to the same elastic virtual switch can communicate with each other.

Types of VMs for Network Virtualization in Oracle Solaris

Although you can assign VNICs to resources in a single instance of the Oracle Solaris, you can extend their use in network virtualization by using them in virtualized environments such as Oracle Solaris Zones, Oracle Solaris Kernel Zones, or Oracle VM Server for SPARC.

Oracle Solaris Zones

A *zone* is a virtualized operating system environment created within a single instance of Oracle Solaris. Etherstubs and VNICs are only a part of the virtualization features of Oracle Solaris. By assigning VNICs or etherstubs for use by Oracle Solaris zones, you can create a network within a single system.

An Oracle Solaris Kernel Zone, also called a `solaris-kz` branded zone, uses the branded zones framework to run a zone with a separate kernel and operating system (OS) installation from the global zone. The separate kernel and OS installation provide for greater independence and enhanced security of operating system instances and applications.

For information about zones, refer to the zone documentation on the *Creating and Using Oracle Solaris Virtual Environments* shelf of the Oracle Solaris 11.4 Information Library.

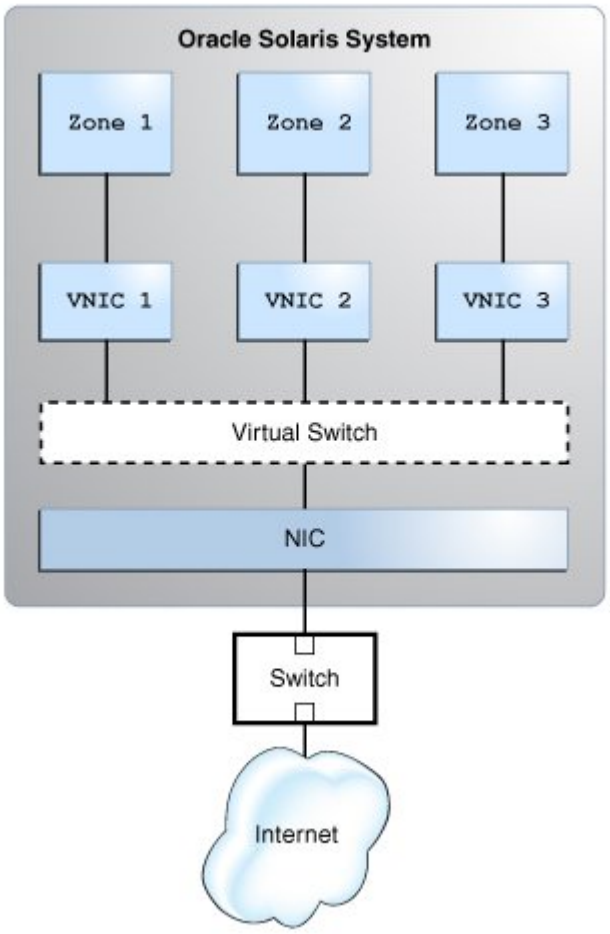
Oracle VM Server for SPARC

Oracle VM Server for SPARC provides highly efficient, enterprise-class virtualization capabilities for SPARC T-Series, SPARC M5, Fujitsu SPARC M12, and Fujitsu M10 platforms. You can create virtual servers called "logical domains" that can run an instance of an operating system to enable multiple operating systems on the same computer. For more information, see the appropriate administration guide in the Oracle VM Server for SPARC (`https://docs.oracle.com/en/virtualization/oracle-vm-server-sparc/`) documentation library.

How a Virtual Network Works

The following figure shows the working of a virtual network and its components in a system.

FIGURE 1 Working of a Virtual Network



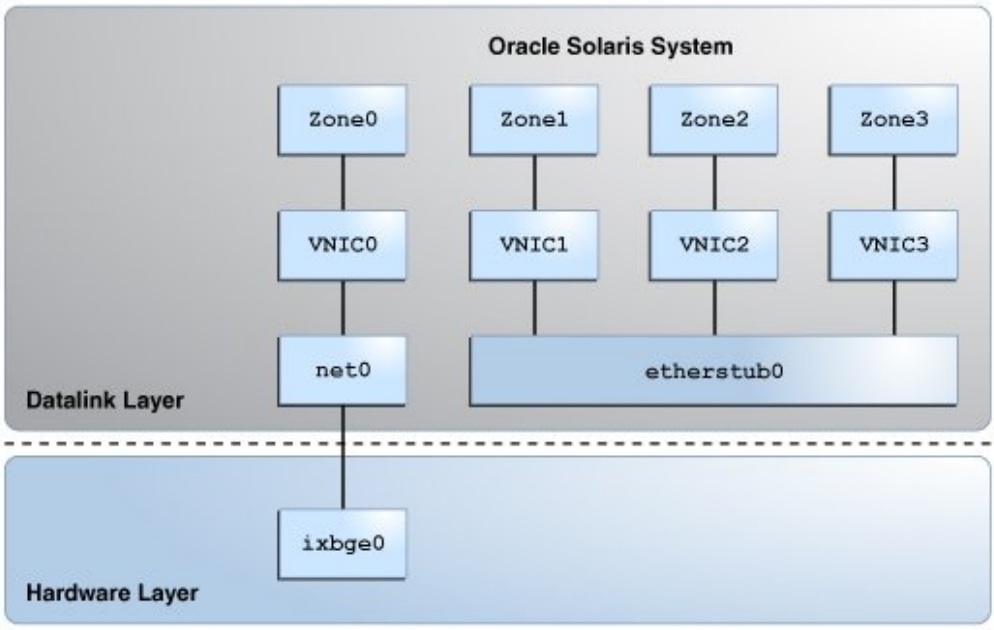
The figure shows a single system with one NIC. The NIC is configured with three VNICs. Each VNIC is assigned to a zone. The zones communicate with each other and with the external network by using their respective VNICs. The three VNICs connect to the underlying physical NIC through the virtual switch. The function of a virtual switch is equivalent to the function of a physical switch, which is to provide connectivity to the systems.

When a virtual network is configured, a zone sends traffic to an external host as usual. Traffic flows from the zone, through the VNIC to the virtual switch, and then to the physical interface, which sends the data to the network.

The zones can also exchange traffic with one another inside the system if all the VNICs configured to the zones are part of the same VLAN. The traffic is managed by the virtual switch. In this case, the traffic never leaves the system, and therefore never violates the Ethernet restrictions.

You can also create a virtual network based on the etherstub. Etherstubs are entirely software based and do not require a network interface as the basis for the virtual network. See the following figure:

FIGURE 2 Private Virtual Network



This figure shows `etherstub0` over which `VNIC1`, `VNIC2`, and `VNIC3` are configured. Each VNIC is assigned to a zone. The private virtual network based on the etherstub cannot be accessed by external networks. For more information, see "Use Case: Configuring a Private Virtual Network" on page 53.

You can use Oracle Enterprise Manager to manage some aspects of network virtualization, for example, the ability to create virtual networks inside a virtual data center. For more information about Oracle Enterprise Manager, see https://www.oracle.com/enterprise-manager/technologies/.

With the release of Oracle Virtual Networking Drivers for Oracle Solaris, Oracle Virtual Networking now supports Oracle Solaris on x86 and SPARC servers. For more information about Oracle Virtual Networking, see Oracle Virtual Networking Documentation (https://docs.oracle.com/cd/E38500_01/).

Overview of Network Resource Management

In Oracle Solaris, quality of service (QoS) is obtained more easily and dynamically by managing network resources. Network resource management is comparable to creating dedicated lanes for traffic. When you combine different resources to provide to the specific types of network packets, those resources form a network lane for those packets. Resources can be assigned differently for each network lane. For example, you can allocate more resources to a lane where network traffic is the heaviest. By configuring network lanes where resources are distributed according to the actual need, you increase the system's efficiency in processing network packets. See “Overview of Monitoring Network Traffic Statistics of Datalinks and Flows” on page 207.

By using network resource management, you can isolate, prioritize, track, and control data traffic on an individual system without the complex QoS rule definitions.

The following network resources are used to increase the system's efficiency in processing packets:

- **Bandwidth** – You can limit the bandwidth of the datalink according to the actual need of the networking processes using by the datalink.
- **Priority** – You can prioritize the order in which the packets are processed. The latency is reduced for the packets with higher priority because they are processed ahead of the other packets.
- **NIC rings** – If a NIC supports ring allocation, its transmit and receive rings can be dedicated for use by datalinks.
- **CPU pools** – Pools of CPUs are created and associated with specific zones. These pools can be further assigned to datalinks to manage the network processes of their associated zones.
- **CPUs** – On a system with multiple CPUs, you can dedicate a given number of CPUs for specific network processing.

Network resources on a system can be managed by using either datalink properties or flows.

Managing Network Resources Through Datalinks

Managing network resources by using datalinks improves the system's efficiency in processing packets. You can allocate resources when you create the link. Alternatively, you can allocate resources to a datalink, for example, after studying resource usage over time and determining how to better allocate the resource. By allocating network resources, you can decide the amount of a given resource that can be used for the networking processes. The procedures for allocating resources apply to the virtual network as well as the physical network. For more information about datalink properties and how to configure them, see "Managing Network Resources by Using Datalink Properties" on page 175.

Managing Network Resources Through Flows

A *flow* is a customized way of categorizing network packets based on a single attribute or a combination of attributes. Flows help you to differentiate different services on the same datalink. The attributes that serve as the basis for creating flows are derived from the information in a network packet's header. After setting datalink properties for network resource management, flows can be used to further control how resources are used to process network packets. Flows alone can also be used to manage network resources without setting datalink properties.

Using flows for managing resources involves the following steps:

1. Creating a flow based on a single attribute or a combination of attributes.
2. Customizing a flow's use of resources by setting properties that pertain to network resources. Currently, bandwidth, priority, and rank properties can be associated with flows.

For more information about configuring flows, see "Managing Network Resources by Using Flows" on page 191.

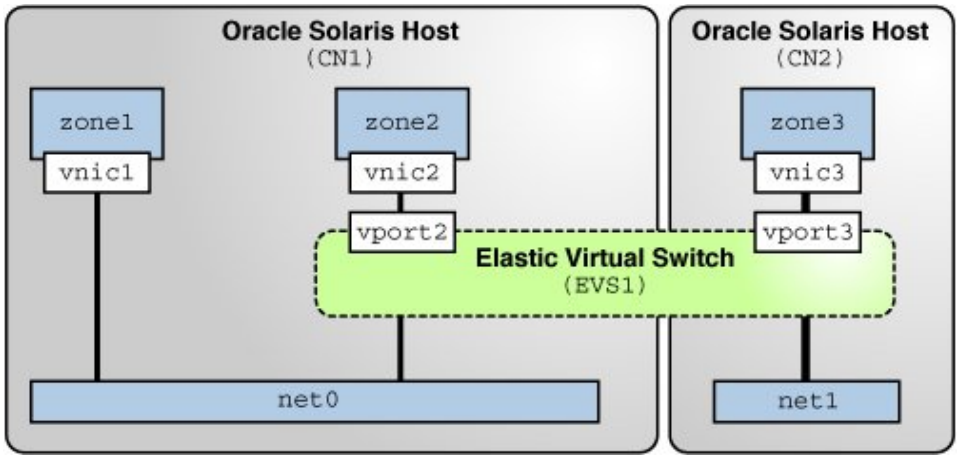
Scenario: Combining Network Virtualization and Network Resource Management

Network virtualization with network resource management helps you to manage flow control, improve system performance, and configure the network utilization needed to achieve OS virtualization, utility computing, and server consolidation. This section uses a scenario to show

how to use network virtualization with network resource management to optimize system performance.

The following figure shows the network virtualization setup that is used in this scenario.

FIGURE 3 Use Case: Network Virtualization Setup



The setup consists of the following components:

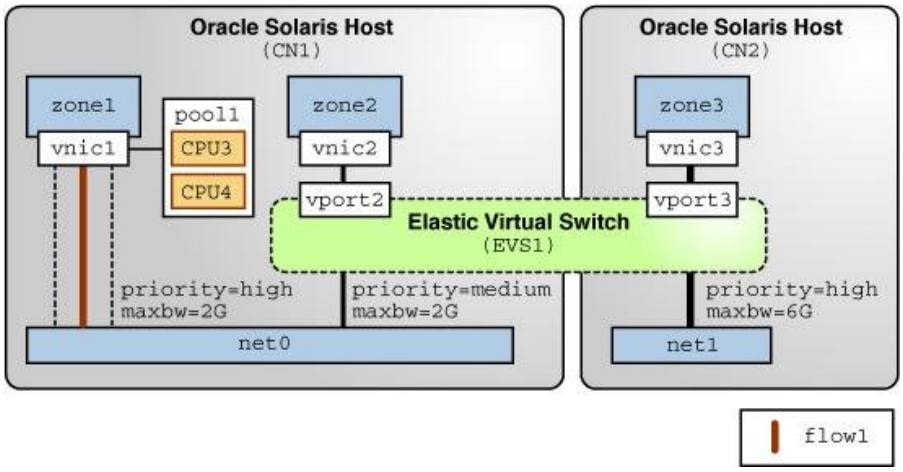
- Oracle Solaris hosts `CN1` and `CN2` that are configured with the datalinks `net0` and `net1` respectively.
- Oracle Solaris Zones `zone1` and `zone2` that are configured on `CN1` and `zone3` that is configured on `CN2`.
- Zones `zone1`, `zone2`, and `zone3` that are configured with the VNICs `vnic1`, `vnic2`, and `vnic3` respectively.
- Elastic virtual switch `EVS1` that is set up between `zone2` on `CN1` and `zone3` on `CN2`. Zones `zone2` and `zone3` are in the same network and hence are configured on the same elastic virtual switch.
- Virtual ports `vport2` and `vport3` that are the points of attachment between the VNICs and `EVS1`.

You can allocate network resources based on the priority and the rate of processing packets of different applications that run on zones. You can use datalink properties and flows to allocate network resources for the VNICs appropriately. This scenario is based on the following assumptions for allocating network resources:

- `zone1` hosts the applications `app1` and `app2`. You need to configure a flow on `vnic1` to isolate traffic and implement control over how packets belonging to the flows use resources. You also need to configure a separate pool of CPUs for `vnic1`.
- `zone2` hosts the application `app3` that communicates with a database on `zone3`.
- `zone3` hosts a database that communicates with `zone2`. Because `vnic3` receives and transmits more packets than the other VNICs, it uses more bandwidth. So, you need to set a higher bandwidth limit to cap the bandwidth usage of `vnic3`. You can also set a high priority for packet processing.

Based on these assumptions, the network resources are allocated for the VNICs that are configured on the Oracle Solaris hosts. The following figure shows the allocation of network resources for the VNICs by using the datalink properties and flows.

FIGURE 4 Use Case: Network Virtualization With Flows and Resource Allocation



This figure shows the datalink and flow properties that are set for the VNICs and `flow1`. The following table describes these properties and their values.

TABLE 1 Datalink and Flow Properties of VNICs

VNICs	Datalink Properties	Flow Properties	Description

`vnic1`	`pool=pool1`	`priority=high`
		
		`maxbw=2G`
			`flow1` is created on `vnic1` based on the transport protocol and IP address. The `priority` and `max-bw` properties are set to `flow1` to control the flow packets.

VNICs	Datalink Properties	Flow Properties	Description
			Additionally, the `pool` property is set to allocate a pool of CPUs to `vnic1`.

`vnic2`	`priority=medium`
	
	`max-bw=2G`
			The `max-bw` property is set to `vnic2` to allocate bandwidth. The `priority` property is set to `medium` by default.

`vnic3`	`priority=high`
	
	`max-bw=6G`
			The properties `max-bw` and `priority` are set to `vnic3` to allocate bandwidth and prioritize the order in which the packets are processed.

Using Rights Profiles to Perform Network Configuration

Oracle Solaris implements role-based access control (RBAC) to control system access. To perform tasks associated with network configuration, you must be assigned at least the Network Management profile. This profile is a superset that consists of other network-related profiles such as in the following partial list:

- Name Service Management for configuring name services.
- Network Wifi Management for configuring WiFi.
- Elastic Virtual Switch Administration for configuring the elastic virtual switch.
- Network Observability for accessing observability devices.

To obtain a complete list of the profiles in the Network Management profile, type:

```
$ profiles -p "Network Management" info
```

An administrator that has the `solaris.delegate.*` authorization can assign the Network Management profile to users to enable them to administer the network.

For example, an administrator assigns the Network Management rights profile to user `jdoe`. Before `jdoe` executes a privileged network configuration command, `jdoe` must be in a profile shell. The shell can be created by issuing the `pfbash` command. Or, `jdoe` can combine `pfexec` with every privileged command that is issued, such as `pfexec dladm`.

As an alternative, instead of assigning the Network Management profile directly to individual users, a system administrator can create a role that would contain a combination of required profiles to perform a range of tasks.

Suppose that a role `netadmin` is created with the profiles for network configuration as well as zone creation and configuration. User `jdoe` can issue the `su` command to assume that role. All roles automatically get `pfbash` as the default shell.

For more information about rights profiles, see "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

♦ ♦ ♦ CHAPTER 2

Creating and Managing Virtual Networks

This chapter describes tasks for building and managing a virtual network on a single system, including creating a virtual function (VF) based VNIC on a network device that supports single root I/O virtualization (SR-IOV).

The chapter contains the following topics:

- "Configuring the Components of a Virtual Network"
- "Building Virtual Networks"
- "Managing VNICs"
- "Using Single Root I/O Virtualization With VNICs"
- "Creating and Viewing Paravirtualized IPoIB Datalinks in Kernel Zones"

Note - To configure and manage virtual networks and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 34.

Configuring the Components of a Virtual Network

In Oracle Solaris, VNICs and etherstubs are the basic components of a virtual network. This section describes the steps to configure these components in preparation for building the virtual network.

When configuring VNICs, note the following:

- Certain property modifications work only with VNICs. For example, with the `dladm create-vnic` command, you can configure a MAC address as well as assign a VLAN ID to create a VNIC as a VLAN. However, you cannot configure a MAC address directly for a VLAN by using the `dladm create-vlan` command.
- You can create only one VNIC at a time over a datalink. Like datalinks, VNICs have link properties that you can further configure as needed.

A virtual network device in a Oracle VM Server for SPARC can support multiple Oracle Solaris 11 VNICs. The virtual network device must be configured to support multiple MAC addresses, one for each VNIC that the virtual network device supports. Oracle Solaris zones in the logical domain connect to the VNICs. For more information, see "Using Virtual NICs on Virtual Networks" in *Oracle VM Server for SPARC 3.6 Administration Guide*.

In Oracle VM Server for SPARC, the trusted virtual network feature enables you to dynamically create virtual devices such as VNICs and VLANs on top of virtual network devices (`vnets`). For more information, see "Using Trusted Virtual Networks" in *Oracle VM Server for SPARC 3.6 Administration Guide*.

How to Configure VNICs and Etherstubs

The VNIC connects the virtual network to the external network. Zones communicate through their VNIC's virtual switch. You must create a VNIC for every zone in the virtual network.

This task includes a step to provide names to the components you are creating. For guidelines on assigning names, see "Rules for Valid Link Names" in *Configuring and Managing Network Components in Oracle Solaris 11.4*.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **(Optional) Create an etherstub.**

   ```
   $ dladm create-etherstub etherstub-name
   ```

   Perform this step only if you are creating an isolated private virtual network. See also "Use Case: Configuring a Private Virtual Network" on page 53.

2. **Create a VNIC.**

   ```
   $ dladm create-vnic -l link VNIC
   ```

   link — The name of the link over which the VNIC is configured. In the case of a private virtual network, you specify the name of the etherstub.

   VNIC — The name of the VNIC.

3. **Create an IP interface over the VNIC.**

   ```
   $ ipadm create-ip interface
   ```

interface refers to the VNIC you just created.

4. **Assign an IP address to the VNIC interface.**

```
$ ipadm create-addr -a address  interface
```

5. **(Optional) Verify the VNIC that has been created.**

```
$ dladm show-link
```

Example 1 Configuring a VNIC

This example shows how to configure `vnic1` over the datalink `net0`.

```
$ dladm create-vnic -l net0 vnic1
$ ipadm create-ip vnic1
$ ipadm create-addr -a 192.0.2.10/24 vnic1
$ dladm show-link
LINK          CLASS     MTU    STATE    OVER
net0          phys      1500   up       --
vnic1         vnic      1500   up       net0
```

Example 2 Configuring VNICs Over an Etherstub

This example shows how you can create an etherstub `etherstub0` and configure VNICs `vnic1` and `vnic2` over the etherstub.

```
$ dladm create-etherstub etherstub0
$ dladm create-vnic -l etherstub0 vnic1
$ dladm create-vnic -l etherstub0 vnic2
$ ipadm create-ip vnic1
$ ipadm create-addr -a 192.0.2.20/24 vnic1
$ ipadm create-ip vnic2
$ ipadm create-addr -a 192.0.2.30/24 vnic2
$ dladm show-etherstub -o all
LINK          ZONE
etherstub0    global
$ dladm show-link
LINK          CLASS     MTU      STATE       OVER
net0          phys      1500     up          --
etherstub0    etherstub 9000     unknown     --
vnic1         vnic      9000     up          etherstub0
vnic2         vnic      9000     up          etherstub0
```

How to Configure VNICs as VLANs

You can configure VNICs with VLAN IDs to host VLAN traffic. As part of this configuration, you also set the link property `vlan-announce` to propagate the VLAN configurations of each individual VNIC to the network.

Unlike a regular VLAN link, the VNIC configured as a VLAN has its own MAC address. For information about regular VLANs, see Chapter 4, "Configuring Virtual Networks by Using Virtual Local Area Networks" in *Managing Network Datalinks in Oracle Solaris 11.4*.

This procedure contains only the steps to create the VNIC with a VLAN ID and to set the appropriate properties that enable the VNIC to service VLAN traffic. The intermediary ports and switches are automatically updated when you enable the `vlan-announce` property. However, you must still configure these intermediary ports and switches must be separately to define VLANs at these points.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Create a VNIC with a VLAN ID.**

   ```
   $ dladm create-vnic -l link -v vid VNIC
   ```

   vid refers to the VLAN ID of the VNIC.

2. **(Optional) Broadcast the VNIC's VLAN configuration to the network.**

   ```
   $ dladm set-linkprop -p vlan-announce=gvrp link
   ```

   This step enables a GARP VLAN Registration Protocol (GVRP) client system that automatically registers VLAN IDs with attached switches. By default, the `vlan-announce` property is set to `off` and no VLAN broadcast messages are sent to the network. After you set the property to `gvrp`, the VLAN configuration for that link is propagated to enable automatic VLAN port configuration of the network devices. VLAN traffic can then be accepted and forwarded by these devices. For more information about GVRP, see "*Configuring GVRP* in *Sun Ethernet Fabric Operating System, VLAN Administration Guide*.

3. **(Optional) Set the `gvrp-timeout` property to configure the wait period between VLAN broadcasts.**

   ```
   $ dladm set-linkprop -p gvrp-timeout=time link
   ```

   The default value is 250 milliseconds. A system with a heavy load might require a shorter interval when rebroadcasting VLAN information.

4. **(Optional) Display the value of the properties `vlan-announce` and `gvrp-timeout`.**

```
$ dladm show-linkprop -p vlan-announce,gvrp-timeout
```

Example 3 Configuring a VNIC as a VLAN

This example shows how to create a VNIC named `vnic0` on the datalink `net0` with a VLAN ID `123` and how to enable the VLAN configuration to be announced to the network.

```
$ dladm create-vnic -l net0 -v 123 vnic0
$ dladm set-linkprop -p vlan-announce=gvrp net0
$ dladm set-linkprop -p gvrp-timeout=250 net0
$ dladm show-linkprop -p vlan-announce,gvrp-timeout net0
LINK     PROPERTY        PERM    VALUE     EFFECTIVE    DEFAULT   POSSIBLE
net0     vlan-announce   rw      gvrp      gvrp         off       off,gvrp
net0     gvrp-timeout    rw      250       250          250       100-100000
```

The output shows the following information:

`LINK`	Physical datalink, identified by a name.
`PROPERTY`	Property of the link. A link can have several properties.
`PERM`	Permissions of the property, which can be either `read-only` or `read-write`.
`VALUE`	Current (or persistent) link property value. If the value is not set, it is shown as `--`. If it is unknown, the value is shown as `?`.
`DEFAULT`	Default value of the link property. If the link property has no default value, `--` is shown.
`POSSIBLE`	A comma-separated list of the values that the link property can have. If the possible values are unknown or unbounded, `--` is shown.

▼ How to Configure VNICs as PVLANs

You can configure VNICs with primary and secondary VLAN IDs of a PVLAN to host the PVLAN traffic. For more information about PVLANs, see Chapter 4, “Configuring Virtual Networks by Using Virtual Local Area Networks” in *Managing Network Datalinks in Oracle Solaris 11.4*.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See “Using Rights Profiles to Perform Network Configuration” on page 34.

1. **Create a PVLAN VNIC by specifying the primary VLAN ID and secondary VLAN ID.**

   ```
   $ dladm create-vnic -l link -v VLAN-ID,PVLAN-SVID,PVLAN-type VNIC
   ```

link	Specifies the Ethernet link over which the VLAN is created.
VLAN-ID	Primary ID associated with a VLAN.
PVLAN-SVID	Secondary VLAN ID associated with the PVLAN.
PVLAN-type	The PVLAN type associated with the VLAN, which can be either `isolated` or `community`. The default value is `isolated`.
VNIC	Name of the VNIC.

2. **(Optional) Display the PVLAN or PVLAN VNIC that is created.**

   ```
   $ dladm show-vnic -v
   ```

Example 4 Creating a PVLAN VNIC

The following example shows how to create a PVLAN with the primary VLAN ID as `4`, secondary VLAN ID as `110`, and PVLAN type as `isolated`.

```
$ dladm create-vnic -v 4,110,community -l net1 vnic2
$ dladm show-vnic -v
LINK        VID  SVID PVLAN-TYPE  OVER
vnic2       4    110  community   net1
```

Configuring IPoIB VNICs

IP over InfiniBand (IPoIB) devices enable transporting IP packets over IB connections. You configure IPoIB VNICs by specifying the partition key. Although you can migrate IPoIB VNICs from one underlying datalink to another underlying datalink, IPoIB partition links do not support migration. For more information, see "About InfiniBand Devices" in *Managing Devices in Oracle Solaris 11.4*.

▼ How to Configure IPoIB VNICs

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **(Optional) Check the information about the IB physical link over which you want to create the IPoIB datalink.**

```
$ dladm show-ib link
```

The output shows the datalinks and information about their partition keys.

2. **Create an IPoIB VNIC by specifying the partition key.**

```
$ dladm create-vnic [-f] -l link -P pkey -p [prop=value] VNIC
```

`-f` — Optional. Forces the creation of the IPoIB VNIC even if the partition key on the port or the multicast group is absent, or the port is down.

`-P` *pkey* — The partition key that needs to be used. This option is mandatory for IPoIB VNICs and not applicable for other types of datalinks. When you specify the partition key, it is always considered a hexadecimal, regardless of whether it has the prefix `0x`.

`-p` *prop=value* — Used to specify the value of the `linkmode` property of the IPoIB VNIC, which enables you to set the link transport service type on an IB partition datalink. You can set the following values for the `linkmode` property:

- `cm` - connected mode. This mode uses a default maximum transmission unit (MTU) of 65520 bytes and supports a maximum MTU of 65535 bytes. If connected mode is not available for a remote node, unreliable datagram mode is automatically used. `cm` is the default value.
- `ud` - unreliable datagram mode. This mode uses a default MTU of 2044 bytes and supports a maximum MTU of 4092 bytes.

3. **(Optional) Display the IPoIB VNIC that is created.**

```
$ dladm show-vnic
```

Note - To display IPoIB datalinks created with the `dladm create-part` command, use `dladm show-part` instead.

---

4. **Plumb and assign an IP address to an IPoIB VNIC.**

```
$ ipadm create-ip interface
$ ipadm create-addr -a address interface [address-object]
```

interface — The name of the IPoIB VNIC that you created.

-a *address*	Specifies the IP address.
address-object	A name that identifies the IP address in association with the IP interface. If *address-object* is not specified, the OS automatically assigns a name by using the format *IP-name/protocol*.

Example 5 Creating IPoIB VNICs

The following example shows how to create an IPoIB VNIC over the datalink `net4` by using the partition key `0xffff`.

```
$ dladm show-ib net4
LINK      HCAGUID        PORTGUID       PORT STATE GWNAME GWPORT  PKEYS
net4      21280001A0A58C 21280001A0A58D 1    up      --     --    FFFF
$ dladm create-vnic -l net4 -P 0xffff ipoib_vnic0
$ dladm show-vnic
LINK         OVER    SPEED  MACADDRESS        MACADDRTYPE IDS
eth_vnic0    net0    1000   2:8:20:ef:d2:77   random      VID:0
ipoib_vnic0  net4    32000  80:0:0:4a:fe:..   fixed       PKEY:0xFFFF
$ ipadm create-ip ipoib_vnic0
$ ipadm create-addr -a 192.0.2.10 ipoib_vnic0/v4
```

To display information about selected properties only, use the `-o link,`[*properties*[option. Separate multiple properties with a comma.

Identifying the Link Status of an IPoIB VNIC

IPoIB VNICs can be down in any of following scenarios:

- HCA port is down
- Partition key is not present
- IPoIB VNIC fails to join the broadcast multicast group

You can use the `dladm show-ib` command to check the HCA port status and the configured partition key. To find out if the IPoIB VNIC has failed to join the broadcast multicast group, you can use the `dladm show-linkprop` command to display the `broadcast-group` datalink property of the IPoIB VNIC. The valid values of this read-only property are `unknown`, `joined`, `assigned`, and `unsuccessful`, and the default value is `unknown`.

EXAMPLE 6 Identifying the Link Status of an IPoIB VNIC

This example shows how to find the link state of an IPoIB VNIC.

```
$ dladm show-ib net1
LINK      HCAGUID        PORTGUID       PORT STATE GWNAME GWPORT  PKEYS
net4      21290001A0A58C 21280001A0A59D 1    up      --     --    FFFF
$ dladm create-vnic -l net4 -P 0xffff ip_vnic0
$ dladm show-vnic
LINK         OVER    SPEED  MACADDRESS        MACADDRTYPE IDS
eth_vnic0    net0    1000   2:8:20:ef:d2:77   random      VID:0
ip_vnic0     net1    32000  80:0:0:4a:fe:..   fixed       PKEY:0xFFFF
$ dladm show-linkprop -p broadcast-group vnic0
LINK      PROPERTY        PERM VALUE   EFFECTIVE  DEFAULT   POSSIBLE
ip_vnic0  broadcast-group r-   joined  joined     unknown   unknown
                                                            absent,
                                                            joined,
                                                            unsuccessful
```

The `broadcast-group` property can have the following values:

`unknown`	Initial state of the datalink after it is created and before an IP address is configured over it.
`absent`	Broadcast group is not created on the subnet manager (SM).
`joined`	Broadcast group is configured and IPoIB has successfully joined the broadcast group.
`unsuccessful`	Broadcast group is configured but IPoIB has failed to join the broadcast group. This can occur if one or more parameters such as MTU, state, and key are different from broadcast group created on the subnet manager. See SM log to learn the exact reason for the failure of the link to join the broadcast group.

The value of the `broadcast-group` property is displayed as `joined` when the VNIC is created.

The `broadcast-group` property displays the value `unsuccessful` when the broadcast group is configured but IPoIB failed to join the broadcast group.

For information about IPoIB devices, see "Administering IPoIB Devices" in *Managing Devices in Oracle Solaris 11.4*.

Managing IPoIB VNICs

You migrate and delete IPoIB VNICs just as you would with Ethernet VNICs. For more information, see "Migrating VNICs" on page 65 and "Deleting VNICs" on page 67.

The following example shows how to migrate `ipoib_vnic0` to the datalink `net5`.

```
$ dladm modify-vnic -l net5 ipoib_vnic0
```

The following example shows how to delete `ipoib_vnic0`.

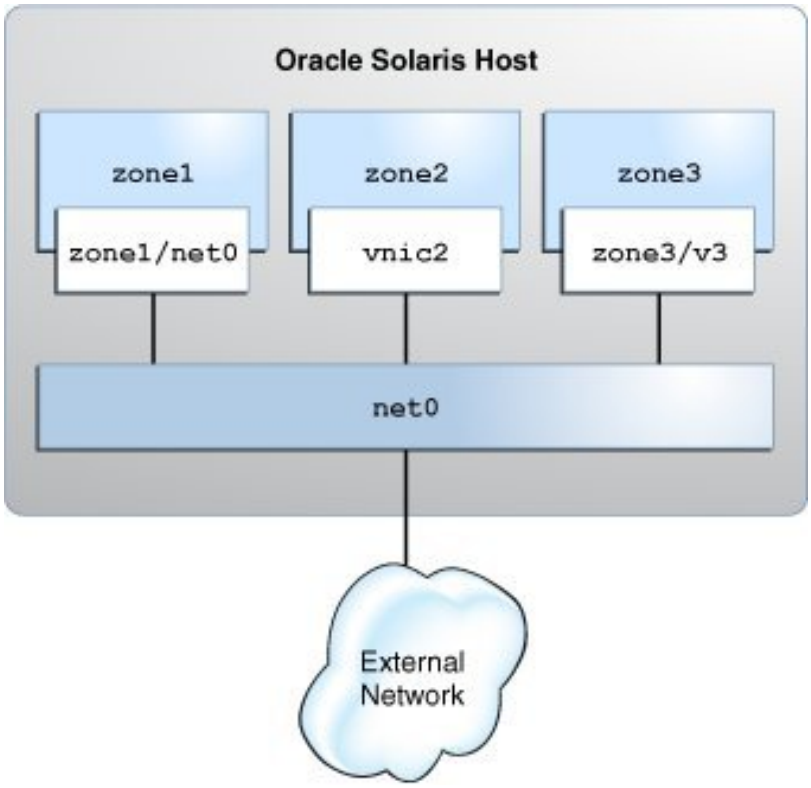
```
$ dladm delete-vnic ipoib_vnic0
```

Building Virtual Networks

You can configure more than one VNIC on a physical device. Thus, you can create a multinode network within a single system. The network would consist of virtual servers (zones) that are connected in a network within a single operating system instance.

The following figure shows an example of a virtual network setup in an Oracle Solaris host.

FIGURE 5 Virtual Network Setup



The procedures in this section are based on the following assumptions:

- The virtual network on the system consists of three zones. The procedures in this section are based on the following zone configurations:
  - The first zone `zone1` is created as a new zone with an `anet` resource. For information, see "How to Configure a Zone for the Virtual Network" on page 48.
  - The second zone `zone2` already exists on the system and needs to be reconfigured to use a VNIC. For information, see "How to Reconfigure a Zone to Use a VNIC" on page 50.
  - The third zone `zone3` already exists on the system. You need to temporarily create the VNIC `zone3/v3` in `zone3` from the global zone. For information, see "Creating VNICs Directly for Zones" on page 52.

- The system's physical interface is configured with the IP address `192.0.2.20`.
- The router's IP address is `192.0.2.25`.

When building the virtual network, some steps are performed in a global zone and some are performed in a non-global zone. For clarity, the prompts in the examples after each step indicate in which zone a specific command is issued. However, the actual path that the prompts display might vary depending on the prompts specified for your system.

For a demonstration of configuring a virtual network, see Configuring a Virtual Network in Oracle Solaris - Part 1 (`https://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part1/VirtualDemo_Part1.htm`) and Configuring a Virtual Network in Oracle Solaris - Part 2 (`https://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part2/VirtualDemo_Part2.htm`).

How to Configure a Zone for the Virtual Network

This procedure explains how to configure a new zone with the VNIC `anet` resource. Note that only the steps related to network virtualization are included in the procedure. For more information about how to configure zones, see *Creating and Using Oracle Solaris Zones*.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Create the zone.**

```
global$ zonecfg -z zone
zonecfg:zone> create -t SYSsolaris
```

When you create a zone, a VNIC `anet` resource is added to the zone by default. The lower link for the VNIC `anet` resource is selected automatically. You can manually set the lower link for the VNIC `anet` resource as described in the next step.

2. **Select the default VNIC `anet` resource of the zone and set the lower link.**

```
zonecfg:zone> select anet linkname=net0
zonecfg:zone:anet> set lower-link=NIC
```

3. **Configure the IP address and the default router for the `anet` resource of the zone.**

```
zonecfg:zone:anet> set allowed-address=IP-address-of-the-anet-resource
zonecfg:zone:anet> set defrouter=IP-address-of-the-default-router
```

```
zonecfg:zone:anet> end
```

4. **Verify and commit the changes that you have implemented and then exit the zone.**

```
zonecfg:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

5. **Install and boot the zone.**

```
global$ zoneadm -z zone install
global$ zoneadm -z zone boot
```

6. **Log in to the zone and complete the zone configuration.**

```
global$ zlogin -C zone
```

During the zone configuration, you can specify most of the information by selecting from a list of choices. Usually, the default options suffice. You can skip the network configuration because you have already set the `allowed-address` and `defrouter` properties for the `anet` resource.

Example 7 Configuring a Zone for the Virtual Network

In this example, `zone1` is created for the virtual network with the VNIC `anet` resource. Note that only the zone parameters that are relevant to the creation of a virtual network are listed.

```
global $ zonecfg -z zone1
Use 'create' to begin configuring a new zone.
zonecfg:zone1> create -t SYSsolaris
zonecfg:zone1> select anet linkname=net0
zonecfg:zone1:anet> set lower-link=net0
zonecfg:zone1:anet> set allowed-address=192.0.2.10/24
zonecfg:zone1:anet> set defrouter=192.0.2.1
zonecfg:zone1:anet> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
global$ zoneadm -z zone1 install
.
.
.
global$ zoneadm -z zone1 boot

global$ zlogin -C zone1
```

Specify the information for the zone as you are prompted.

How to Reconfigure a Zone to Use a VNIC

This procedure assumes the case of an existing zone whose current configuration prevents it from becoming a part of the virtual network. To join the network, its current interface needs to be changed.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Create the VNIC.**

   ```
   global$ dladm create-vnic -l link VNIC
   ```

   You will configure the VNIC's interface later in this procedure.

2. **Change the zone's interface to use a VNIC.**

   ```
   zonecfg:zone> remove net physical=NIC
   zonecfg:zone> add net
   zonecfg:zone:net> set physical=VNIC
   zonecfg:zone:net> end
   ```

3. **Verify and commit the changes that you have implemented and then exit the zone.**

   ```
   zonecfg:zone> verify
   zonecfg:zone> commit
   zonecfg:zone> exit
   ```

4. **Reboot the zone.**

   ```
   global$ zoneadm -z zone reboot
   ```

5. **Log in to the zone.**

   ```
   global$ zlogin zone
   ```

6. **In the zone, create an IP interface over the VNIC that is now assigned to the zone.**

   ```
   zone$ ipadm create-ip interface
   ```

7. **Configure the VNIC with a static IP address or a Dynamic Host Configuration Protocol (DHCP) IP address.**

- **Assign a static IP address.**

  zone$ **ipadm create-addr -a** *address interface*

  -a *address* Specifies the IP address, which can be in CIDR notation.

- **Assign a DHCP IP address.**

  zone$ **ipadm create-addr -T dhcp** *interface*

8. **Exit the zone.**

   zone$ **exit**

9. **From the global zone, add the address information to the `/etc/hosts` file.**

Example 8 Reconfiguring a Zone to Use a VNIC

In this example, the zone uses the primary interface of the system rather than a virtual link. You need to modify `zone2` to use `vnic2`. Note that some of the output is truncated to focus on the relevant information that relates to virtual networks.

```
global$ dladm create-vnic -l net0 vnic2

global$ zonecfg -z zone2
zonecfg:zone2> remove net physical=net0
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vnic2
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global$ zoneadm -z zone2 reboot

global$ zlogin zone2
zone2$ ipadm create-ip vnic2
zone2$ ipadm create-addr -a 192.0.2.85/24 vnic2
ipadm: vnic2/v4

zone2$ exit

global$ pfedit /etc/hosts
#
::1             localhost
127.0.0.1       localhost
192.0.2.20      loghost   #For net0
```

```
192.0.2.80      zone1   #using vnic1
192.0.2.85      zone2   #using vnic2
```

Creating VNICs Directly for Zones

VNICs can be created directly in a non-global zone from a global zone by specifying the link as *zone*/*link*. This method creates the VNIC directly in the namespace of the non-global zone. Make sure that the non-global zone exists and is running. Then use the following command syntax:

```
global$ dladm create-vnic -l link zone/vnic
```

To create a temporary VNIC, use the `-t` option. The VNIC will persist until the next reboot of the zone.

You can view VNIC information in two ways. From the global zone, you would use `dladm show-link -Z`. From within the non-global zone, you use the same command without the `-Z` option.

In addition to temporarily creating VNICs, you can also temporarily create VLANs and IP over InfiniBand (IPoIB) partitions. See the `dladm`(8) man page for complete instructions.

In the following example, `vnic1` is created for `zone1` from the global zone.

```
global$ dladm create-vnic -l net0 zone1/vnic1
global$ dladm show-link -Z
LINK                ZONE      CLASS     MTU    STATE    OVER
net0                global    phys      1500   up        --
zone1/vnic1         zone1     vnic      1500   down     net0
```

From `zone1`, you would display the link information as follows:

```
zone1$ dladm show-link
LINK                CLASS     MTU    STATE    OVER
vnic1               vnic      1500   down     ?
```

In the following example, a temporary VNIC as a VLAN is created for `zone3` from the global zone.

```
global$ dladm create-vlan -t -l net0 -v 3 zone1/vlan3
```

In he following example, an IPoIB partition `part1` with a temporary VNIC is created in `zone1` from the global zone.

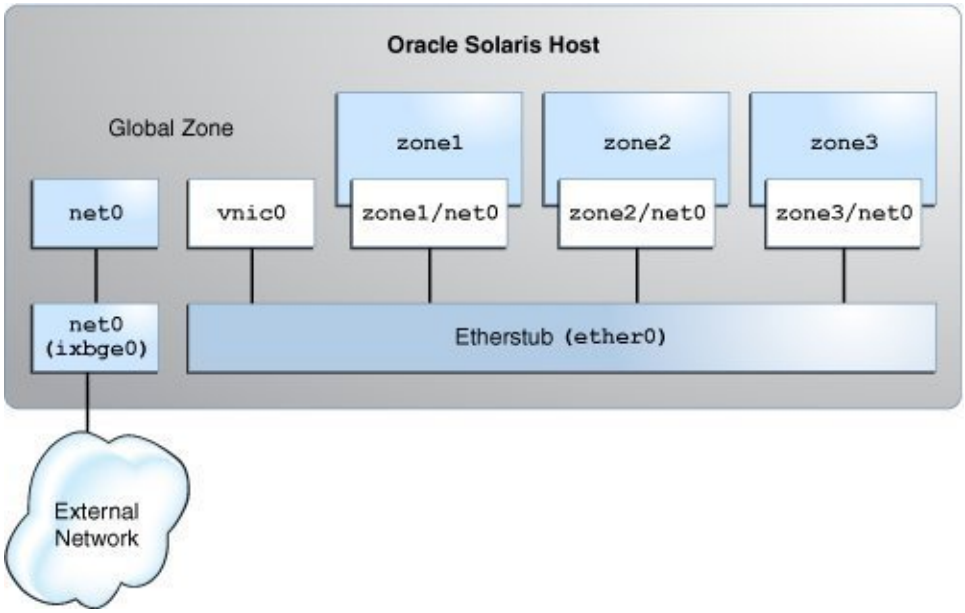
```
global$ dladm create-part -t -l net1 -P FFFF zone1/part1
```

The `-P` option specifies the partition key that is used for creating a partition link.

Use Case: Configuring a Private Virtual Network

This use case shows how to create a private virtual network and enable it to send network traffic outside the system. The scenario is based on the configuration in the following figure:

FIGURE 6 Use Case: Private Virtual Network Setup



The configuration on the Oracle Solaris host is as follows:

- The primary interface of the global zone is `net0`.
- The zones, `zone1`, `zone2`, and `zone3`, are configured with the VNIC `anet` resources and the etherstub `ether0` is set as the lower link for the zones.
- The VNIC `vnic0` is configured over the etherstub `ether0`.

This use case is based on the following assumptions:

- The primary interface, `net0`, is configured for the system with the IP address `192.0.2.20 /27` and the default router IP address `192.0.2.1/27`.
- The first zone, `zone1`, is created as a new zone. The second zone `zone2` already exists on the system and needs to be reconfigured with an `anet` resource.
- The third zone, `zone3`, is reconfigured by using Live Zone Reconfiguration. For examples, see "How to Modify Zone Resources and Properties" in *Creating and Using Oracle Solaris Zones*.

The following table shows the IP addresses that are configured for the zones and their respective default routers in the private virtual network setup.

TABLE 2 IP Addresses Configured for the Zones in the Private Virtual Network Setup

Zone	IP Address of the `anet` Resource	IP Address of the Default Router
`zone1`	`192.0.2.34/27`	`192.0.2.33/27`
`zone2`	`192.0.2.35/27`	`192.0.2.33/27`
`zone3`	`192.0.2.36/27`	`192.0.2.33/27`

The tasks that follow show how to configure this network.

How to Configure the Private Virtual Network

The global zone performs routing and NAT, so you need to connect the global zone to both the private virtual network and the physical NIC. You connect the global zone to the physical NIC by configuring the primary interface in the global zone. You connect the global zone to the private virtual network by creating `vnic0` over the etherstub `ether0`.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Create the etherstub `ether0`.**

```
$ dladm create-etherstub ether0
```

2. **Create the VNIC `vnic0` over `ether0` with `192.0.2.33/27` as its IP address.**

```
$ dladm create-vnic -l ether0 vnic0
$ ipadm create-ip vnic0
$ ipadm create-addr -a 192.0.2.33/27 vnic0
```

The VNIC `vnic0` acts as the default router for the zones.

3. **Create `zone1` with the VNIC `anet` resource and set `ether0` as the lower link.**

   Configure `zone1` with the IP addresses that are displayed in Table 2, “IP Addresses Configured for the Zones in the Private Virtual Network Setup,” on page 54.

```
global$ zonecfg -z zone1
Use 'create' to begin configuring a new zone.
zonecfg:zone1> create -t SYSsolaris
zonecfg:zone1> select anet linkname=net0
zonecfg:zone1:anet> set lower-link=ether0
zonecfg:zone1:anet> set allowed-address=192.0.2.34/27
zonecfg:zone1:anet> set defrouter=192.0.2.33/27
zonecfg:zone1:anet> end
zonecfg:zone1> commit
zonecfg:zone1> exit
```

4. **Install and boot `zone1`.**

```
global$ zoneadm -z zone1 install
global$ zoneadm -z zone1 boot
```

5. **Log in to `zone1` and complete the zone configuration.**

```
global$ zlogin -C zone1
```

6. **Reconfigure `zone2` with an `anet` resource and set `ether0` as the lower link.**

   Configure `zone2` with the IP addresses that are displayed in Table 2, “IP Addresses Configured for the Zones in the Private Virtual Network Setup,” on page 54.

```
global$ zonecfg -z zone2
zonecfg:zone2> select anet linkname=net0
zonecfg:zone2:anet> set lower-link=ether0
zonecfg:zone2:anet> set allowed-address=192.0.2.35/27
zonecfg:zone2:anet> set defrouter=192.0.2.33/27
zonecfg:zone2:anet> end
zonecfg:zone2> commit
zonecfg:zone2> exit
```

7. **Reboot and log in to the zone `zone2`.**

```
global$ zoneadm -z zone2 reboot
global$ zlogin zone2
```

8. **Reconfigure `zone3` and set `ether0` as the lower link.**

```
global$ zonecfg -z zone3
zonecfg:zone3> select anet linkname=net0
zonecfg:zone3:anet> set lower-link=ether0
```

```
zonecfg:zone3:anet> end
zonecfg:zone3> commit
zonecfg:zone3> exit
```

9. **Configure the IP address and default gateway for `zone3`.**

   Configure `zone3` with the IP addresses that are displayed in Table 2, "IP Addresses Configured for the Zones in the Private Virtual Network Setup," on page 54.

```
global$ zoneadm -z zone3 apply
global$ zlogin zone3
zone3$ ipadm create-ip net0
zone3$ ipadm create-addr -a 192.0.2.36/27 net0/v4
zone3$ route -p add default 192.0.2.33/27
```

How to Enable IP Forwarding and NAT

You can enable the private virtual network to send network traffic outside the system by enabling IP forwarding and network address translation (NAT) in the global zone.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Enable IP forwarding in the global zone.**

```
global$ ipadm set-ifprop -p forwarding=on -m ipv4 net0
global$ ipadm set-ifprop -p forwarding=on -m ipv4 vnic0
```

2. **From the global zone, configure NAT in the `/etc/firewall/pf.conf` file for the primary interface.**

```
global$ cat /etc/firewall/pf.conf
map net0 192.0.2.0/2 -> 0/32  portmap tcp/udp auto
map net0 192.0.2.0/27 -> 0/32
```

   For information about packet filter, see Chapter 4, "Oracle Solaris Firewall" in *Securing the Network in Oracle Solaris 11.4*.

3. **Start the firewall service to enable NAT.**

```
global$ svcadm enable network/firewall
```

4. **(Optional) Check whether you can send the network traffic outside the system by pinging the default router of the system from any of the zones.**

```
$ ping 192.0.2.1/27
```

Managing VNICs

This section describes tasks that you can perform on VNICs after performing basic configuration. For information about how to perform basic configuration of VNICs, see "How to Configure VNICs and Etherstubs" on page 38.

You can modify the VLAN ID, the MAC address, and the underlying datalink of a VNIC. Modifying the underlying datalink means moving a VNIC to another datalink. You can either globally modify the attribute of all the VNICs on a datalink or selectively modify the attribute of only specified VNICs.

Displaying VNICs

To obtain information about the VNICs on your system, use the `dladm show-vnic` command.

EXAMPLE 9 Displaying VNICs on a System

```
$ dladm show-vnic
LINK      OVER     SPEED      MACADDRESS          MACADDRTYPE       IDS
vnic1     net0     1000       2:8:20:c2:39:38     random            VID:123
vnic2     net0     1000       2:8:20:5f:84:ff     random            VID:456
```

The output shows the following information:

`LINK` Virtual datalink, identified by a name.

`OVER` Physical or virtual datalink over which the VNIC is configured.

`SPEED` Maximum speed of the VNIC, in megabits per second.

`MACADDRESS` MAC address of the VNIC.

`MACADDRTYPE` MAC address type of the VNIC, which can be one of the following:

- `random` – The random address assigned to the VNIC
- `factory` – The factory MAC address of the NIC used by the VNIC
- `fixed` – The MAC address assigned by the user

`VID` VLAN ID of the VNIC.

You can use any `dladm` command that shows information about datalinks to include information about VNICs if they exist on the system. For example, the `dladm show-link` command displays

VNICs with other datalinks. You can use the `dladm show-linkprop` command to display the properties of VNICs.

To obtain information about the datalink property of a single VNIC, specify the VNIC in the following command syntax:

```
$ dladm show-linkprop [-p property] vnic
```

EXAMPLE 10 Displaying VNICs That Are Attached to Zones

In this example, information is displayed for the primary datalink and VNICs that are attached to the zones. The primary datalink `net0` is attached to the global zone. The VNICs, `vnic1` and `vnic2`, are attached to `zone1` and `zone2` respectively.

```
$ dladm show-link -Z
LINK                ZONE      CLASS     MTU    STATE    OVER
net0                global    phys      1500   up        --
zone1/vnic1         zone1     vnic      1500   up       net0
zone2/vnic2         zone2     vnic      1500   up       net0
```

Displaying VNICs With Multiple MAC Addresses

Multiple MAC addresses are associated with system-created VNICs in Oracle VM Server for SPARC and the `anet` resources in Oracle Solaris Kernel Zones. In Oracle VM Server for SPARC, you need to create a `vnet` with the `alt-mac-addrs` property to support VNICs and zones inside a guest domain. In this case, the system automatically creates a VNIC with multiple MAC addresses. These multiple MAC addresses are obtained from the `vnet` that you created. For more information, see *Oracle VM Server for SPARC 3.6 Administration Guide*.

To support zones or VNICs inside kernel zones, you configure the `anet` resources with multiple MAC addresses. You use the `zonecfg` command to specify multiple MAC addresses to the `anet` resources created for network access in kernel zones. For more information, see the `solaris-kz`(7) man page. For information about configuring kernel zones, see *Creating and Using Oracle Solaris Kernel Zones*.

When multiple MAC addresses are associated with VNICs, one MAC address is used by the virtual network driver. You can use the remaining MAC addresses to create VNICs inside kernel zones or a guest domain. For example, if a VNIC is associated with three MAC addresses, one MAC address is assigned for the virtual network driver. Hence, you can create only two VNICs with the remaining two MAC addresses.

You can use the following command to display multiple MAC addresses associated with VNICs:

```
$ dladm show-vnic -m
```

EXAMPLE 11 Displaying VNICs With Multiple MAC Addresses in Kernel Zones

```
$ dladm show-vnic -m
LINK                OVER       MACADDRESSES      MACADDRTYPES     IDS
gz_vnic0            net0       2:8:20:d7:27:9d   random           VID:0
zone1/net0          net0       2:8:20:70:52:9    random           VID:0
                               2:8:20:c9:d:4c    fixed
                               2:8:20:70:db:3    random
zone1/net1          net0       0:1:2:3:4:5       fixed            VID:0
                               0:1:2:3:4:6       fixed
```

In this example, kernel zone `zone1` has two `anet` resources: `net0` and `net1`. Both resources have more than one MAC address configured. Therefore, inside kernel zone `zone1`, you can create up to two VNICs on top of the virtual NIC driver `zvnet` associated with datalink `net0`. You can create only one VNIC on top of the virtual NIC driver `zvnet` associated with datalink `net1`.

EXAMPLE 12 Displaying System-Created VNICs With Multiple MAC Addresses

```
$ dladm show-vnic -m
LINK                OVER       MACADDRESSES      MACADDRTYPES     IDS
ldoms-vsw0.vport0   net1       0:14:4f:fb:e1:8f  fixed            VID:0,21
                               0:14:4f:f8:6b:9   fixed
                               0:14:4f:fa:48:7f  fixed
ldoms-vsw0.vport1   net1       0:14:4f:f9:1b:8d  fixed            VID:45,44
                               0:14:4f:f9:27:4   fixed
```

In this example, you can create up to two VNICs on top of the guest domain's virtual network driver `vnet` associated with `ldoms-vsw0.vport0`. You can create up to one VNIC on top of the virtual NIC driver `vnet` associated with `ldoms-vsw0.vport1`.

Displaying the Physical and Virtual Link State of Datalinks

The physical link state of a datalink identifies whether the physical device has connectivity with the external network. If the cable is plugged in and the state of the port on the other end of the cable is `up`, then the physical device has connectivity with the external network.

You can use the following commands to display the physical link state of a datalink:

```
$ dladm show-phys [link]

$ dladm show-ether [link]
```

For more information, see the dladm(8) man page.

EXAMPLE 13 Displaying the Physical Link State of Datalinks

The following example displays the physical link state of datalinks on a system by using the `dladm show-phys` command.

```
$ dladm show-phys
LINK        MEDIA        STATE      SPEED  DUPLEX    DEVICE
net1        Ethernet     down       0      unknown   e1000g1
net2        Ethernet     down       0      unknown   e1000g2
net3        Ethernet     down       0      unknown   e1000g3
net0        Ethernet     up         1000   full      e1000g0
```

The following example displays the physical link state of datalinks on a system by using the `dladm show-ether` command.

```
$ dladm show-ether
LINK        PTYPE    STATE    AUTO  SPEED-DUPLEX    PAUSE
net1        current  down     yes   0M              bi
net2        current  down     yes   0M              bi
net3        current  down     yes   0M              bi
net0        current  up       yes   1G-f            bi
```

When multiple VNICs are created over a NIC, a virtual switch is created internally to enable VNICs and the primary datalink to communicate when they are on the same VLAN. These datalinks can communicate with each other even if the physical datalink has no connection with the external network. This forms the virtual link state of the datalink, which can be `up`, `down`, or `unknown`. The virtual link state of a datalink identifies whether a datalink has connectivity with internal networks within the system even if the physical cable is unplugged.

You use the following command to display the virtual link state of a datalink:

```
$ dladm show-link [link]
```

EXAMPLE 14 Displaying the Virtual Link State of Datalinks

This example displays the virtual link state of datalinks on a system.

```
$ dladm show-link
LINK        CLASS     MTU    STATE    OVER
net0        phys      1500   up       --
net2        phys      1500   down     --
net4        phys      1500   down     --
net1        phys      1500   up       --
net5        phys      1500   up       --
vnic0       vnic      1500   up       net5
```

```
vnic1       vnic      1500   up       net5
vnic2       vnic      1500   up       net1
```

Modifying the VLAN IDs of VNICs

VNICs can be configured as VLANs. You need to modify the VLAN IDs of VNICs on a datalink when you want the VNICs to host a specific VLAN's traffic.

The `dladm` subcommand that you use depends on whether you are modifying VLANs or VNICs configured as VLANs:

- For VLANs that are created with the `dladm create-vlan` command, use the `dladm modify-vlan` command. To display these VLANs, use the `dladm show-vlan` command.
- For VLANs that are created with the `dladm create-vnic` command, use the `dladm modify-vnic` command. To display these VNICs, including those with VLAN IDs, use the `dladm show-vnic` command.

You can modify the VLAN ID of a single VNIC or multiple VNICs that are configured on the datalink. You can also modify the VLAN IDs of VNICs as a group by configuring all the VNICs with the same VLAN ID.

If only one VNIC is configured on the datalink, use the following command syntax to modify the VLAN ID of the VNIC:

```
$ dladm modify-vnic -v vid -L link
```

where *vid* is the new VLAN ID that you assign to the VNIC.

EXAMPLE 15 Modifying the VLAN ID of a VNIC on a Datalink

In this example, the VLAN ID of `vnic0` that is configured over the datalink `net0` is modified.

```
$ dladm modify-vnic -v 123 -L net0
$ dladm show-vnic
LINK      OVER     SPEED         MACADDRESS          MACADDRTYPE        IDS
vnic0     net0     1000          2:8:20:c2:39:38     random             VID:123
```

If multiple VNICs are configured on the datalink, use the following command syntax to modify the VLAN IDs of the VNICs:

```
$ dladm modify-vnic -v vid VNIC
```

You must change the VLAN IDs one at a time because each VLAN ID is unique for VNICs on the same datalink.

EXAMPLE 16 Modifying the VLAN ID of Multiple VNICs on a Datalink

In this example, the VLAN IDs of `vnic0`, `vnic1`, and `vnic2` are modified.

```
$ dladm modify-vnic -v 123 vnic0
$ dladm modify-vnic -v 456 vnic1
$ dladm modify-vnic -v 789 vnic2
$ dladm show-vnic
LINK      OVER     SPEED         MACADDRESS         MACADDRTYPE         IDS
vnic0     net0     1000          2:8:20:c2:39:38    random              VID:123
vnic1     net0     1000          2:8:20:5f:84:ff    random              VID:456
vnic2     net0     1000          2:8:20:5f:84:ff    random              VID:789
```

If each VNIC is configured on a different datalink, use the following command syntax to modify the VLAN ID of VNICs as a group:

```
$ dladm modify-vnic -v vid VNIC,VNIC,[...]
```

EXAMPLE 17 Modifying the VLAN IDs of VNICs as a Group

In this example, the VLAN IDs of `vnic0`, `vnic1`, and `vnic2` are modified as a group. These VNICs are configured over the datalinks `net0`, `net1`, and `net2` respectively.

```
$ dladm modify-vnic -v 123 vnic0,vnic1,vnic2
$ dladm show-vnic
LINK      OVER     SPEED         MACADDRESS          MACADDRTYPE         IDS
vnic0     net0     1000          2:8:20:c2:39:38     random              VID:123
vnic1     net1     1000          2:8:20:5f:84:ff     random              VID:123
vnic2     net2     1000          2:8:20:5f:84:ff     random              VID:123
```

Modifying PVLAN VNICs

You can modify the primary and secondary VLAN IDs and the PVLAN type of PVLAN VNICs by using the `dladm modify-vnic` command. The syntax is as follows:

```
$ dladm modify-vnic [-v VLAN-ID[,PVLAN-SVID[,PVLAN-type]]] VNIC
```

EXAMPLE 18 Modifying a PVLAN VNIC

The following example modifies the primary VLAN ID to `5`, the secondary VLAN ID to `102`, and the PVLAN type to `isolated`.

```
$ dladm show-vnic -v
```

```
LINK       VID  SVID  PVLAN-TYPE  OVER
vnic2      4    101   community   net1
$ dladm modify-vnic -v 5,102,isolated vnic2
$ dladm show-vnic -v
LINK       VID  SVID  PVLAN-TYPE  OVER
vnic2      5    102   isolated    net1
```

For information about the PVLANs, see Chapter 4, "Configuring Virtual Networks by Using Virtual Local Area Networks" in *Managing Network Datalinks in Oracle Solaris 11.4*.

Modifying VNIC MAC Addresses

Any VNIC that a user creates can only have one MAC address. You can modify the MAC address by using the `dladm modify-vnic` command. You can configure the VNICs created for kernel zones with one or more MAC addresses.

You can modify the existing MAC address of a VNIC configured on a datalink. You can either modify the MAC addresses of all the VNICs or selectively modify the MAC addresses of the specified VNICs. You can also modify the VLAN ID and the MAC address of a VNIC simultaneously.

To modify the MAC address of a VNIC, use the following command syntax:

```
$ dladm modify-vnic -m MAC-address VNIC
```

where *MAC-address* is the new MAC address that you want to assign to the VNIC.

EXAMPLE 19 Modifying the MAC Address of a VNIC

In this example, `vnic0` is assigned a specific MAC address.

```
$ dladm modify-vnic -m 3:8:20:5f:84:ff vnic0
$ dladm show-vnic
LINK      OVER     SPEED         MACADDRESS         MACADDRTYPE       IDS
vnic0     net0     1000         3:8:20:5f:84:ff     fixed             VID:0
```

To modify the MAC addresses of all the VNICs on a datalink, use the following command syntax:

```
$ dladm modify-vnic -m random -L link
```

In this command syntax, the `-m random` option is equivalent to the `-m auto` option. The MAC address is assigned automatically to the VNICs on a random basis.

EXAMPLE 20 Modifying the MAC Addresses of All the VNICs on a Datalink

In this example, the MAC addresses of all the VNICs configured over the datalink `net0` are automatically modified on a random basis.

```
$ dladm modify-vnic -m random -L net0
$ dladm show-vnic
LINK      OVER     SPEED        MACADDRESS           MACADDRTYPE       IDS
vnic0     net0     1000         2:8:20:22:9d:bb      random            VID:0
vnic1     net0     1000         2:8:20:72:2e:9       random            VID:0
vnic2     net0     1000         2:8:20:2f:e5:83      random            VID:0
```

To modify the MAC addresses of VNICs on a selective basis, use the following command syntax:

```
$ dladm modify-vnic -m random VNIC,VNIC,[...]
```

For both the global and selective modifications, you specify `random` for the `-m` option.

EXAMPLE 21 Modifying the MAC Addresses of VNICs on Selective Basis

In this example, the MAC addresses of `vnic0` and `vnic2` that are configured over the datalink `net0` are selectively modified.

```
$ dladm modify-vnic -m random vnic0,vnic2
$ dladm show-vnic
LINK      OVER     SPEED         MACADDRESS         MACADDRTYPE       IDS
vnic0     net0     1000         2:8:20:2f:e5:83     random            VID:0
vnic1     net0     1000         2:8:20:5f:84:ff     fixed             VID:0
vnic2     net0     1000         2:8:20:2f:e5:83     random            VID:0
```

To modify the VLAN ID and the MAC address of a VNIC simultaneously, use the following command syntax:

```
$ dladm modify-vnic -m random -v vid VNIC
```

Caution - Modifying multiple attributes of the VNICs globally might cause unexpected behavior with the VNICs. Instead, modify the multiple attributes of the VNICs separately.

EXAMPLE 22 Modifying the VLAN ID and the MAC Address of a VNIC

In this example, the VLAN ID and the MAC address of `vnic0` are modified simultaneously.

```
$ dladm modify-vnic -m random -v 123 vnic0
$ dladm show-vnic vnic0
LINK      OVER     SPEED  MACADDRESS        MACADDRTYPE    IDS
```

```
vnic0     net0     1000   2:8:20:2f:e5:83   random         VID:123
```

Migrating VNICs

You can move one or more VNICs from one underlying datalink to another underlying datalink without deleting and reconfiguring the VNICs. The underlying datalink can be a physical link, a link aggregation, or an etherstub.

You usually migrate a VNIC in any of the following situations:

- When you need to replace the existing NIC with a new NIC
- When the target NIC has more bandwidth than the existing NIC
- When the target NIC implements certain features in hardware, such as a large receive offload (LRO), a large segment offload (LSO), and checksum

To successfully migrate VNICs, the target datalink to which the VNICs are moved must be able to accommodate the datalink properties of the VNICs. If those properties are not supported, then migration fails and the user is notified. After a successful migration, all the applications that use the VNICs continue to operate normally, provided that the target datalink is connected to the network.

Certain hardware-dependent properties might change after a VNIC migration, such as the datalink state, link speed, and MTU size. The values of these properties are inherited from the datalink to which the VNICs are migrated. You can migrate all the VNICs that are configured over a datalink or selectively migrate the specified VNICs. You can also migrate the VNICs and modify their VLAN IDs simultaneously.

To migrate all the VNICs configured over the source link to the target link, use the following command syntax:

```
$ dladm modify-vnic -l target-link -L source-link
```

`-l` *target-link* Refers to the link over which the VNICs are migrated

`-L` *source-link* Refers to the link over which the VNICs were previously configured

EXAMPLE 23 Migrating All the VNICs From a Source Link to a Target Link

In this example, all the VNICs from the source link `ether0` are moved to the target link `net1`.

```
$ dladm modify-vnic -l net1 -L ether0
$ dladm show-vnic
LINK      OVER      SPEED         MACADDRESS         MACADDRTYPE        IDS
vnic0     net1      1000        2:8:20:c2:39:38     random             VID:321
```

```
vnic1     net1      1000         2:8:20:5f:84:ff     random            VID:656
vnic2     net1      1000         2:8:20:5f:84:ff     random            VID:0
```

To migrate the specified VNICs configured over the source link to the target link, use the following command syntax:

```
$ dladm modify-vnic -l target-link VNIC,VNIC,[...]
```

To perform selective VNIC migration, you need to specify only the target link.

EXAMPLE 24 Migrating Specified VNICs From a Source Link to a Target Link

In this example, `vnic0`, `vnic1`, and `vnic2` are selectively moved to the target link `net1` from the source link `net0`.

```
$ dladm modify-vnic -l net1 vnic0,vnic1,vnic2
$ dladm show-vnic
LINK      OVER      SPEED         MACADDRESS         MACADDRTYPE        IDS
vnic0     net1      1000         2:8:20:c2:39:38     random            VID:321
vnic1     net1      1000         2:8:20:5f:84:ff     random            VID:656
vnic2     net1      1000         2:8:20:5f:84:ff     random            VID:0
vnic3     net0      1000         2:8:20:5f:84:ff     random            VID:345
```

To modify the VLAN IDs of the VNICs configured over the source link and migrate them to the target link simultaneously, use the following command syntax:

```
$ dladm modify-vnic -l target-link -v vid VNIC
```

To assign new VLAN IDs, you must migrate the VNICs one at a time.

EXAMPLE 25 Migrating and Modifying the VLAN IDs of VNICs

In this example, `vnic0`, `vnic1`, and `vnic2` are migrated to the target datalink `net1`. With the migration, the VLAN IDs of all the VNICs are also modified simultaneously.

```
$ dladm modify-vnic -l net1 -v 123 vnic0
$ dladm modify-vnic -l net1 -v 456 vnic1
$ dladm modify-vnic -l net1 -v 789 vnic2
$ dladm show-vnic
LINK      OVER      SPEED         MACADDRESS         MACADDRTYPE        IDS
vnic0     net1      1000         2:8:20:c2:39:38     random            VID:123
vnic1     net1      1000         2:8:20:5f:84:ff     random            VID:456
vnic2     net1      1000         2:8:20:5f:84:ff     random            VID:789
```

When you migrate VNICs from the source link to the target link, randomly assigned MAC addresses are unaffected and retained by their respective VNICs after migration. See Example 25, “Migrating and Modifying the VLAN IDs of VNICs,” on page 66.

However, the MAC address will change if the VNIC is using a factory MAC address from the source link. If you do not specify a MAC address during migration, the factory MAC address of the VNIC is replaced by a randomly assigned MAC address. If you specify a MAC address with `-m` during migration, the factory MAC address of the VNIC is replaced by the specified MAC address.

You have multiple MAC addresses associated with VNICs created by kernel zones. When you migrate VNICs created by kernel zones, all the multiple MAC addresses associated with VNICs are migrated to the target NIC.

Deleting VNICs

This section describes how to delete a VNIC.

▼ How to Delete a VNIC

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **(Optional) Check whether the VNIC is busy.**

   You can delete a VNIC only when it is not busy. A VNIC can be busy for multiple reasons. You need to perform the following steps to check whether the VNIC busy:

   - **Check whether the VNIC is plumbed and associated with an IP address.**

     ```
     $ ipadm show-if
     $ ipadm show-addr
     ```

     If the VNIC is plumbed and associated with IP addresses, remove the IP interface.

     ```
     $ ipadm delete-ip interface
     ```

   - **Check whether there are any flows configured over the VNIC.**

     ```
     $ flowadm
     ```

     If flows are configured over the VNIC, remove the flow.

     ```
     $ flowadm remove-flow flowname
     ```

   - **Check whether the VNIC is assigned to a zone.**

```
$ dladm show-link -Z
```

For more information about how to delete a VNIC that is attached to a zone, see "How to Delete a VNIC Attached to a Zone" on page 68.

- **Check whether the VNIC is created by the system.**

```
$ dladm show-vnic
```

Only a system-created VNIC contains a hyphen (-), which helps you to differentiate between a system-created VNIC and a user-created VNIC. You cannot modify, rename, plumb, or delete system-created VNICs.

- **Check whether the VNIC is snooped.**

```
$ snoop
$ tshark
```

If the VNIC is snooped by using the `snoop` command, terminate the process.

```
$ pkill snoop
```

If the VNIC is snooped by using the `tshark` command, terminate the process.

```
$ pkill tshark
```

2. **Delete the VNIC.**

```
$ dladm delete-vnic VNIC
```

How to Delete a VNIC Attached to a Zone

This procedure assumes that the VNIC is attached to a zone. You must be in the global zone to perform this procedure.

1. **Halt the zone.**

```
global$ zoneadm -z zone halt
```

Note - To determine the links used by a zone, use the `dladm show-link` command.

2. **Remove or detach the VNIC from the zone.**

```
global$ zonecfg -z zone remove net physical=VNIC
```

3. **Delete the VNIC from the system.**

```
global$ dladm delete-vnic VNIC
```

4. **Reboot the zone.**

```
global$ zoneadm -z zone boot
```

Example 26 Deleting a VNIC Attached to a Zone

In this example, `vnic1` is removed from `zoneB` and from the system.

```
global$ dladm show-link
LINK            CLASS   MTU    STATE   OVER
net0            phys    1500   up      --
net2            phys    1500   up      --
net1            phys    1500   up      --
net3            phys    1500   up      --
zoneA/net0      vnic    1500   up      net0
zoneB/net0      vnic    1500   up      net0
vnic0           vnic    1500   up      net1
zoneA/vnic0     vnic    1500   up      net1
vnic1           vnic    1500   up      net1
zoneB/vnic1     vnic    1500   up      net1

global$ zoneadm -z zoneB halt
global$ zonecfg -z zoneB remove net physical=vnic1
global$ dladm delete-vnic vnic1
global$ zoneadm -z zoneB boot
```

Using Single Root I/O Virtualization With VNICs

Starting with the Oracle Solaris 11.2 release, you can manage network devices that support single root I/O virtualization (SR-IOV) by using the `dladm` command. SR-IOV, a standard that is implemented in the hardware enables efficient sharing of Peripheral Component Interconnect Express (PCIe) devices among virtual machines. For more information, see Chapter 21, “SR-IOV Drivers” in *Writing Device Drivers in Oracle Solaris 11.4*.

For information about using Oracle VM Server for SPARC Direct I/O (DIO) and SR-IOV features, and assigning Direct I/O devices or SR-IOV virtual functions to logical domains, refer to the MOS article Oracle VM Server for SPARC PCIe Direct I/O and SR-IOV Features (Doc ID 1325454.1) (`https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=181092870858172&id=1325454.1&_afrWindowMode=0&_adf.ctrl-state=fx25uncl3_53`).

Enabling the SR-IOV Mode of Datalinks

In Oracle Solaris, you can associate the virtual function (VF) of a network device that supports SR-IOV with a VNIC or a VLAN. A VF VNIC is a VNIC that owns a dedicated VF. A VF VNIC differs from a regular VNIC in the sharing of resources. A regular VNIC needs to share resources with other regular VNICs, but a VF VNIC need not share resources. Each VF is a separate hardware resource for the VF VNIC.

You can create VF VNICs only over datalinks that support the SR-IOV mode. By default, the SR-IOV mode of a datalink is disabled. You can enable the SR-IOV mode of a datalink by setting the `iov` property to `on`. For information about creating VF VNICs after you enable the SR-IOV mode of a datalink, see "Creating VF VNICs" on page 71.

You can check the SR-IOV mode of a datalink by specifying the link property `iov` with the `dladm show-linkprop` command. If the value under the `EFFECTIVE` column of the output is `off`, the SR-IOV mode of the datalink is disabled.

The following example shows how you can check the SR-IOV mode of the datalink `net0`.

```
$ dladm show-linkprop -p iov net0
LINK     PROPERTY   PERM     VALUE     EFFECTIVE     DEFAULT      POSSIBLE
net0     iov        rw       auto      off           auto         auto,on,off
```

In this example, the SR-IOV mode of the datalink `net0` is disabled. The output shows the following information:

`VALUE` — Specifies the value that you have set for the `iov` link property. If you have not modified the `iov` link property, the default value of the `iov` link property is `auto`. The value of `auto` means that the OS determines whether the SR-IOV mode is enabled by default on a particular physical datalink.

`EFFECTIVE` — The actual SR-IOV mode of the datalink. By default, all SR-IOV-capable NICs show the value `off` under the `EFFECTIVE` column.

You can enable the SR-IOV mode of the datalink `net0` by setting the `iov` property to `on` as follows:

```
$ dladm set-linkprop -p iov=on net0
$ dladm show-linkprop -p iov net0
LINK     PROPERTY   PERM     VALUE     EFFECTIVE     DEFAULT      POSSIBLE
net0     iov        rw       on        on            auto         auto,on,off
```

Similarly, you can disable the SR-IOV mode of a datalink by setting the `iov` link property to `off`. For more information about the `dladm` command, see the `dladm`(8) man page.

Creating VF VNICs

To create a VF VNIC on a datalink, you need to enable the SR-IOV mode of a datalink. For more information, see "Enabling the SR-IOV Mode of Datalinks" on page 70. After you enable the SR-IOV mode of a datalink, VFs are automatically allocated to VNICs when you create VNICs by using the `dladm create-vnic` command. Similarly, VFs are automatically allocated to VLANs when you create VLANs by using the `dladm create-vlan` command.

You can also explicitly specify whether a VF needs to be allocated to a VNIC or a VLAN by specifying the `iov` VNIC link property with the `dladm create-vnic` or the `dladm create-vlan` commands.

You use the following command syntax to explicitly create a VF VNIC:

```
$ dladm create-vnic [-p iov=value] -l link VNIC
```

When you are creating a VF VNIC, specifying the `iov` VNIC link property is optional. If you do not specify the `iov` VNIC link property, then the default value `inherit` is assigned to this property. You can specify the following values for the `iov` VNIC link property:

`inherit` — Default value of the `iov` VNIC link property. Determines whether a VF needs to be allocated based on the effective `iov` property value of the underlying datalink:

- `off` – Does not allocate a VF for a VNIC.
- `on` – Tries to allocate a VF for a VNIC. If not possible, a regular VNIC is created.

`on` — Allocates a VF. If a VF is not found, the creation of a VNIC fails.

`off` — Creates a VNIC without a VF.

The effective value of a datalink property is the value displayed under the `EFFECTIVE` column when you use the `dladm show-linkprop` command for a datalink.

The difference between the `iov` VNIC link property and other datalink properties is that you can specify the `iov` VNIC link property only when you are creating a VNIC or a VLAN. You cannot modify the `iov` VNIC link property after you create a VNIC or a VLAN.

The `iov` VNIC link property has an effective value that indicates whether a VF is allocated for the VNIC or VLAN. The value `on` under the `EFFECTIVE` column means that the VF is allocated and the value `off` under the `EFFECTIVE` column means that the VF is not allocated.

EXAMPLE 27 Creating a VF VNIC

The following example shows how to create the VF VNIC `vfvnic1` and the regular VNIC `vnic1` on the datalink `net0` by explicitly specifying the `iov` VNIC link property. This example assumes that you have enabled the SR-IOV mode of the datalink `net0`.

```
$ dladm show-linkprop -p iov net0
LINK     PROPERTY  PERM    VALUE  EFFECTIVE  DEFAULT   POSSIBLE
net0     iov       rw      on     on         auto      auto,on,off
$ dladm create-vnic -l net0 vfvnic1
$ dladm show-linkprop -p iov vfvnic1
LINK      PROPERTY   PERM   VALUE    EFFECTIVE   DEFAULT    POSSIBLE
vfvnic1   iov        r-     inherit  on          inherit    inherit,on,off
$ dladm create-vnic -p iov=off -l net0 vnic1
$ dladm show-linkprop -p iov vnic1
LINK    PROPERTY   PERM   VALUE    EFFECTIVE   DEFAULT     POSSIBLE
vnic1   iov        r-     off      off         inherit     inherit,on,off
```

This example provides the following information:

- You need to set the `iov` property for the datalink `net0` to `on` before you create the VF VNICs.
- If you do not specify a value for the `iov` property when creating a VNIC, then the default value `inherit` is assigned to the `iov` property. The VF VNIC `vfvnic1` is created with a VF.
- If you explicitly specify the value `off` for the `iov` property when creating a VNIC, a regular VNIC is created without a VF even though the `iov` property of the underlying datalink `net0` is `on`. The VNIC `vnic1` is created without a VF.

Migrating VF VNICs

You can move VF VNICs or VF VLANs from one datalink to another datalink. Note the following requirements:

- The target datalink must support SR-IOV and the `iov` property must be set to `on`. For more information about how to check the status of the `iov` property for a datalink, see "Enabling the SR-IOV Mode of Datalinks" on page 70.
- A VF must be available on the target datalink. For more information about how to check the number of VFs available on a datalink, see "Displaying VF Information" on page 75.

If these requirements are not met, then the VF VNIC is migrated to the target datalink as a regular VNIC without a VF.

If you migrate a VF VNIC, that was created by specifying `iov=inherit`, the migration succeeds even if the target datalink does not support the `iov` property or the `iov` property is disabled. If

you try to migrate a VF VNIC, that was created with `iov=on`, the migration succeeds only if the SR-IOV mode is enabled on the target datalink.

For more information about how to migrate a VNIC, see "Migrating VNICs" on page 65.

Configuring Oracle Solaris Kernel Zones With SR-IOV VFs

You can configure the `anet` resource of a kernel zone with the available SR-IOV VF by setting the `iov` property through the `zonecfg` command.

`auto`	Allocates a VF if it is available. Otherwise, uses a paravirtual device.
`on`	Allocates a VF. If a VF is not available, the `anet` resource creation fails. For information about how to check the available VFs on a datalink, see Example 29, "Displaying VFs Information for Datalinks," on page 75.
`off`	VF is not allocated. The `off` value is the default value for the `iov` property.

For more information about SR-IOV on kernel zones, see "Managing Single-Root I/O NIC Virtualization on Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones*.

EXAMPLE 28 Configuring Kernel Zones With SR-IOV VFs

This example shows how to configure the `anet` resource of the kernel zone `kz1` with a SR-IOV VF.

```
$ zonecfg -z kz1
zonecfg:kz1> select anet id=0
zonecfg:kz1:anet> set iov=auto
zonecfg:kz1:anet> end
zonecfg:kz1> exit
```

If you configure the `anet` resource over the lower datalink `net1`, you must ensure that the `iov` link property for `net1` is set to `on` before booting the kernel zone `kz1`. You can check the `iov` property for the lower datalink `net1`.

```
$ dladm show-linkprop -p iov net1
LINK     PROPERTY    PERM   VALUE     EFFECTIVE    DEFAULT   POSSIBLE
```

```
net1     iov         rw     off       off          auto      auto,on,off
```

The output shows that the value of the `iov` property is `off` for the lower datalink `net1`. Set the `iov` property to `on`.

```
$ dladm set-linkprop -p iov=on net1
```

After you boot the kernel zone, a VF is successfully allocated to the `anet` resource. Verify whether the VF is added to the kernel zone `kz1`.

```
$ zlogin kz1
kz1$ dladm show-phys
LINK              MEDIA               STATE      SPEED  DUPLEX    DEVICE
net0              Ethernet            up         10000  full      ixgbevf0
```

The limitations of using the `iov` property with kernel zones are as follows:

- You cannot use the `iov` property with native zones because the `iov` property does not provide any benefit for native zones.
- You cannot set the `iov` property to `auto` or `on` if the `anet` resource is configured with any of the following properties:
  - `allowed-address`
  - `configure-allowed-address`
  - `defrouter`
  - `allowed-dhcp-cids`
  - `link-protection`
  - `vlan-id`
  - `tx-rings`
  - `rx-rings`
  - `mtu`
  - `rx-fanout`
  - `vsi-type-id`
  - `vsi-version`
  - `vsi-manager-id`
  - `ets-bw-local`
  - `cos`
  - `evs`
  - `vport`

  Similarly, you cannot set these properties if you have already set the `iov` property to `auto` or `on`.

- After you create a VF `anet` resource, it appears as a VNIC in the host similar to the other regular `anet` resources. The only difference is that you cannot modify any link property for the VF `anet` resource.
- You can add multiple VF `anet` resources to a kernel zone. However, the VF physical links that appear in a kernel zone cannot be aggregated.
- If you set the `iov` property to `on` or `auto`, the kernel zone does not support live migration and suspend or resume operations. The `zoneadm migrate` or `zoneadm suspend` commands fail.

Displaying VF Information

To display information about the availability of VFs on a datalink, use the following command:

```
$ dladm show-phys -V
```

The output shows the following information:

`LINK`	Name of the datalink.
`VFS-AVAIL`	Number of VFs available on a datalink that can be assigned to a VNIC. If the datalink does not support SR-IOV, `VFS-AVAIL` is shown as `--`.
`VFS-INUSE`	Number of VFs that are used by a datalink. If the datalink does not support SR-IOV, `VFS-INUSE` is shown as `--`.
`FLAGS`	The `l` flag indicates that the datalink is managed by Oracle VM Server for SPARC.

EXAMPLE 29 Displaying VFs Information for Datalinks

```
$ dladm show-phys -V
LINK       VFS-AVAIL      VFS-INUSE       FLAGS
net0       30             1               -----
net1       0              0               l----
net2       --             --              -----
```

In this example, the datalink `net0` has `30` available VFs and one VF in use. The datalink `net1` has zero (`0`) available VFs and it is currently being used by Oracle VM Server for SPARC. The datalink `net2` does not support SR-IOV.

You can display the VF devices assigned to VNICs on a system by using the following command:

```
$ dladm show-vnic -V
```

The output shows the following information:

`LINK` Name of the VNIC.

`VF-ASSIGNED` VF device assigned to the VNIC. If the VNIC does not have a VF, `VF-ASSIGNED` is shown as `--`.

EXAMPLE 30 Displaying VF Devices Assigned to VNICs

```
$ dladm show-vnic -V
LINK         VF-ASSIGNED
vnic1        ixgbevf0
vnic2        --
vnic3        ixgbevf1
```

In this example, the VF device `ixgbevf0` is assigned to `vnic1`. The VNIC `vnic2` does not have an allocated VF device. The VF device `ixgbevf1` is assigned to `vnic3`.

Setting Hardware SLA Properties for VF VNICs

If a NIC supports hardware SLAs that enable you to set SLA properties for VF VNICs, the SLA implementation is offloaded to the NIC automatically by the system. This behaviour helps you to save CPU cycles.

You can use the `dladm show-linkprop` command with the `-H` option to check the capabilities of the underlying datalink. The command syntax is:

```
$ dladm show-linkprop -H -p prop link
```

where *prop* refers to the SLA properties such as `max-bw`, `priority`, and `bw-share`.

The output displays the following columns:

`HWPOSSIBLE` Displays a value if there is hardware support for the property. The physical NIC does not support the property if the value is displayed as `--`.

`SWPOSSIBLE` Displays a value if there is software support in the networking stack for the property. The datalink does not support the property if the value is displayed as `--`.

Note - For both the HWPOSSIBLE and SWPOSSIBLE columns, the step value requirement for the value is displayed after the number range followed by a colon (:), for example, 50-40000:50. Currently, only the max-bw property shows a value for the step value.

MODE	Displays the current mode that is used for the datalink to implement the property. The possible values are sw for software only, hw for hardware only, and none for no support. Note that MODE can be none even though there is hardware or software support.
HWFLAGS or SWFLAGS	Displays o for outbound, i for inbound, and oi for outbound and inbound. Currently, these flags are displayed for the SLA properties max-bw, bw-share, and priority.

If the datalink supports hardware SLAs, you can set the hardware SLA properties on the datalink by using the following command:

```
$ dladm set-linkprop -p prop=value link
```

where *prop* refers to the SLA properties such as max-bw, priority, and bw-share.

EXAMPLE 31 Displaying the Hardware and Software Capabilities of Datalinks

The following example shows the output of the max-bw property for the VF VNIC z1/net1 that is configured over the Intel XL710 10/40 Gigabit Ethernet controller NIC. The output shows that there is both hardware and software support because values are displayed under the columns HWPOSSIBLE and SWPOSSIBLE.

```
$ dladm show-linkprop -H -p max-bw z1/net1
LINK          PROPERTY       MODE  HWPOSSIBLE   HWFLAGS SWPOSSIBLE     SWFLAGS
z1/net1       max-bw         hw    50-40000:50  o       0-40000:0.001  oi
```

The following example shows the output of the max-bw property for the VF VNIC z2/net2 that is configured over the Niantic NIC. The output shows that there is only software support for the VF VNIC z2/net2 because values are displayed under the column SWPOSSIBLE.

```
$ dladm show-linkprop -H -p max-bw z2/net2
LINK          PROPERTY       MODE  HWPOSSIBLE   HWFLAGS SWPOSSIBLE     SWFLAGS
z2/net2       max-bw         none  --                   0-10000:0.001  oi
```

The following example shows the output of the bw-share property for the VF VNIC z1/net1 that is configured over the Intel XL710 10/40 Gigabit Ethernet controller NIC. For information about the bw-share property, see “Bandwidth Share for VNICs” on page 78The output shows that there is only hardware support for the VF VNIC z1/net1 for the bw-share property. The values are displayed under the column HWPOSSIBLE.

```
$ dladm show-linkprop -H -p bw-share z1/net1
LINK          PROPERTY       MODE  HWPOSSIBLE   HWFLAGS SWPOSSIBLE     SWFLAGS
z1/net1       bw-share       hw    1-100        o       --
```

The following example shows the output of the `bw-share` property for the VF VNIC `z2/net2` that is configured over the Niantic NIC. The output shows that there is no support for the VF VNIC `z2/net2` for the `bw-share` property. The values are not displayed under the columns `HWPOSSIBLE` and `SWPOSSIBLE`.

```
$ dladm show-linkprop -H -p bw-share z2/net2
LINK          PROPERTY       MODE  HWPOSSIBLE   HWFLAGS SWPOSSIBLE     SWFLAGS
z2/net2       bw-share       none  --                   --
```

EXAMPLE 32 Setting Maximum Bandwidth for a VF VNIC

The following example shows how to set the `max-bw` property for the VF VNIC `z1/net21`.

```
$ dladm set-linkprop -p max-bw=20 -t z1/net21
$ dladm show-linkprop -p max-bw z1/net21
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
z1/net21 max-bw          rw   20           50           --        --
```

In certain cases, the effective value for the `max-bw` property can be different from the set value for the VF VNIC that is configured over a hardware SLA capable link as shown in this example.

Bandwidth Share for VNICs

Bandwidth share for a VNIC is the minimum share of the bandwidth that the VNIC will get when there is competition from other VNICs on the same datalink. You use the `bw-share` property to allocate the bandwidth share for a VNIC. You can allocate the bandwidth share only on the datalink that supports the `bw-share` property. Currently, only the Intel XL710 10/40 Gigabit Ethernet controller NIC supports the `bw-share` property. You can check whether a datalink supports the `bw-share` property by using the `dladm show-linkprop` command. See Example 33, “Determining Whether a Datalink Supports the `bw-share` Property,” on page 79.

Note that the bandwidth is allocated among all the active VNICs. The amount of bandwidth that is allocated to a VNIC is proportional to the bandwidth share that is set for the VNIC. For example, consider two VNICs, `vnic1` and `vnic2`, configured on a 1 gigabits per second (Gbps) link. You set the `bw-share` property on `vnic1` and `vnic2` as follows:

```
$ dladm set-linkprop -p bw-share=40 vnic1
```

```
$ dladm set-linkprop -p bw-share=10 vnic2
```

In this example, the bandwidth share of `vnic1` is `40` and `vnic2` is `10`. Because the VNICs are configured on a 1 Gbps link, `vnic1` can use up to 800 megabits per second (Mbps) of bandwidth (1Gbps * 40/(40+10)) and `vnic2` can use up to 200 Mbps of bandwidth (1Gbps * 10/(40+10)).

This example assumes that both the VNICs have network traffic to consume their share of the bandwidth. However, if `vnic1` uses only 100 Mbps, then `vnic2` can use up to 900 Mbps. By using bandwidth shares, no bandwidth is wasted when there is a VNIC that can use the bandwidth. At the same time, bandwidth shares ensure an allocated share for a VNIC when there is competition from other VNICs.

Considerations for the `bw-share` Property

Note the following considerations when using the `bw-share` property:

- You can assign a value from `1` to `100` for the `bw-share` property. The value is a relative share value and does not indicate a percentage of the bandwidth. The value can be indicated as a percentage if you keep the sum of the values for the `bw-share` property for all the VNICs on a link at or below `100`.
- For the `dladm show-linkprop` command output, the effective value for the `bw-share` property is displayed as a percentage. The effective value is the minimum percentage of the bandwidth guaranteed to the VNIC when there is competition from other VNICs on the same datalink. The effective value changes depending on the other VNICs that are configured on the datalink.
- If you have set the `max-bw` property for the VNIC, the traffic is limited by the `max-bw` value. The `max-bw` property is enforced on the VNIC before the `bw-share` property is applied.
- You can have VNICs that are set with the `bw-share` property and VNICs that are not set with `bw-share` property on the same datalink. In this case, the share of the bandwidth is undefined for the VNICs that are not set with the `bw-share` property. There is no change to the current behavior, if you have not set the `bw-share` on any VNIC on a link.

EXAMPLE 33 Determining Whether a Datalink Supports the `bw-share` Property

The following example shows how to check whether a datalink supports the `bw-share` property. In this example, the `z1/net1` datalink is a VF VNIC. The value `1-100` under the `POSSIBLE` column in the output indicates that the underlying datalink supports the `bw-share` property.

```
$ dladm show-linkprop -p bw-share
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
z1/net1  bw-share         rw   --           --           --        1-100
```

The following example shows the output for a net0 datalink that does not support the bw-share property.

```
$ dladm show-linkprop -p bw-share
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     bw-share         r-   --           --           --        --
```

EXAMPLE 34 Setting the bw-share Bandwidth Property for a VNIC

The following example shows how to set the bw-share property for the VF VNIC z1/net1.

```
$ dladm set-linkprop -t -p bw-share=60 z1/net1
$ dladm show-linkprop -p bw-share
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
z1/net1  bw-share         rw   60           100%         --        1-100
```

The value under the EFFECTIVE column indicates that the VNIC z1/net1 uses 100% of the bandwidth. However, the effective value changes when you set the bw-share property for the VNIC z1/net2 configured on the same underlying datalink.

```
$ dladm show-linkprop -p bw-share
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
z1/net1  bw-share         rw   60           50%          --        1-100
z1/net2  bw-share         rw   60           50%          --        1-100
```

The output shows that the effective value for z1/net1 has changed from 100% to 50%.

Interaction of the bw-share Property With DCB Bandwidth Shares

The etsbw_lcl property supports the setting of the bandwidth share as a fixed percentage of the bandwidth of the physical NIC. However, it is supported only if the NIC is in DCB mode. DCB mode is not on by default and DCB is only used when the switch supports DCB.

You cannot set the bw-share property if the NIC is in DCB mode. The bw-share property is not effective if you set the bw-share property and then set DCB mode to on. In this case, the EFFECTIVE value is displayed as -- for the dladm show-linkprop command output.

Bandwidth Share for VNIC anet Resources

You can set the bw-share property for a VNIC anet resource that is configured with a zone. The bw-share property does not have a default value. You can assign a value from 1 to 100.

The booting of the zone fails if you specify a lower link that does not support the `bw-share` property. For information about how to check whether a link supports `bw-share`, see Example 33, “Determining Whether a Datalink Supports the `bw-share` Property,” on page 79.

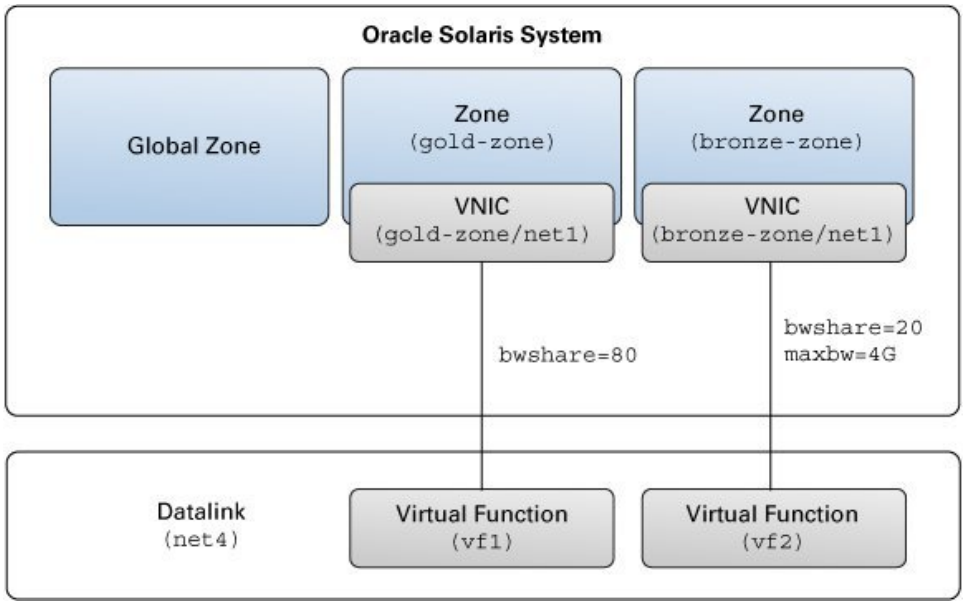
Use Case: Offloading Hardware SLAs to a NIC

Objective - This use case shows how to configure two kernel zones with VF VNICs and offload the SLAs of the VF VNICs to the underlying physical NIC.

Typically, you cannot set SLA properties such as `max-bw` and `priority` on the VF VNIC because the VF VNIC bypasses the global zone. However, you can offload the SLA implementation to the NIC if it is supported by the NIC. The Intel XL710 10/40 Gigabit Ethernet controller NIC supports the offloading of SLAs and supports bandwidth shares in addition to the `max-bw` property. For information about bandwidth shares, see “Bandwidth Share for VNICs” on page 78.

The following figure shows the Oracle Solaris system setup used in this use case.

FIGURE 7 Use Case: Kernel Zones With SR-IOV VF VNICs



The setup is as follows:

- An Oracle Solaris system with a global zone.
- The datalink `net4`, which is configured over the Intel XL710 10/40 Gigabit Ethernet controller NIC with 10 Gbps bandwidth.
- Two kernel zones: `gold-zone` and `bronze-zone`.
- `gold-zone` is assigned a bandwidth share of 80% (`bw-share=80`). The kernel zone `bronze-zone` is assigned a bandwidth share of 20% (`bw-share=20`) and maximum bandwidth of 4 Gbps (`maxbw=4G`).

▼ How to Offload Hardware SLAs to a NIC (Use Case)

You need to perform the following steps to offload the SLA properties to the NIC:

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Set the `iov` property for the datalink `net4` to `on` before you create the VF VNICs.**

```
$ dladm set-linkprop -p iov=on net4
```

2. **Check whether the datalink `net4` supports the `bw-share` property.**

```
$ dladm show-linkprop -H -p bw-share net4
LINK     PROPERTY        MODE HWPOSSIBLE    HWFLAGS SWPOSSIBLE SWFLAGS
net4     bw-share         none 1-100         --      --         --
```

The output shows that the physical datalink `net4` supports the `bw-share` property because the value `1-100` is displayed under the column `HWPOSSIBLE`.

3. **Create a VF VNIC for `gold-zone` and set the bandwidth share to `80`.**

```
$ zonecfg -z gold-zone
zonecfg:gold-zone> add anet
zonecfg:gold-zone:anet> set lower-link=net4
zonecfg:gold-zone:anet> set iov=on
zonecfg:gold-zone:anet> set bw-share=80
zonecfg:gold-zone:anet> end
zonecfg:gold-zone> verify
zonecfg:gold-zone> commit
zonecfg:gold-zone> exit
```

4. **Create a VF VNIC for `bronze-zone` and set the bandwidth share to `20` and the maximum bandwidth to `4G`.**

```
$ zonecfg -z bronze-zone
zonecfg:bronze-zone> add anet
zonecfg:bronze-zone:anet> set lower-link=net4
zonecfg:bronze-zone:anet> set iov=on
zonecfg:bronze-zone:anet> set bw-share=20
zonecfg:bronze-zone:anet> set maxbw=4G
zonecfg:bronze-zone:anet> end
zonecfg:bronze-zone> verify
zonecfg:bronze-zone> commit
zonecfg:bronze-zone> exit
```

5. **Boot the kernel zones.**

```
$ zoneadm -z gold-zone boot
$ zoneadm -z bronze-zone boot
```

6. **Check the bandwidth share of the VF VNICs.**

```
$ dladm show-linkprop -p bw-share
LINK             PROPERTY    PERM  VALUE     EFFECTIVE    DEFAULT   POSSIBLE
gold-zone/net1   bw-share     rw    80        80%          --        1-100
bronze-zone/net1 bw-share     rw    20        20%          --        1-100
```

Note - In this use case, the total bandwidth share is kept at 100. A relative share is assigned to the VF VNICs if the total bandwidth share exceeds 100. For more information, see the dladm(8) man page.

7. **Check the maximum bandwidth allocated to the VF VNICs.**

```
$ dladm show-linkprop -p max-bw
LINK              PROPERTY   PERM  VALUE      EFFECTIVE    DEFAULT   POSSIBLE
gold-zone/net1    max-bw     rw    --         --           --        --
bronze-zone/net1  max-bw     rw    4000       4000         --        --
```

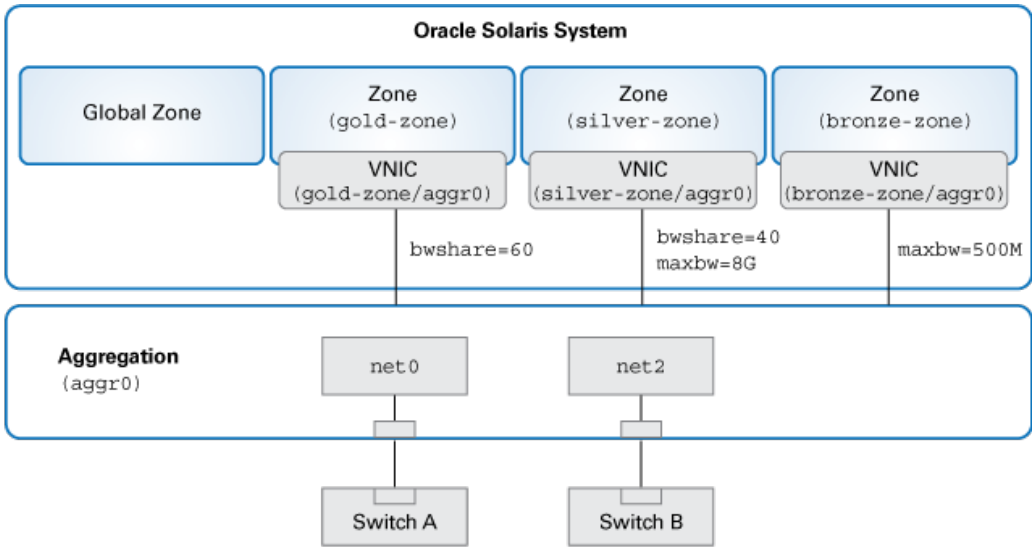
Use Case: Offloading Hardware SLAs to DLMP Aggregated Datalink for High Availability

Objective - This use case shows how to configure a DLMP aggregation of SR-IOV- enabled datalinks, and configure the aggregated datalink as the lower link in the kernel zone and offload hardware SLAs to the aggregated datalink. DLMP aggregation helps to achieve high availability without compromising hardware SLA offloading capability.

The objective for creating DLMP as opposed to trunk Aggregation is to achieve high availability without comprising hardware SLA offloading capability.

The following figure shows the Oracle Solaris system setup used in this use case.

FIGURE 8 Offloading Hardware SLAs to DLMP Aggregated Datalink



The setup is as follows:

- An Oracle Solaris system with a global zone.
- The datalinks `net0` and `net2` are aggregated to form the DLMP aggregation, `aggr0`.
- Three kernel zones: `gold-zone`, `silver-zone`, and `bronze-zone`.
- `gold-zone` is assigned a bandwidth share of 60% (`bw-share=60`). The kernel zone `silver-zone` is assigned a bandwidth share of 40% (`bw-share=40`) and a maximum bandwidth of 8G. The kernel zone `bronze-zone` is assigned a maximum bandwidth of 500Mbps (`maxbw=500M`).

▼ How to Offload Hardware SLAs to DLMP Aggregated Datalink for High Availability (Use Case)

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Set the `iov` property to `on` for the datalinks before you create the aggregation.**

```
$ dladm set-linkprop -p iov=on net0
$ dladm set-linkprop -p iov=on net2
```

2. **Check whether the member port datalinks `net0` and `net2` for DLMP support the `bw-share` property.**

```
$ dladm show-linkprop -H -p bw-share net0
LINK     PROPERTY        MODE HWPOSSIBLE    HWFLAGS SWPOSSIBLE SWFLAGS
net0     bw-share        none  1-100         --      --         --
$ dladm show-linkprop -H -p bw-share net2
LINK     PROPERTY        MODE HWPOSSIBLE    HWFLAGS SWPOSSIBLE SWFLAGS
net0     bw-share        none  1-100         --      --         --
```

   The output shows that the physical datalink `net0` and `net4` support the `bw-share` property because the value `1-100` is displayed under the column `HWPOSSIBLE` for both the datalinks.

3. **Create a DLMP aggregation.**

```
$ dladm create-aggr -l net0 -l net2 -m dlmp aggr0
```

4. **Configure the kernel zone, `gold-zone`, and set the bandwidth share to 60.**

```
$ zonecfg -z gold-zone
zonecfg:gold-zone> create -t SYSsolaris-kz
zonecfg:gold-zone> add anet
zonecfg:gold-zone:anet> set lower-link=aggr0
zonecfg:gold-zone:anet> set iov=auto
zonecfg:gold-zone:anet> set bw-share=60
zonecfg:gold-zone:anet> end
zonecfg:gold-zone> commit
zonecfg:gold-zone> exit
```

5. **Configure the kernel zone, `silver-zone`, and set the bandwidth share to `60` and maximum bandwidth to `8G`.**

```
$ zonecfg -z silver-zone
zonecfg:silver-zone> create -t SYSsolaris-kz
zonecfg:silver-zone> add anet
zonecfg:silver-zone:anet> set lower-link=aggr0
zonecfg:silver-zone:anet> set iov=auto
zonecfg:silver-zone:anet> set bw-share=40
zonecfg:silver-zone:anet> set id=0
zonecfg:silver-zone:anet> end
zonecfg:silver-zone> add anet
zonecfg:silver-zone:anet> set lower-link=aggr0
```

```
zonecfg:silver-zone:anet> set iov=auto
zonecfg:silver-zone:anet> set maxbw=8G
zonecfg:silver-zone:anet> set id=1
zonecfg:silver-zone:anet> end
zonecfg:silver-zone> commit
zonecfg:silver-zone> exit
```

6. **Configure the kernel zone, `bronze-zone`, and set the maximum bandwidth to 500M.**

```
$ zonecfg -z bronze-zone
zonecfg:bronze-zone> create -t SYSsolaris-kz
zonecfg:bronze-zone> add anet
zonecfg:bronze-zone:anet> set lower-link=aggr0
zonecfg:bronze-zone:anet> set iov=off
zonecfg:bronze-zone:anet> set maxbw=500M
zonecfg:bronze-zone:anet> end
zonecfg:bronze-zone> commit
zonecfg:bronze-zone> exit
```

In this example, the three zones obtain different levels of SLA in addition to the failover protection of high availability.

Creating and Viewing Paravirtualized IPoIB Datalinks in Kernel Zones

Paravirtual (PV) drivers are high-performance network and disk drivers that significantly reduce the overhead of the traditional implementation of I/O device emulation. These drivers provide improved network performance, disk throughput, and system efficiency because these drivers do no not emulate other devices such as physical NICs. The paravirtualized network driver `ZVNET` for Oracle Solaris Kernel Zones, interact with the hypervisor in the host OS through hypercall to achieve low-delay and high-throughput network performance.

About Paravirtualized IPoIB Datalinks in Oracle Solaris

Starting with Oracle Solaris 11.3, the paravirtualized IPoIB datalink is created as an `anet` resource in Oracle Solaris Kernel Zone and you can configure this datalink by using the `zonecfg` command. The `anet` resource creates an IPoIB VNIC when the kernel zone boots up. The IPoIB VNIC is created over a partition of the lower link InfiniBand host channel adapter

(IB HCA) and the port tuple in the global zone. Each IPoIB VNIC has one-to-one match and communicates with paravirtualized IPoIB datalink in the kernel zone. Each of these VNICs have a unique MAC address and can have a unique or different partition key (`pkey`). For each `anet` resource, you can configure the mode over which the IPoIB datalinks are run. Connected mode (CM) and unreliable datagram (UD) mode are supported and you can configure these modes by using the `zonecfg` command. For more information, see "Zone Resource Types and Their Properties" in *Oracle Solaris Zones Configuration Resources*.

To display the configured IPoIB datalinks within the kernel zone, use the `dladm` command.

EXAMPLE 35 Creating a Paravirtualized IPoIB Datalink

You create a paravirtualized IPoIB datalink by creating an automatic network (`anet`) in the kernel zone and specifying the mandatory properties `lower-link` and `pkey`. Set the `lower-link` property to one of the valid IB partitions and set `pkey` to one of the partition keys provided by that partition. The property `linkmode`, which can be either `cm` or `ud`, is optional. If you do not specify a value, the value is set to `cm` by default.

```
$ zonecfg -z kzone0
zonecfg:kzone0> add anet
zonecfg:kzone0:anet> set lower-link=net1
zonecfg:kzone0:anet> set pkey=0x8001
zonecfg:kzone0:anet> set linkmode=cm
zonecfg:kzone0:anet> end

$ zoneadm -z kzone0 boot
```

EXAMPLE 36 Displaying Physical Device Information in Kernel Zones

The following example displays the physical device and attributes of all physical datalinks in a kernel zone including the InfiniBand devices.

```
solariskzone0:~$ dladm show-phys
LINK              MEDIA               STATE      SPEED DUPLEX    DEVICE
net0              Ethernet            up         1000  full      zvnet0
net1              Infiniband          up         32000 full      zvnet1
```

The following example displays the physical device and all the key attributes of physical links in a kernel zone.

```
solariszone1:~$ dladm show-phys -o all
LINK   MEDIA        STATE    SPEED DUPLEX    DEVICE    VFS-AVAIL   VFS-INUSE   FLAGS
net0   Ethernet       up     1000  full      zvnet0     --          --          -----
net1   Infiniband     up     32000 full      zvnet1     --          --          -----
```

EXAMPLE 37 Displaying MAC Addresses for the Physical Device

The following example displays the MAC addresses for the physical device in a kernel zone.

```
solariszone1:~$ dladm show-phys -m
LINK                SLOT     ADDRESS             INUSE  CLIENT
net0                primary  2:8:20:5:32:5a      yes    net0
net1                primary  80:0:0:4a:fe:80:... yes    net1
```

EXAMPLE 38 Displaying the IPoIB VNIC in the Host

The following example displays the IPoIB VNIC in the host.

```
solariszone1:~$ dladm show-vnic
LINK            OVER     SPEED  MACADDRESS      MACADDRTYPE IDS
kzone1/net0     net0     1000   2:8:20:5:32:5a  random      VID:0
kzone1/net1     net4     32000  80:0:0:4a:fe:.. fixed       PKEY:0x8001
```

In this example, the notation `PKEY` in the `IDS` field indicates that the VNIC is an IPoIB VNIC.

The following example displays the MAC addresses of the IPoIB VNICs.

```
solariszone1:~$ dladm show-vnic -o macaddress
MACADDRESS
2:8:20:5:32:5a
80:0:0:4a:fe:80:0:0:0:0:0:0:0:21:28:0:1:a0:e5:55
```

EXAMPLE 39 Displaying Datalinks in the Host

The following example displays the data links in the host including the IPoIB VNIC created on the kernel zone.

```
solariszone1:~$ dladm show-link

LINK                CLASS     MTU    STATE    OVER
net0                phys      1500   up       --
net1                phys      1500   unknown  --
kzone1/net0         vnic      1500   up       net0
kzone1/net1         vnic      65520  up       net1
```

For more information, see the `dladm`(8) man page.

Creating VNICs Over Paravirtualized IPoIB Datalink

You can create an IPoIB VNIC over a paravirtualized IPoIB datalink in an Oracle Solaris Kernel Zone. The IPoIB VNIC is assigned the primary MAC address of the IPoIB datalink instance if it is not already used by another MAC client. As a result, you can create a non-global zone inside a kernel zone by using the paravirtualized IPoIB datalink as the lower link. The IPoIB VNIC inherits the partition key stored in the paravirtualized IPoIB link, which is again inherited from the `pkey` of the IB partition in the global zone.

To support multiple Oracle Solaris zones in a kernel zone, multiple IPoIB `anet` instances need to be assigned to the kernel zone from the global zone, and the kernel zone needs to assign a different `anet` to each of its zones.

You can create the IPoIB VNIC over paravirtualized IPoIB datalink similar to the way in which you create IPoIB VNICs over the IB partition in the global zone. For information, see "Configuring IPoIB VNICs" on page 42.

You can specify the partition key while creating the IPoIB VNIC. A paravirtualized IPoIB datalink stores the `pkey` that is inherited from the partition created in the host. You can specify this `pkey` value when you create the VNIC. You can use the `dladm show-phys -v` command to see the `pkey` value of the paravirtualized datalink. However, you can also create a VNIC without specifying the `pkey`. In this case, the inherited `pkey` is assigned by default.

EXAMPLE 40 Creating an IPoIB VNIC Over Paravirtualized IPoIB Datalink by Specifying the `pkey`

The following example shows how to create the IPoIB VNIC by specifying the `pkey` over the IPoIB datalink `net1`.

```
$ dladm show-phys -v
LINK    IDS           INUSE CLIENT
net0    VIDS:23       yes   --
net1    PKEYS:ffff    no    --
$ dladm show-phys
LINK          MEDIA             STATE      SPEED  DUPLEX    DEVICE
net0          Ethernet          up         1000   full      zvnet0
net1          Infiniband        up         32000  full      zvnet1
$ dladm create-vnic -l net1 -P 0xffff vnic1
$ dladm
LINK                CLASS     MTU    STATE    OVER
net0                phys      1500   up       --
net1                phys      65520  up       --
vnic1               vnic      65520  up       net1
$ dladm show-vnic vnic1
```

```
LINK       OVER        SPEED  MACADDRESS        MACADDRTYPE IDS
vnic1      net1        32000  80:0:0:4a:fe:..   fixed       PKEY:0xffff
$ dladm show-phys -v
LINK    IDS           INUSE CLIENT
net0    VIDS:23       yes   --
net1    PKEYS:ffff    yes   vnic1
```

EXAMPLE 41 Creating an IPoIB VNIC Over Paravirtualized IPoIB Datalink Without Specifying the `pkey`

The following example shows how to create IPoIB VNIC `vnic1` without specifying the `pkey` over the IPoIB datalink `net1`.

```
$ dladm show-phys -v
LINK    ID           INUSE CLIENT
net0    VIDS:23       yes   --
net1    PKEYS:ffff    no    --

$ dladm create-vnic -l net1 vnic1
$ dladm
LINK                CLASS     MTU    STATE    OVER
net0                phys      1500   up       --
net1                phys      65520  up       --
vnic1               vnic      65520  up       net1
```

In this example, the `pkey` value inherited by the IPoIB datalink is assigned to the VNIC.

EXAMPLE 42 Displaying the Partition Key

The following example displays the partition key `pkey` for an IPoIB datalink.

```
$ dladm show-phys -v -o link,vids,pkeys
LINK    VIDS       PKEYS
net0    23         --
net1    --         ffff
```

Configuring a Virtual Network Interface

Virtual network interface is a software-only interface that does not have any hardware associated with it. This interface does not send or receive any data because there is no physical hardware associated with it. This interface provides a datalink provider interface (DLPI) and identifies itself with an IP address and a private media type. The virtual network interface is configured by using the `ipadm` command.

The virtual network interface can handle both IPv4 and IPv6 packets. By default, the interface is enabled for both IPv4 and IPv6 addresses when the interface is created. The virtual network interfaces are persistent.

This interface is useful in hosting an IP address when you use it in conjunction with the `usesrc` interface property of the IP interface. The virtual interface is also useful in hosting a virtual IP (VIP) address that is used for Integrated Load Balancer (ILB) in the Direct Server Return (DSR) mode. A back end server in an ILB set up needs to have a VIP address of an ILB rule hosted by a virtual interface so that the server accepts packets from client destined to the VIP. For more information about ILB, see "Configuring ILB for High Availability By Using the DSR Topology" in *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer*.

You can create a virtual network interface by using the `ipadm create-vni` command.

EXAMPLE 43 Creating a Virtual Network Interfaces for ILB in DSR mode

Assume that you have an ILB DSR set up with a virtual server IP address `192.0.2.200` and the VIP needs to be hosted in each of the back end servers. You can create the virtual network interface to host the VIP in the back end servers.

```
$ ipadm create-vni vip0
$ ipadm create-addr -T static -a 192.0.2.200/32 vip0/v4
```

In this example, a back end server accepts ILB forwarded packets from a client to the VIP, `192.0.2.200`.

♦ ♦ ♦ CHAPTER 3

Configuring Virtual Extensible Local Area Networks

This chapter provides an overview of deploying VXLANs and describes how to configure them. It contains the following topics:

Note - To configure VXLANs and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 34.

About VXLANs

In a cloud environment, systems might be located in different Layer 2 networks. In such cases, creating virtual machines (VMs) or *tenants* over a Layer 2 network restricts the number of systems that you can use for provisioning these VMs. You can use systems in different Layer 2 networks for provisioning VMs. However, as the migration between different systems is restricted to the same Layer 2 network, the utilization of the physical resource is not optimized.

VXLAN is a Layer 2 technology that enables you to create a Layer 2 network on top of a Layer 3 network, thereby providing further network isolation. VXLAN provides a virtual Layer 2 network that stretches over multiple physical Layer 2 networks. Therefore, provisioning

resources in a cloud environment is not restricted to a single physical Layer 2 network. Systems can be a part of a VXLAN network as long as they are connected by IPv4 or IPv6 networks. In Oracle Solaris, VXLAN technology is implemented through its Elastic Virtual Switch (EVS) feature.

VXLAN provides isolated Layer 2 segment that is identified by the VXLAN segment ID or VXLAN network identifier (VNI). All VMs in the same VXLAN segment belong to the same virtual Layer 2 broadcast domain.

Communication in VXLANs is similar to that in isolated VLANs. Hence, only VMs that are in the same VXLAN segment can talk to each other.

VXLAN Naming Convention

In Oracle Solaris, a VXLAN endpoint is represented by a VXLAN datalink. This VXLAN datalink is associated with an IP address (IPv4 or IPv6) and a VXLAN network identifier (VNI). Even though multiple VXLAN datalinks can use the same IP address, the combination of the IP address and VNI must be unique. You can configure a VXLAN datalink with an optional multicast address, which is used for discovering the peer VXLAN endpoints on the same VNI and also to implement broadcast within a VXLAN segment. VXLAN datalinks in the same VNI must be configured with the same multicast address.

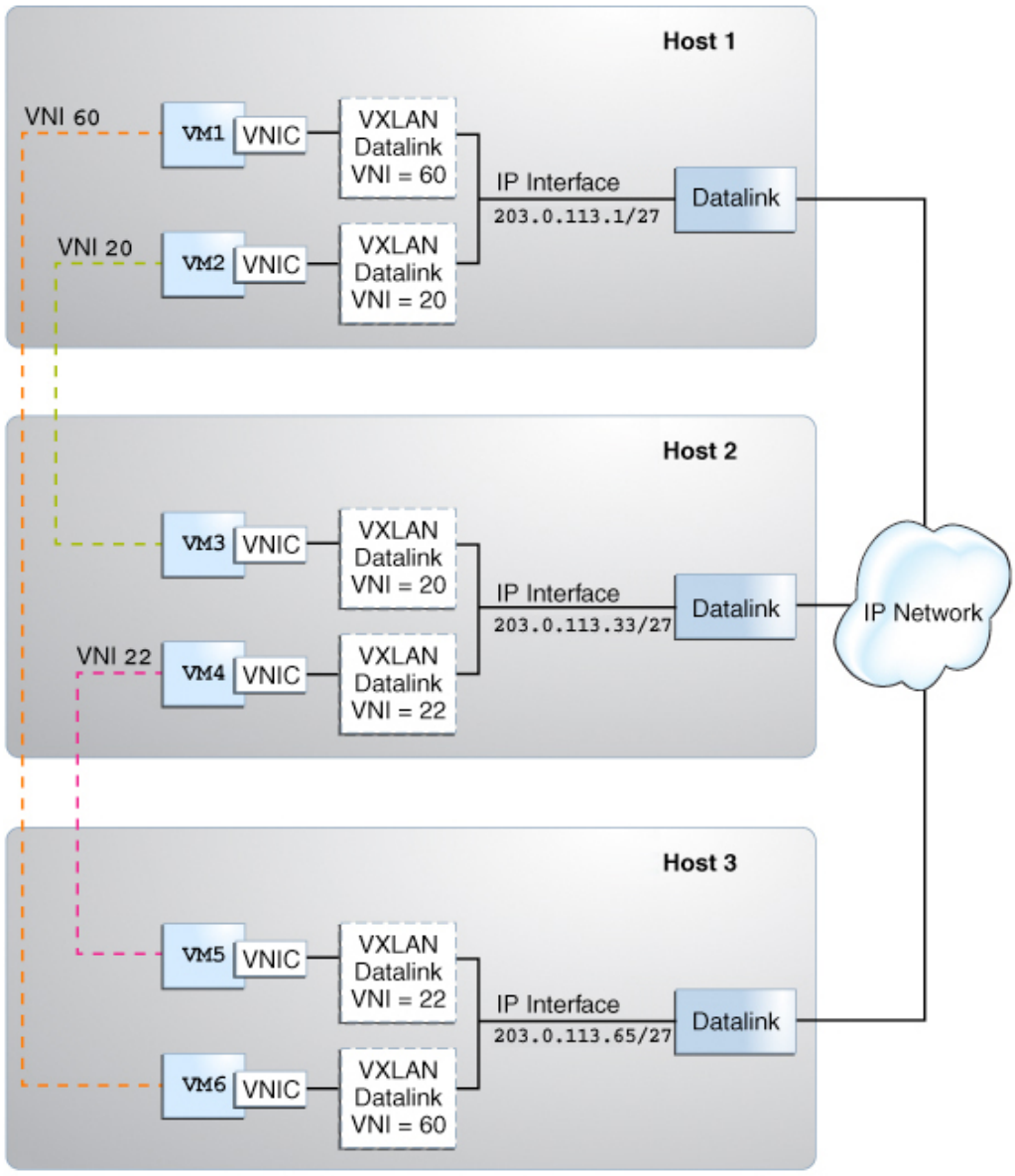
The convention for naming VXLAN datalinks is same as that for links or VLANs. For information about providing valid datalink names, see “Rules for Valid Link Names” in *Configuring and Managing Network Components in Oracle Solaris 11.4*.

VXLAN Topology

VXLAN enables you to organize systems on a Layer 3 network within their own VXLAN segments.

The following figure illustrates a VXLAN network that is configured over multiple systems.

FIGURE 9 VXLAN Topology

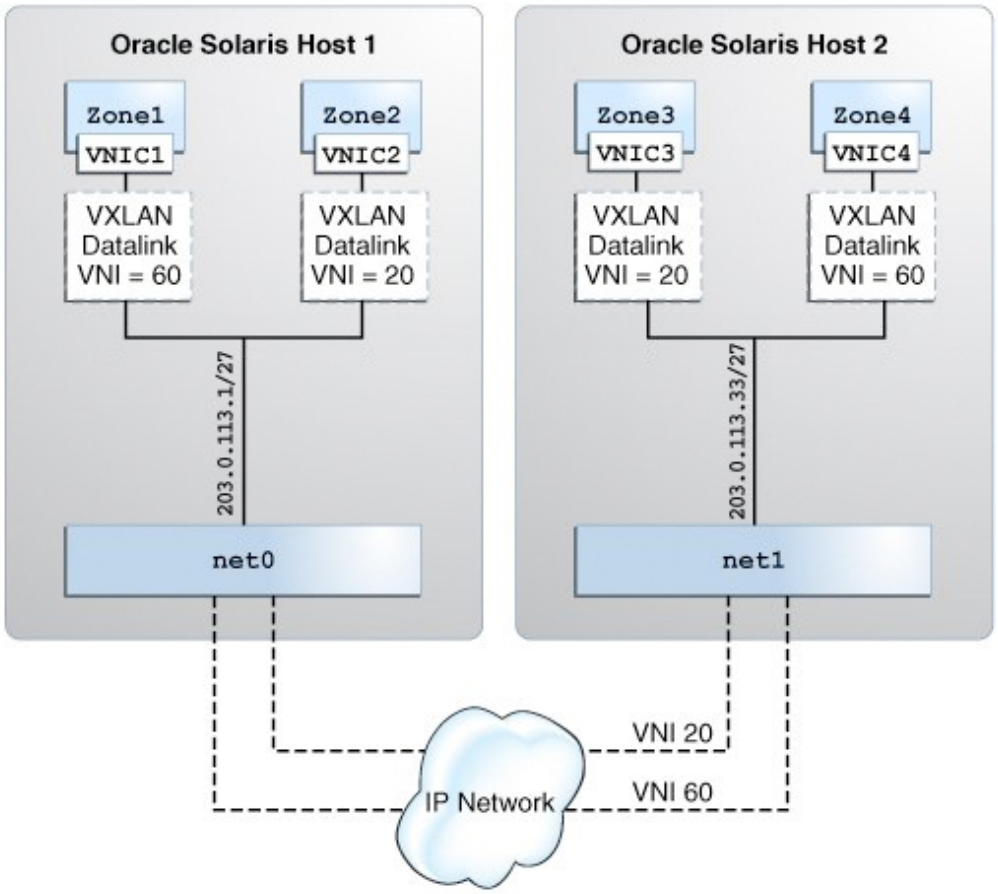


The figure shows three virtualized hosts attached to an IP network infrastructure. There are three VXLAN overlay networks identified by the VXLAN segment IDs or VNIs, `60`, `20`, and `22`. The VMs `VM1` and `VM6` are on the overlay network identified by the VNI `60`, the VMs `VM2` and `VM3` are on the overlay network identified by the VNI `20`, and the VMs `VM4` and `VM5` are on the overlay network identified by the VNI `22`.

Using VXLAN With Zones

The following figure shows two virtualized Oracle Solaris hosts attached to an IP network infrastructure with two VXLAN overlay networks identified by the VNIs, `20` and `60`.

FIGURE 10 VXLAN With Zones



To create zones that are a part of a VXLAN segment, use one of the following methods:

- Create a VNIC over a VXLAN and assign the VNIC to the zone. For more information, see "Configuring a VXLAN" on page 99.
- Assign the VXLAN as the underlying link for the zone's `anet` (VNIC) resource. For more information, see "Assigning a VXLAN to a Zone" on page 104.

In any case, the VNIC that is created in a zone is a part of a VXLAN segment identified by the underlying VXLAN datalink.

Assigning VNICs to VXLAN links is similar to creating a VLAN link and assigning it to a zone. See "How to Configure a VLAN" in *Managing Network Datalinks in Oracle Solaris 11.4*.

▼ How to Plan for a VXLAN Configuration

This procedure provides a general outline of the steps necessary to deploy a VXLAN network.

Before You Begin Verify the following:

- IP multicasting is supported on the network. If IP multicasting is not supported, VMs in the VXLAN cannot communicate with each other.
- If the VXLAN includes systems in different IP subnets, then multicast routing must be supported across the subnets. If multicasting routing is not supported, only the VMs over the VXLANs on the same IP subnet can communicate with each other and VMs over VXLANs on different IP subnets, for example, geographically dispersed data centers cannot communicate with each other.

Likewise, ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

Note - The destination UDP port used for VXLAN is IANA port 4789, in accordance with RFC 7348 (`https://tools.ietf.org/html/rfc7348`).

---

1. **Determine the virtual network topology in a physical network.**
   For example, if you are hosting a service that consists of several VMs on different systems, identify which VMs should belong to the VXLAN segment you want to create so that these can communicate with each other in the segment.
2. **Verify that the systems are connected through an IP interface.**
3. **Ensure that IP multicasting is enabled on the physical network.**
4. **Create a numbering scheme for the VXLAN segments.**
   For example, you can assign the VXLAN segments (VNIs) based on the application hosted by the VMs.
5. **Create a VXLAN datalink by specifying the IP address and the VXLAN segment ID.**
   Optionally, you can assign the VXLAN segments with their own multicast address.
6. **Create VNICs over VXLAN datalinks and assign the VNICs to zones.**

Alternatively, you can assign the VXLAN links as the underlying link for the zone's `anet` link.

Configuring a VXLAN

The following procedure assumes that the zones are already created on the system. For information about zone configuration, see Chapter 2, "Setting Up a Non-Global Zone" in *Creating and Using Oracle Solaris Zones*.

▼ How to Configure a VXLAN

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Determine the IP addresses that are available on the system.**

   ```
   $ ipadm show-addr
   ```

2. **Create the VXLAN datalink by specifying the IP address or IP interface.**

   - **To create the VXLAN by specifying the IP address:**

     ```
     $ dladm create-vxlan -p prop=value  vxlan-link
     ```

     `-p` *prop=value*
     : Specifies a comma-separated list of VXLAN datalink properties that can be set to the specified values on the VXLAN datalink that you create. You set the following properties:

       - `addr` – Specifies the IPv4 or IPv6 address for the VXLAN network. This address can be a specific address or a combination of address/prefix length.
       - `vni` – Specifies the network identifier of the VXLAN segment. You can specify a number between 0 and 16777215.
       - `mgroup` – (Optional) Specifies the multicast group name. You can specify this option only if the VXLAN segment has its own multicast group.

         If the multicast address is not specified, the VXLAN segment uses the `All Host` multicast address, which addresses all the hosts on the same network segment.

     vxlan-link
     : Name of the VXLAN.

- **To create the VXLAN by specifying the IP interface:**

  ```
  $ dladm create-vxlan -p prop=value
  ```

  -p *prop=value* — Specifies a comma-separated list of VXLAN datalink properties that can be set to the specified values on the VXLAN datalink that you create. You set the following properties:

  - `interface` – Specifies the IP interface for the VXLAN network.
  - `vni` – Specifies the network identifier of the VXLAN segment. You can specify a number between 0 and 16777215.

  VXLAN — Name of the VXLAN.

  When you specify the IP interface and the IP version, the VXLAN datalink is created over an available IP address of the version that is specified on that interface. For example, if you have an IP address `203.0.113.1` configured over `net0`, a VXLAN datalink is created over `203.0.113.1`. By default, an IP version is an IPv4 address. However, if you need an IPv6 address, you must specify the version by using the `ipvers` property.

Note - You can create VXLAN datalinks on IP addresses that are hosted on physical aggregated links (trunk or DLMP aggregation) or IPoIB links. However, you cannot create VXLAN datalinks on IP addresses hosted on IPMP, a virtual network interface, or loopback interfaces.

3. **Verify the VXLAN that you created.**

   ```
   $ dladm show-vxlan
   ```

4. **Create a VNIC over the VXLAN datalink.**

   ```
   $ dladm create-vnic -l vxlan-link vnic
   ```

   You can create VLAN VNIC over a VXLAN datalink. To create a VLAN VNIC, you must specify the `-f` (force) option. For information, see "How to Configure VNICs as VLANs" on page 40.

5. **Configure an IP interface over the VNIC directly or by assigning the VNIC to a zone first.**

   - **Configure an IP interface over the VNIC.**

     ```
     $ ipadm create-ip vnic
     ```

     ```
     $ ipadm create-addr -a address vnic
     ```

- **Assign the VNIC to a zone and configure an IP interface over the VNIC within the zone.**

  a. **Assign the VNIC with the zone's interface.**

  ```
  zonecfg:zone> add net
  zonecfg:zone:net> set physical=vnic
  zonecfg:zone:net> end
  ```

  b. **Verify and commit the changes that you have implemented and then exit the zone.**

  ```
  zonecfg:zone> verify
  zonecfg:zone> commit
  zonecfg:zone> exit
  ```

  c. **Reboot the zone.**

  ```
  global$ zoneadm -z zone reboot
  ```

  d. **Log in to the zone.**

  ```
  global$ zlogin zone
  ```

  e. **In the zone, create an IP interface over the VNIC that is now assigned to the zone.**

  ```
  zone$ ipadm create-ip interface
  ```

  f. **Configure the VNIC with a valid IP address.**

  If you are assigning a static address to the VNIC, you would type the following:

  ```
  zone$ ipadm create-addr -a address interface
  ```

  -a *address* Specifies the IP address, which can be in CIDR notation.

  g. **Exit the zone.**

For information about the `dladm` and `ipadm` commands, see the `dladm`(8) and `ipadm`(8) man pages.

Example 44 Creating a VXLAN and Configuring an IP Interface for the VNIC Created Over the VXLAN

This example shows the entire process of configuring a VXLAN. It includes displaying the information so you can see the progress of the configuration process.

```
$ ipadm show-addr net4
ADDROBJ   TYPE   STATE   ADDR
net4/v4   static ok      203.0.113.1/27

$ dladm create-vxlan -p addr=203.0.113.1/27,vni=10 vxlan1

$ dladm show-vxlan
LINK   ADDR            VNI  MGROUP
vxlan1 203.0.113.1/27  10   224.0.0.1

$ dladm show-link vxlan1
LINK   CLASS MTU  STATE OVER
vxlan1 vxlan 1440 up    --

$ dladm create-vnic -l vxlan1 vnic1

$ dladm show-vnic
LINK   OVER   SPEED  MACADDRESS      MACADDRTYPE  IDS
vnic1  vxlan1 10000  2:8:20:fe:58:d4 random       VID:0

$ ipadm create-ip vnic1
$ ipadm create-addr -T static -a local=203.0.113.34/27 vnic1/v4

$ ipadm show-addr vnic1
ADDROBJ  TYPE   STATE  ADDR
vnic1/v4 static ok     203.0.113.34/27
```

Example 45 Assigning the VNIC Created Over a VXLAN to a Zone and Configuring an IP Interface

This example assumes that you have completed steps 1 to 6 in Example 44, “Creating a VXLAN and Configuring an IP Interface for the VNIC Created Over the VXLAN,” on page 102.

After you create the VNIC, assign the VNIC to a zone and configure the IP interface.

```
global$ zonecfg -z zone2
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vnic1
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
```

```
global$ zoneadm -z zone2 reboot

global$ zlogin zone2
zone2$ ipadm create-ip vnic1
zone2$ ipadm create-addr -a 192.0.2.85/24 vnic1
ipadm: vnic1/v4

zone2$ exit
```

You have assigned the VNIC to a zone and then configured the IP interface over the VNIC.

Displaying VXLAN Information

You can use the `dladm show-link` command to view generic link information about VXLAN links. To view information that is specific to a VXLAN, use the `dladm show-vxlan` command.

```
$ dladm show-link
LINK                CLASS     MTU    STATE    OVER
net6                phys      1500   down     --
net0                phys      1500   up       --
net2                phys      1500   unknown  --
net3                phys      1500   unknown  --
net1                phys      1500   unknown  --
net5                phys      1500   unknown  --
net4                phys      1500   up       --
vxlan1              vxlan     1440   up       --
vnci1               vnic      1440   up       vxlan1

$ dladm show-vxlan vxlan1
LINK           ADDR            VNI   MGROUP
vxlan1         203.0.113.1     10    224.0.0.1
```

Deleting a VXLAN

To delete a VXLAN link, use the `dladm delete-vxlan` command. Before deleting a VXLAN link, you must ensure that there are no VNICs configured on that VXLAN link by using the `dladm show-link` command.

Become an administrator and issue the following command:

```
$ dladm delete-vxlan vxlan
```

For example, if you want to delete `vxlan1`, type the following command:

```
$ dladm delete-vxlan vxlan1
```

Assigning a VXLAN to a Zone

You can create zones that are a part of a VXLAN segment by assigning VXLAN as an underlying link to the zone's `anet` resource. For information about configuring a zone, see *Creating and Using Oracle Solaris Zones*.

▼ How to Assign a VXLAN to a Zone

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Determine the available IP addresses on the system.**

   ```
   $ ipadm show-addr
   ```

2. **Create the VXLAN by specifying the IP address.**

   ```
   $ dladm create-vxlan -p prop=value  VXLAN-LINK
   ```

3. **Verify the VXLAN that you created.**

   ```
   $ dladm show-vxlan
   ```

4. **Configure the zone by assigning the VXLAN that you created as the underlying link for the zone's `anet`.**

   ```
   global$ zonecfg -z zone
   zonecfg:zone2> add anet
   zonecfg:zone2:anet> set linkname=datalink
   zonecfg:zone2:anet> set lower-link=VXLAN-LINK
   zonecfg:zone2:net> end
   zonecfg:zone2> verify
   zonecfg:zone2> commit
   zonecfg:zone2> exit
   global$ zoneadm -z zone reboot
   ```

   VXLAN is assigned as the underlying link for the zone's `anet`.

Example 46 Assigning a VXLAN to a Zone's `anet`

```
$ ipadm show-addr net4
ADDROBJ   TYPE   STATE   ADDR
net4/v4   static ok      203.0.113.1/24 2

$ dladm create-vxlan -p addr=203.0.113.1,vni=10 vxlan1

$ dladm show-vxlan
LINK    ADDR         VNI  MGROUP
vxlan1  203.0.113.1  10   224.0.0.1
```

Because you have not specified a multicast address, this VXLAN segment uses the `All Host` multicast address, which addresses all the hosts on the same network segment.

```
$ dladm show-link vxlan1
LINK   CLASS MTU  STATE OVER
vxlan1 vxlan 1440 up    --
```

`vxlan1` is created and the link state is `up`.

```
global$ zonecfg -z zone2
zonecfg:zone2> add anet
zonecfg:zone2:anet> set linkname=net1
zonecfg:zone2:anet> set lower-link=vxlan1
zonecfg:zone2:anet> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global$ zoneadm -z zone2 reboot
```

`vxlan1` is assigned as the underlying link for the zone's `anet`.

When the zone boots up, `net1` is created in `zone2` over `vxlan1`.

Use Case: Configuring a VXLAN Over a Link Aggregation

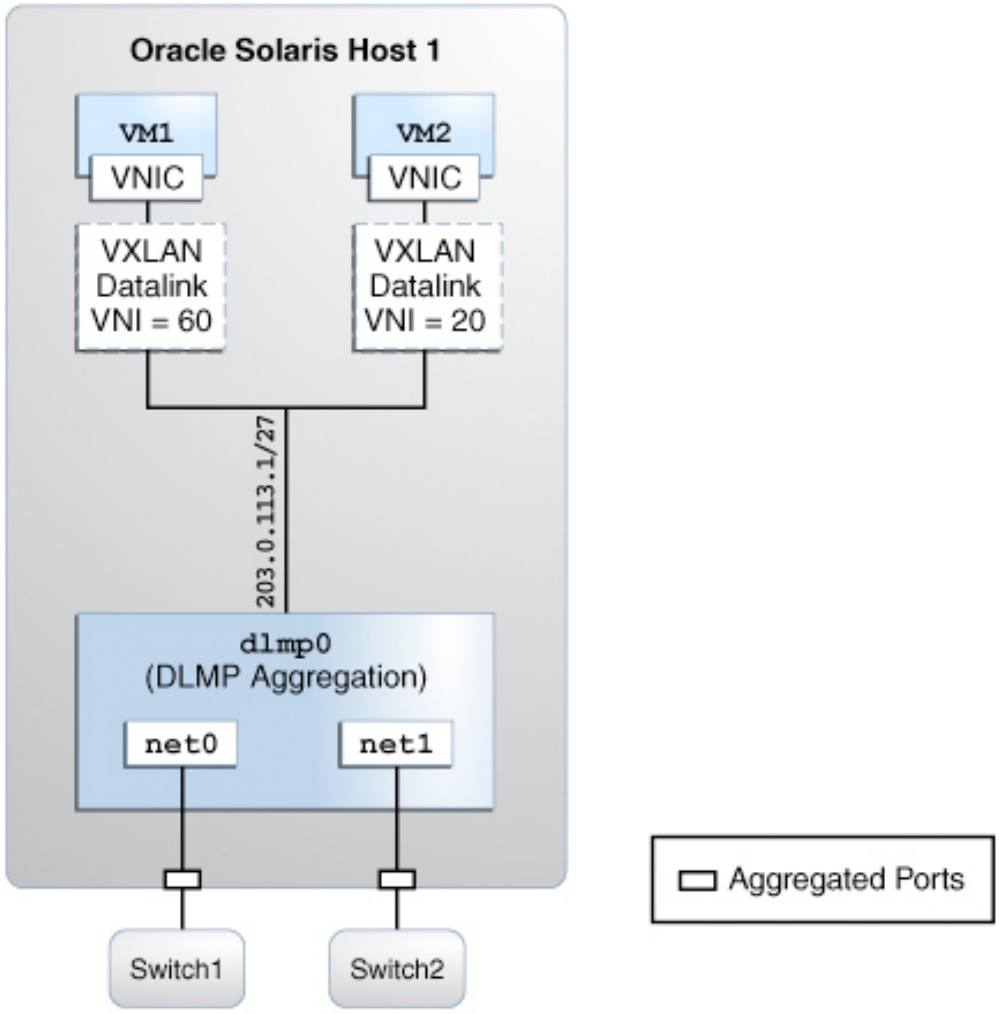
The following use case shows how to accomplish the following:

- Create a DLMP aggregation
- Configure an IP address over the aggregation
- Create two VXLANs over the aggregation
- Configure two zones with VXLAN datalinks as the lower links

For information about link aggregation, see Chapter 2, “Configuring High Availability by Using Link Aggregations” in *Managing Network Datalinks in Oracle Solaris 11.4*.

The following figure shows VXLAN configuration over a DLMP aggregation.

FIGURE 11 VXLAN Over a Link Aggregation



When an aggregated port or an external switch fails, VXLAN datalinks over the aggregation continue to exist as long as at least one port and a switch is functional, thereby providing

network high availability during failover. For example, if `net0` fails, then DLMP aggregation shares the remaining port `net1`, between VXLAN datalinks. The distribution among the aggregated ports occurs transparently to the user and independently of the external switches connected to the aggregation.

```
$ dladm show-link
LINK      CLASS     MTU     STATE   OVER
net0      phys      1500    up      --
net1      phys      1500    up      --
net2      phys      1500    up      --

$ ipadm show-if
IFNAME       CLASS        STATE     ACTIVE     OVER
lo0          loopback     ok        yes        --
net0         ip           ok        no         --
$ ipadm delete-ip net0

$ dladm create-vxlan -p addr=203.0.113.1,vni=20 vxlan20
$ dladm create-vxlan -p addr=203.0.113.1,vni=60 vxlan60

global$ zonecfg -z VM1
zonecfg:VM1> add anet
zonecfg:VM1:net> set linkname=net0
zonecfg:VM1:net> set lower-link=vxlan20
zonecfg:VM1:net> end
zonecfg:VM1> verify
zonecfg:VM1> commit
zonecfg:VM1> exit
global$ zoneadm -z VM1 reboot

$ dladm create-aggr -m dlmp -l net0 -l net1 dlmp0

$ ipadm create-ip dlmp0
$ ipadm create-addr -T static -a local=203.0.113.1 dlmp0/v4

global$ zonecfg -z VM2
zonecfg:VM2> add anet
zonecfg:VM2:net> set linkname=net0
zonecfg:VM2:net> set lower-link=vxlan60
zonecfg:VM2:net> end
zonecfg:VM2> verify
zonecfg:VM2> commit
zonecfg:VM2> exit
global$ zoneadm -z VM2 reboot
```

The `net0` and `net1` datalinks are aggregated into DLMP aggregation, `dlmp0` and an IP address `203.0.113.1` is configured for the aggregation. The VXLANs, `vxlan20` and `vxlan60` are created over the specified IP address `203.0.113.1`, which is configured for the aggregation. The

VXLAN, `vxlan20` is created in the VXLAN segment `20` and the VXLAN, `vxlan60` is created in the VXLAN segment `60`. The zone `VM1` is configured with the VXLAN datalink, `vxlan20` as the lower link and the zone `VM2` is configured with the VXLAN datalink, `vxlan60` as the lower link.

♦ ♦ ♦ CHAPTER 4

Using Edge Virtual Bridging

This chapter discusses the Edge Virtual Bridging (EVB) feature of Oracle Solaris that enables support for server-network edge virtualization. It contains the following topics:

Note - To configure edge virtual bridging and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 34.

EVB Support in Server-Network Edge Virtualization

In networks, the connection between a server port and its first hop switch port is called a server-network edge. At this connection point, network configurations such as VLANs and link aggregations must be the same on the server port and the switch port. For these types of configuration, the Data Center Bridging Capability Exchange (DCBX) can automate the configuration on both sides of the edge. See Chapter 8, "Managing Converged Networks by Using Data Center Bridging" in *Managing Network Datalinks in Oracle Solaris 11.4*.

However, with network virtualization, multiple virtual ports for virtual machines (VMs) lie behind the server port. Consequently, for network virtualization, the following requirements are added for the server-network edge, which the EVB feature addresses.

- Support for switching between the virtual machines through the external switch so that inter-VM traffic is subjected to policies configured on the switch

- Extension of virtual port properties into the network

In a virtualized server, packets between its VMs are looped back by the virtual switch within the host itself. Therefore, any policies that are configured on the external switch are not applied to inter-VM packets. With EVB, these inter-VM packets can be serviced by the external switch, which means that the switch can enforce policies on the inter-VM packets.

Additionally, EVB enables information exchange between VNICs and the switch. The switch can then automatically configure VNIC properties such as bandwidth limits, bandwidth shares, and MTU on the network. Without this feature, the server administrator and the network administrator must manually coordinate with each other to reconfigure the switch every time a VNIC is created, modified, or deleted on the server.

Overall, EVB helps increases efficiency in the use of networking resources. For example, the switch can enforce bandwidth limit on packets before they arrive at the host.

Reflective Relay

Reflective relay is the mechanism that enables VMs to communicate through the external switch. The switch itself must support the use of LLDP and its EVB type-length value (TLV) unit, which enables automatic configurations. Otherwise, you would have to manually configure reflective relay on the switch.

For information about how to manually configure reflective relay, refer to the switch manufacturer's documentation. See also “Controlling Switching Between VMs Over the Same Physical Port” on page 114. For more information about the LLDP TLV units, see “Information the LLDP Agent Advertises” in *Managing Network Datalinks in Oracle Solaris 11.4*.

Automated VNIC Configuration in the Network

Oracle Solaris uses the Virtual Station Interface Discovery and Configuration Protocol (VDP) defined in IEEE 802.1Qbg to exchange VNIC information with the switch. If the switch supports VDP, then VNIC properties are automatically configured on the switch. The switch can then apply VNIC settings to packets destined for that VNIC. The mechanism parallels DCBX which enables information exchange between host and switch about physical link properties.

See “Exchanging VNIC Information by Using VDP” on page 117.

Network and Server Efficiency Improvements by Using EVB

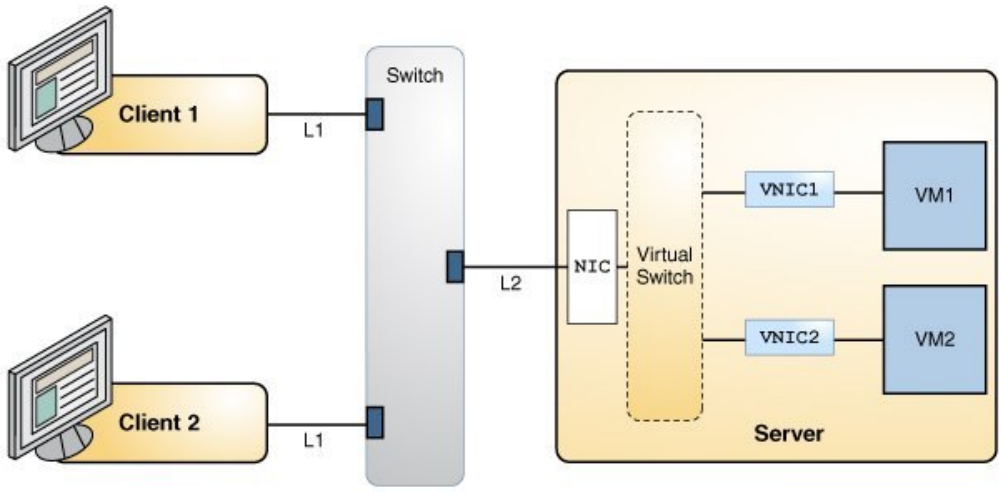
This section shows how server and network efficiency is increased through EVB.

Suppose that the server hosts two applications in a cloud environment.

- Applications are hosted separate virtual machines (VM1 and VM2), each of which has its corresponding VNICs (`VNIC1` and `VNIC2`, respectively).
- Client 1 and Client 2 can access the applications.
- VM1 and VM2 share the resources of the physical system and the bandwidth on link L2.
- Clients connect to the switch through the link L1. The switch is connected to the NIC by using the link L2.
- Predetermined SLA implements the following bandwidth usage on L2 for the VMs:
  - VM1 is running a high priority Transmission Control Protocol (TCP) service. VM1 has the maximum bandwidth limit of 8 Gbps.
  - VM2 is running a User Datagram Protocol (UDP) service that is not high priority. VM2 has the maximum bandwidth limit of 3 Gbps.

The following figure shows the applications hosted on a server.

FIGURE 12 Application Setup Without EVB



The following figure shows the same setup with EVB enabled:

FIGURE 13 Application Setup With EVB Enabled

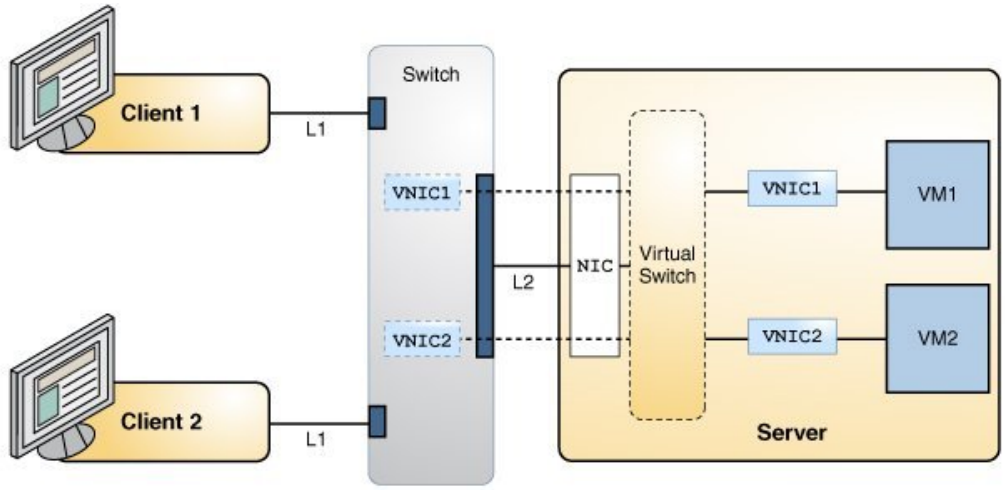


Table compares server efficiency between a network setup without EVB and one with EVB enabled.

TABLE 3 Efficiency of the Server Without EVB and With EVB

Server Efficiency Without EVB	Server Efficiency With EVB
The server regulates incoming traffic from the clients for bandwidth enforcement.	The switch regulates the traffic destined to the server.
System resources are used, thereby affecting the system and network performance.	System resources are not used to process the bandwidth.
When the clients need to utilize the services simultaneously, the clients use the bandwidth of link L2 and server resources. The server enforces the SLA on the VNICs for VM1 and VM2 to regulate the inbound and the outbound traffic of the clients. ■ Traffic from the clients (Client 1 and Client 2) use the bandwidth of link L2 without any restrictions. Also, if there is a bandwidth limit configured on the host, packets that use the bandwidth of L2 might be	■ SLA configured on the VNICs of the server are reflected on the switch. ■ Switch regulates the traffic towards VM1 and VM2 based on the configured bandwidth and therefore helps to utilize the bandwidth of link L2 appropriately. Because the switch regulates the bandwidth, the server does not have to process bandwidth on the receive side.

Server Efficiency Without EVB	Server Efficiency With EVB
dropped on the host, which results in inefficient use of the bandwidth. ■ VM1 provides a high priority TCP service and VM2 provides UDP service that is not high priority. Regulating VM1's bandwidth on the server causes TCP to respond, hence impacting VM1's use of bandwidth on the link L2. However, regulating VM2's service on the server does not impact its usage of the bandwidth of link L2. This affects other services using the link L2.	
Network traffic for UDP and TCP services inbound to the server uses the available bandwidth on the link L2 without any restrictions. After the server receives network traffic, it regulates the network traffic based on the configured bandwidth limit.	The configured bandwidth limits (3 Gbps and 8 Gbps) are regulated. Hence, the shared link L2's usage is based on the configured bandwidth limits.

How to Install EVB

To use EVB with its default configuration, you simply install the EVB package.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Verify whether the EVB package is installed.**

```
$ pkg info evb
```

2. **If the EVB package is not installed, install the package.**

```
$ pkg install evb
```

3. **Verify whether the service is enabled.**

```
$ svcs vdp
```

4. **If the service is not enabled, enable the service.**

```
$ svcadm enable vdp
```

The default EVB configuration is automatically enabled after EVB package installation. By accepting the default EVB configuration, the system can immediately exchange the information about any VNIC that you configure on the system with the external switch.

See Also

- "Exchanging VNIC Information by Using VDP" on page 117.

- "Displaying VDP and ECP State and Statistics" on page 119.
- "Changing the Default EVB Configuration" on page 120.
- Example 48, "Displaying EVB-Related Datalink Properties on a Physical Link," on page 123.

Controlling Switching Between VMs Over the Same Physical Port

The `virtual-switching` datalink property controls the switching between VMs over the same physical port through the following possible values:

- `local` – Enables the network traffic between VMs over the same physical NIC to be exchanged internally. This is the default mode.
- `remote` – Enables the network traffic between VMs over the same physical NIC to be exchanged through the external switch.
- `auto` – Uses LLDP to determine whether reflective relay is supported on the external switch. If reflective relay is supported on the external switch, network traffic between VMs is exchanged through the external switch. Otherwise, network traffic between VMs is exchanged internally.

The next sections explain each property value further.

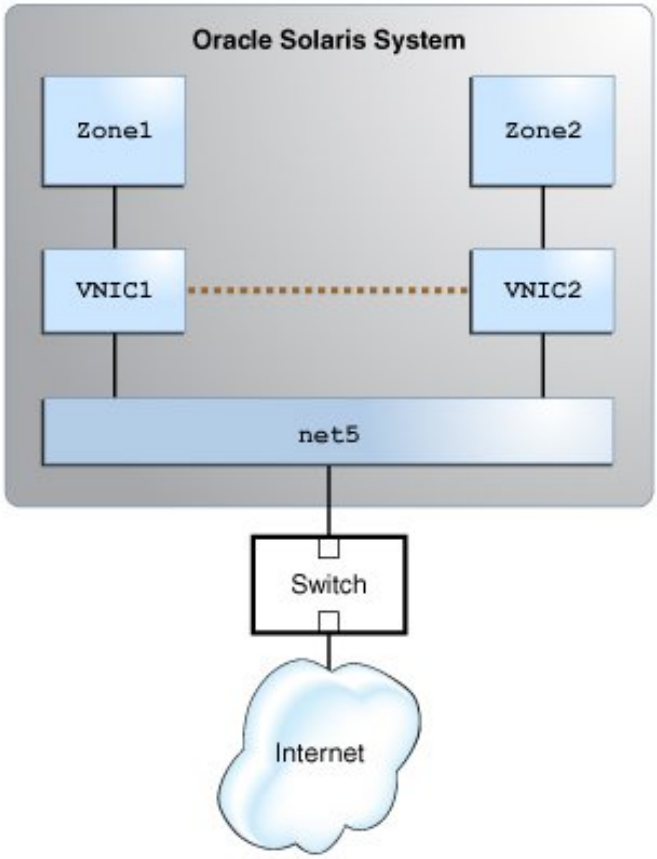
Local Network Traffic Exchange

The following figure shows a virtual network setup where the server is connected to a switch, but where network traffic is internal.

This scenario would be typical of a configuration where the VMs are running services for the same customer. Communication between the two zones can occur internally without any restrictions. In this case, the `virtual-switching` is set to `local`:

```
$ dladm show-linkprop -p virtual-switching net5
LINK  PROPERTY          PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net4  virtual-switching  rw    local  local      local    local,remote,auto
```

FIGURE 14 Internal Communication Between Zones



Remote Network Traffic Exchange

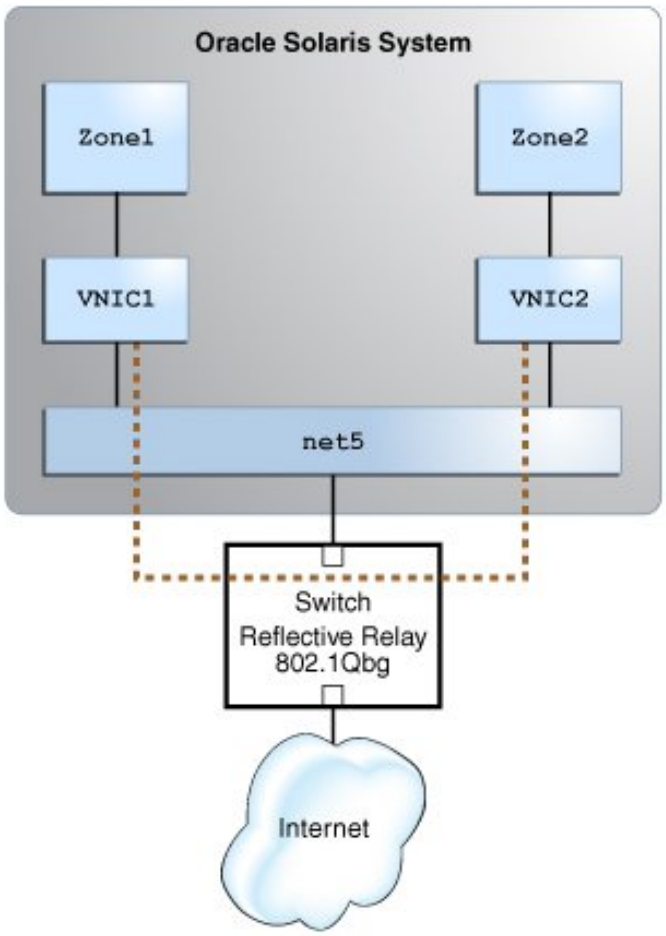
The following figure shows a virtual network setup where the server is connected to a switch, but where communication between VMs passes through the external switch.

This setup is deployed when different VMs run services for different customers. In such cases, the external switch is configured to control and isolate network traffic for the different VMs.

The example assumes that reflective relay is supported on the switch and properly configured before the `virtual-switching` property is set to `remote`:

```
$ dladm set-linkprop -p virtual-switching=remote net5
$ dladm show-linkprop -p virtual-switching net5
LINK  PROPERTY          PERM  VALUE   EFFECTIVE  DEFAULT  POSSIBLE
net5  virtual-switching  rw    remote  remote     local    local,remote,auto
```

FIGURE 15 Communication Between Zones by Using an External Switch



Auto Network Traffic Exchange

Setting the `virtual-switching` property to `auto` works depending on the ability of the switch to support reflective relay. If the support is present, then LLDP automatically configures network traffic exchange as either internal or external that uses the external switch.

To use the `auto` value, ensure the following first:

- The LLDP package is installed.

```
$ pkg info lldp
```

- The LLDP service is online.

```
$ svcs lldp
STATE          STIME    FMRI
online         Jul_13   svc:/network/lldp:default
```

- The `dot1-tlv` property is set to `evb` and the `mode` property is set to `both` for the NIC.

```
$ lldpadm show-agentprop -p mode,dot1-tlv net5
AGENT   PROPERTY   PERM   VALUE   DEFAULT   POSSIBLE
net5    mode       rw     both    disable   txonly,rxonly,both,disable
net5    dot1-tlv   rw     evb     none      none,vlanname,pvid,linkaggr,pfc,
                                             appln,evb,etscfg,etsreco,all
```

After the preceding requirements are met, you can use the `auto` option:

```
$ dladm set-linkprop -p virtual-switching=auto net5
$ dladm show-linkprop -p virtual-switching net5
LINK  PROPERTY           PERM  VALUE   EFFECTIVE  DEFAULT  POSSIBLE
net5  virtual-switching  rw    auto    remote     local    local,remote,auto
```

See Chapter 7, “Exchanging Network Connectivity Information With Link Layer Discovery Protocol” in *Managing Network Datalinks in Oracle Solaris 11.4*.

Exchanging VNIC Information by Using VDP

The VIS discovery and configuration protocol (VDP) enables the exchange of VNIC information, also called VSI, between the system (station) and the external switch or (bridge). VDP itself consists of VDP type-length value (TLV) units. The VDP TLV units are exchanged between station and bridge when you create or delete a VNIC.

The following components are involved in the VSI information exchange between server and switch:

- A VSI profile which consists of link properties that have been configured for the specific VNIC. A system can have as many VSI profiles as there are configured VNICs.
- A VSI identifier that uniquely identifies the profile. In Oracle Solaris, this identifier is the MAC address of the VNIC.
- The VSI Manager manages multiple VSI profiles on the system. It maps the VSI Type ID - VSI Version with a specific set of VNIC properties. In Oracle Solaris, the default VSI Manager is `oracle_v1` with a 3-byte encoding.
- A VSI Manager ID identifies the VSI Manager that is relevant to a specific VSI Type ID - VSI Version pair. Oracle Solaris has defined a default VSI Manager ID called `ORACLE_VSIMGR_V1`.

Note - Currently, there are no defined standards for defining a VSI profile and its specific properties. The definition of VSI types is vendor-specific and is closely linked to a VSI Manager ID.

The VSI Manager `oracle_v1` supports the following properties:

- Bandwidth limit
- Bandwidth share
- Link speed of the underlying link
- Maximum transmission unit (MTU) of the VNIC

By default, an Oracle Solaris host sends the following elements to the external switch:

- Oracle VSI Manager – `oracle_v1`
- VSI Type ID – VNIC properties encoded by using `oracle_v1` encoding
- VSI Version – Always `0`

The external switch is configured to support the Oracle VSI Manager (`oracle_v1`), which the switch uses to determine the properties encoded in the VSI Type ID. Then the switch applies the property settings to packets destined for the specific VNIC.

An Oracle organization-specific OUI TLV unit follows the VSI Manager ID TLV to indicate that it is the Oracle-specific VSI Manager ID. The absence of the Oracle specific TLV unit in the response from the switch indicates to the Oracle Solaris host that the switch does not support Oracle VSI Manager (encodings). Oracle Switch ES1-24 supports the Oracle VSI Manager, `oracle_v1`. For more information about configuration of EVB on Oracle Switch ES1-24, see "Configuring EVB" in *Sun Ethernet Fabric Operating System, EVB Administration Guide*.

Note - In addition to supporting the VDP and ECP protocols, to interoperate with Oracle Solaris system, external switches must also support `ORACLE_VSIMGR_V1`, which is the default Oracle VSI Manager ID, and the Oracle organizationally unique identifier (OUI) TLV (subtype `VDP_ORACLEOUI_VSIMGR_SUBTYPE`, which is used to carry the encoding information).

Displaying VDP and ECP State and Statistics

To display VDP information only for a single link, specify that link in the command. Otherwise, VDP information for all the Ethernet links is displayed.

Displaying the VDP State and Statistics

To display the VDP state, type the following command:

```
$ dladm show-ether -P vdp
VSI      LINK       VSIID               VSI-TYPE-ID  VSI-STATE   CMD-PENDING
vnic1    net0       2:8:20:22:3c:6b     98/0         ASSOC       NONE
vnic2    net0       2:8:20:90:7f:ef     96/0         ASSOC       NONE
```

`VSI-STATE` shows the status of the VDP exchange with the peer. Possible values are:

- `TIMEDOUT` – The peer has not responded to the VDP requests.
- `ASSOC` – The peer processed the request successfully.
- `DEASSOC` – Either the host or the peer has rejected the request. The peer can reject the request if it is not able to determine the profile or the properties specified. The host can reject the exchange of VDP packets if it is using `oracle_v1` encoding and the peer does not include the Oracle OUI in its response.

The sample output shows that two VSIs (VNICs) are configured over the link `net0`. Their specific VSI IDs refer to their respective MAC addresses. The `VSI-TYPE ID` for VNICs, `vnic1` and `vnic2` are generated from their respective properties (bandwidth limit and MTU) and the encoding is defined by `oracle_v1`.

To obtain statistics about the outgoing or incoming VDP packets, type the following command:

```
$ dlstat show-ether -P vdp net1
LINK    IPKTS    OPKTS   KeepAlives
net1    3        2       1
```

Displaying the Link Properties

Use the `dladm show-linkprop -p` command to display link properties. For example:

```
$ dladm show-linkprop -p max-bw,mtu vnic1
LINK     PROPERTY        PERM   VALUE     EFFECTIVE    DEFAULT   POSSIBLE
vnic1    max-bw          rw     100       100          --        --
vnic1    mtu             rw     1500      1500         1500      1500

$ dladm show-linkprop -p max-bw,mtu vnic2
LINK     PROPERTY        PERM  VALUE      EFFECTIVE    DEFAULT   POSSIBLE
vnic2    max-bw          rw    20         20           --        --
vnic2    mtu             rw    1500       1500         1500      1500
```

Displaying ECP State and Statistics

VDP uses ECP to exchange messages. The following example shows state of ECP that is specific to the physical link `net0`.

```
$ dladm show-ether -P ecp net0
LINK        MAX-RETRIES     TIMEOUT
net0        3               164
```

`MAX-RETRIES` Specifies the number of times ECP transmits a packet when it does not get an acknowledgement from the peer.

`TIMEOUT` Specifies the interval (in milliseconds) before retransmitting a packet. The time interval that ECP waits for an acknowledgment before retransmitting a packet.

To obtain the statistics for a physical link, type the following command:

```
$ dlstat show-ether -P ecp
LINK          IPKTS    OPKTS  IERRORS  OERRORS RETRANSMITS TIMEOUTS
net0          3        2      0        0       1            0
```

Changing the Default EVB Configuration

Typically, you do not need to change a default EVB configuration, which is generally sufficient as an operational EVB in a majority of cases. In the default configuration, the Oracle Solaris

VSI Manager ID, `ORACLE_VSIMGR_V1` automatically generates the VSI Type ID for the VNICs that you create. You do not need to set EVB-related datalink properties.

If you want to control and manage EVB configuration, then you must configure the following datalink properties that are related to EVB:

- `vsi-manager-id` – Specifies the VSI Manager ID that is set for a physical link or a VNIC. If this property is not set for a VNIC, the default value, `ORACLE_VSIMGR_V1`, of the underlying physical link is used.

  If you explicitly set the `vsi-manager-id` property, then you also need to explicitly set the VSI Type ID and VSI Version. In addition, you also need to explicitly configure these properties on the datalinks.

  Note - In Oracle Solaris, when you manually configure the VSI Manager ID, VSI Type ID, and VSI Version, the corresponding VNIC properties are not automatically configured.

- `vsi-manager-id-encoding` – Indicates the encoding that is associated with the VSI Manager ID. By default, this property is set to `oracle_v1`. If you do not want to associate `oracle_v1` with the VSI Manager ID, set this property value to `none`. When you set the value `none`, also make sure that you configure the VSI Manager ID, VSI Type ID, and VSI Version manually because they will not be automatically generated.
- `vsi-type-id` – Specifies a VSI Type ID. A VSI Type ID pairs with a VSI Version to be associated with a VSI profile. This 3-byte value is automatically generated if you use the default values for `vsi-manager-id` and `vsi-manager-id-encoding`. Otherwise, you must explicitly specify a value for this property.
- `vsi-version` – Specifies a VSI Version. The VSI Version pairs with a VSI Type ID to be associated with a VSI profile. This 1-byte value is automatically generated if you use the default values for `vsi-manager-id` and `vsi-manager-id-encoding`. Otherwise, you must explicitly specify a value for this property.

You can display EVB-related properties by using the `dladm show-linkprop` command. You can obtain the effective values of the VNIC-related link properties from their respective `EFFECTIVE` field values of the properties. For more information, see Example 48, “Displaying EVB-Related Datalink Properties on a Physical Link,” on page 123.

For more information about the EVB components, see “Exchanging VNIC Information by Using VDP” on page 117. For more information about EVB, see the evb(4P) man page.

How to Change the Default EVB Configuration

You must configure the `vsi-manager-id` and `vsi-manager-id-encoding` properties only on the physical link. The other EVB-related properties, such as `vsi-type-id` and `vsi-version`, must be configured on a VNIC.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Create a VNIC by using the datalink properties mentioned in the profile database.**

   $ **dladm create-vnic -l** *datalink* **-p max-bw=***max-bw-value***,priority=***priority-value* *VNIC*

2. **Set the encoding that is associated with the VSI Manager ID to `none` on the physical link because you are not using the default Oracle VSI Manager ID.**

   $ **dladm set-linkprop -p vsi-manager-id-encoding=none** *datalink*

3. **Set the VSI Manager ID on the physical link with an IPv6 address.**

   $ **dladm set-linkprop -p vsi-manager-id=***IPv6-address* *datalink*

4. **Set the VSI Type ID and VSI Version for the VNIC that you have created.**

   $ **dladm set-linkprop -p vsi-type-id=***VSI-Type-ID***,vsi-version=***vsi-version* *VNIC*

5. **Verify the properties that are set for the VNIC.**

   $ **dladm show-linkprop** *VNIC*

Example 47 Setting EVB-Related Datalink Properties

The following example shows how to set datalink properties that are related to EVB. This example uses a system with a profile that you can access by using an IPv6 address, `IP1`.

Assume that the VSI Manager ID, IP1 has the following profiles defined:

- VSI Type ID: `2`
- VSI Version: `1`
- Datalink properties: `max-bw=20`, `priority=5`

The VDP `ASSOC` TLV unit for `vnic1` contains the following information:

- VSI Manager ID = `IP1`
- VSI Type ID = `2`

- VSI Version = 1

```
$ dladm create-vnic -l net0 -p max-bw=20,priority=5 vnic1

$ dladm set-linkprop -p vsi-manager-id-encoding=none net0
$ dladm set-linkprop -p vsi-manager-id=IP1 net0
$ dladm set-linkprop -p vsi-type-id=2,vsi-version=1 vnic1

$ dladm show-linkprop vnic1
LINK     PROPERTY                PERM VALUE      EFFECTIVE  DEFAULT    POSSIBLE
...
vnic1    vsi-type-id             rw   2          2          --         --
vnic1    vsi-version             rw   1          1          --         --
vnic1    vsi-manager-id          rw   IP1        IP1        --         --
vnic1    vsi-manager-id-encoding rw   --         none       oracle_v1  none,oracle_v1
...
```

Example 48 Displaying EVB-Related Datalink Properties on a Physical Link

The following example displays EVB-related properties on the physical link.

```
$ dladm show-linkprop -p vsi-manager-id,vsi-manager-id-encoding net4
LINK     PROPERTY                PERM VALUE        EFFECTIVE    DEFAULT      POSSIBLE
net4     vsi-manager-id          rw   --           --           ::           --
net4     vsi-manager-id-encoding  rw   --           --           oracle_v1    none,
oracle_v1
```

The output displays the default configuration of EVB in Oracle Solaris. By using the `oracle_v1` encoding, the VSI Type ID and VSI version are automatically generated from the properties that are configured on the VNICs.

Example 49 Displaying EVB-Related Properties on a VNIC

The following example displays EVB-related properties on a VNIC.

```
$ dladm show-linkprop vnic0
LINK     PROPERTY                PERM VALUE      EFFECTIVE  DEFAULT    POSSIBLE
...
vnic0    vsi-type-id             rw   --         94         --         --
vnic0    vsi-version             rw   --         0          --         --
vnic0    vsi-manager-id          rw   --         ::         --         --
vnic0    vsi-manager-id-encoding  rw   --         oracle_v1  oracle_v1  none,oracle_v1
...
```

The output displays the effective encoding for `vnic0` as `oracle_v1`. In turn, the `EFFECTIVE` value for `vsi-type-id` `94` is automatically generated and effective for `vnic0`.

♦ ♦ ♦ CHAPTER 5

About Elastic Virtual Switches

The Elastic Virtual Switch (EVS) manages multiple virtual switches that are spread across several physical machines. This chapter provides an overview of this Oracle Solaris feature and includes the following topics:

Overview of the EVS Feature

Today's data centers or multitenant cloud environments present challenging tasks to administrators such as the following:

- Maintain multiple systems hosting several virtual machines (VMs) and manage their MAC and IP addresses.
- Administer different virtualization technologies such as VLANs and VXLANs.
- Ensure the internal and external connectivity of the VMs and at the same time provide isolation among virtual networks' tenants.
- Implement and enforce service-level agreements (SLAs).

In Oracle Solaris, EVS works with other technologies to meet these requirements. Through the EVS, system administrators can manage multiple systems through a single virtual switch.

About Virtual Switches

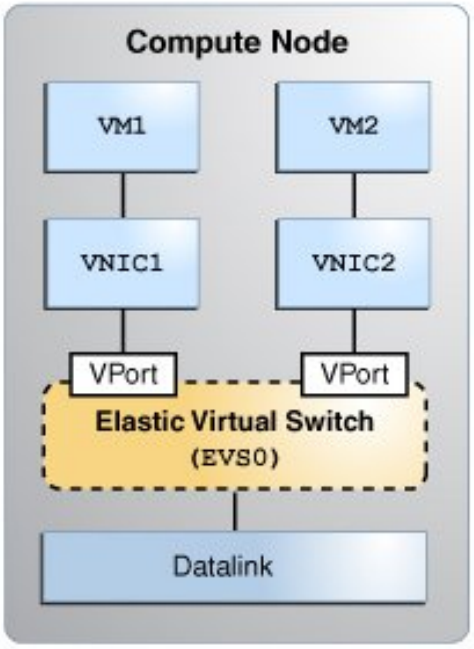
The virtual switch is an entity that facilitates communication between virtual machines. In Oracle Solaris, a virtual switch is implicitly created when you create a VNIC over a datalink

The virtual switch loops traffic between VMs within the physical machine and does not send this traffic out on the wire. To communicate with each other, VMs must be on the same Layer 2 segment. See “Virtual Switch” on page 25.

Beginning with Oracle Solaris 11.2, virtual switches are managed by EVS. As a network component, EVS can be configured with virtual ports, IP addresses and other properties to maintain and improve network operations. However, EVS does not replace the implicit virtual switch in previous Oracle Solaris releases.

The following figure shows the elastic virtual switch `EVS0` in a single compute node.

FIGURE 16 Elastic Virtual Switch in a Compute Node



The Oracle Solaris Elastic Virtual Switch

EVS is a virtual switch that spans one or more compute nodes and their VMs, hence its 'elastic' character. Through the switch, VMs connect to one another from anywhere in the network.

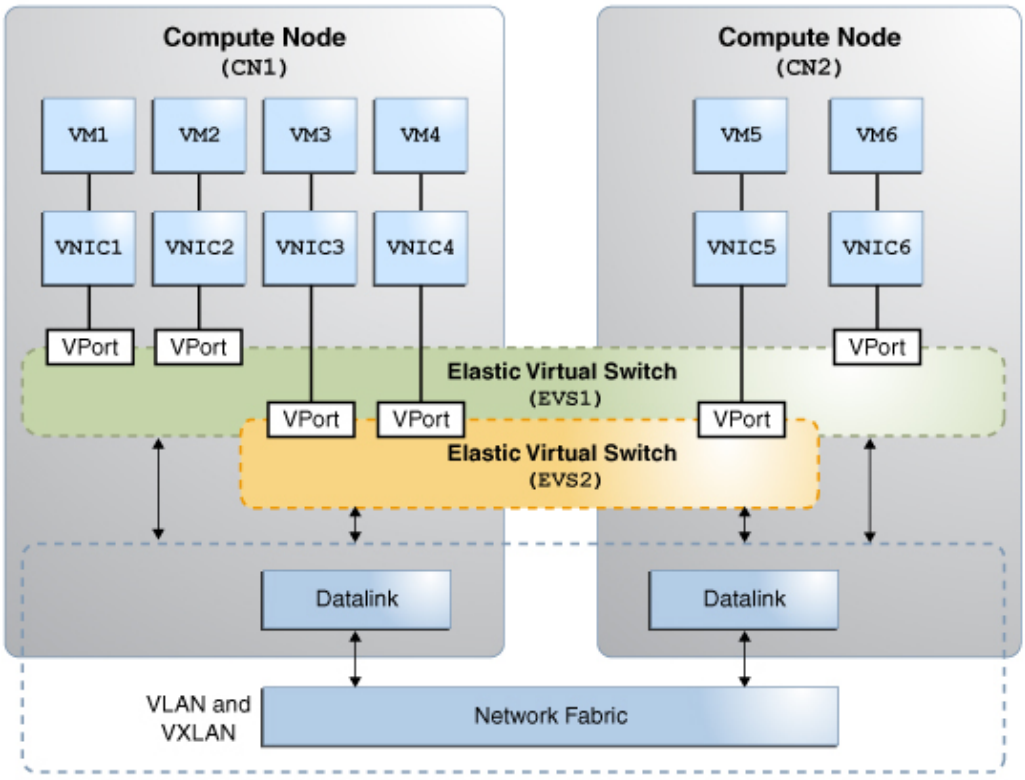
Note - In EVS, VMs specifically refer to Oracle Solaris Zones and Oracle Solaris Kernel Zones.

An EVS represents an isolated L2 segment. The isolation is implemented as a flat (untagged) network, a VLAN or a VXLAN. VLANs and VXLANs are used in an EVS setup that involves multiple nodes and network segments. A flat network suffices if the VM instances are on the same segment in the same node.

The following figure shows two elastic virtual switches (`EVS1` and `EVS2`) that span two compute nodes. Each node connects to the same network fabric through their datalinks. In the context of the cloud, datalinks are also called *uplink ports*.

VNICs connect to their switches through virtual ports. Through the switches, VMs can communicate with one another.

FIGURE 17 Elastic Virtual Switches Between Compute Nodes



Elastic Virtual Switch Resources

An elastic virtual switch is associated with both an IP network and a virtual port. These two resources and the EVS together are associated with a Universal Unique Identifiers (UUID). A UUID is automatically generated by the EVS controller when you create an elastic virtual switch or its resources.

IP Network

The IP network or a subnet, represents a block of IPv4 or IPv6 addresses with a default router for the block. You can associate only one subnet to an elastic virtual switch. All VMs that connect to the elastic virtual switch through a virtual port are assigned an IP address from the subnet. However, if preferred, you can manually assign an IP address to a VM's virtual port through one of the port's properties.

Virtual Port

A virtual port represents the point of attachment between the VNIC and an elastic virtual switch. When a VNIC connects to a virtual port, the VNIC inherits the network configuration parameters that the port encapsulates, such as the following:

- SLA parameters such as maximum bandwidth, class of service, and priority
- MAC address
- IP address

When you create a port, a randomly generated MAC address and the next available IP address from the associated subnet are assigned to the port. However, if preferred, you can specify the IP address and the MAC address through the port's properties.

You do not always need to add a virtual port to an elastic virtual switch. When a VNIC is created, you can specify only the name of the elastic virtual switch to which the VNIC must connect. In such cases, the EVS controller generates a system virtual port. These virtual ports follow the naming convention `sys-`*vportname*, for example, `sys-vport0`. The system virtual port inherits the elastic virtual switch properties.

The following table shows the port properties.

TABLE 4 Virtual Port Properties

Property	Description	Possible Values	Default Value
`cos`	Specifies the 802.1p priority on outbound packets on the VPort.	`0` - `7`	`--`
`maxbw`	Specifies the full-duplex bandwidth for the VPort.	`--`	`--`
`priority`	Specifies the relative priority for the VPort.	`high`, `medium`, or `low`	`medium`
`ipaddr`	Specifies the IP address associated with the virtual port.	`--`	If you do not specify the IP address for the

Property	Description	Possible Values	Default Value
	You can assign the IP address only when you create the VPort.		VPort, the EVS controller automatically selects an IP address from the IPnet associated with the elastic virtual switch.
`macaddr`	Specifies the MAC address associated with the VPort. You can assign the MAC address only when you create the VPort.	`--`	If you do not specify the MAC address for the VPort, the EVS controller generates a random MAC address for the VPort.
`evs`	A read-only property that represents the elastic virtual switch with which the VPort is associated.	`--`	`--`
`tenant`	A read-only property that represents the tenant with which the VPort is associated.	`--`	`--`
`protection`	Enables one or more types of link protection.	`mac-nospoof`, `ip-nospoof`, `dhcp-nospoof`, `restricted`, `none`	The default values are `mac-nospoof` and `ip-nospoof`. When you create a VNIC with a VPort, the `mac-nospoof` and `ip-nospoof` values are set by default for the VNIC. This prevents the VNIC from spoofing the other MAC and IP address.

About EVS Tenants

The elastic virtual switches and their resources are logically grouped together. Each logical group is called a *tenant*. The defined resources for the elastic virtual switch within a tenant are not visible outside that tenant's namespace. The tenant acts as a container to hold all the tenant's resources together.

If you do not specify a tenant, a default tenant `sys0global` is created and all the EVS operations occur in this namespace.

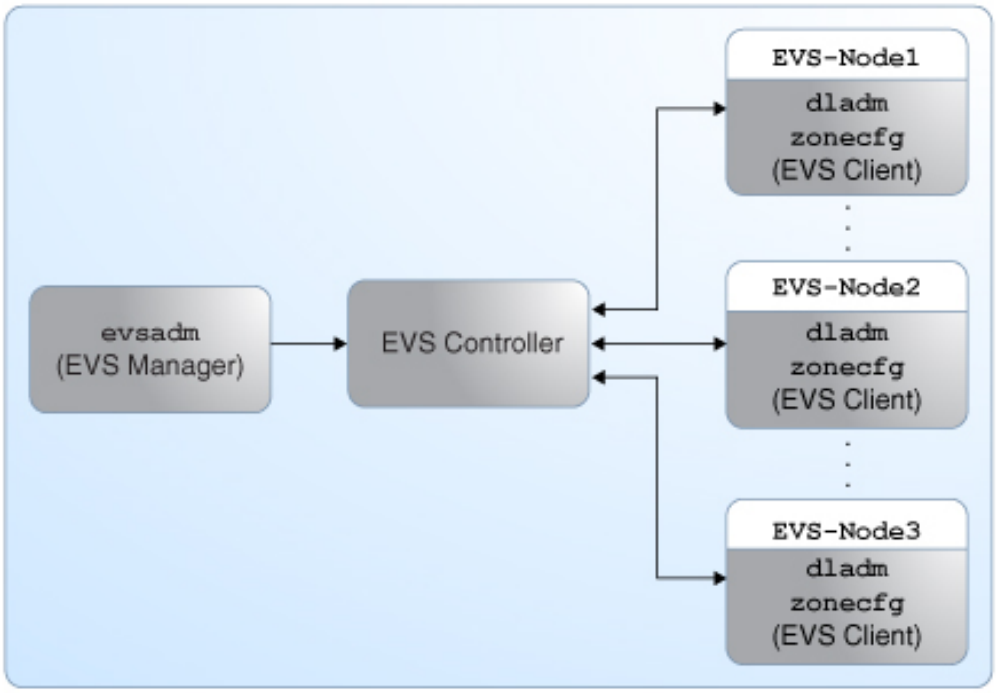
EVS Components

EVS has the following components:

- EVS manager
- EVS controller
- EVS clients
- EVS nodes

The following figure shows the components of EVS.

FIGURE 18 EVS Components



In this figure, the EVS manager and the EVS controller are two separate hosts. The EVS nodes `EVS-Node1`, `EVS-Node2`, and `EVS-Node3` are three hosts whose VNICs or zone's VNIC `anet` resources connect to an elastic virtual switch.

EVS Manager

The EVS manager is the entity that communicates with the EVS controller to define the L2 network topologies and the IP addresses for these L2 networks. The EVS manager communicates with the EVS controller through the `evsadm` command. The EVS manager and the EVS controller can also be on the same compute node.

Note - The L2 network topologies are the network segments and each segment forms a single broadcast domain, which is implemented by using VLANs or VXLANs.

EVS Controller

The EVS controller provides functionality for the configuration and administration of an elastic virtual switch and its associated resources associated. You must set up only one physical machine as the EVS controller in a data center or multitenant cloud environment.

To implement the L2 segments across physical machines, you would configure the properties of an EVS controller with information such as available VLAN IDs, available VXLAN segment IDs, or an uplink port for each EVS node.

Note - You can also push the EVS controller information to each of the EVS nodes in the data center or multitenant cloud environment by using SMF site profiles and the Auto Install (AI) service. For more information about SMF, see *Managing System Services in Oracle Solaris 11.4*. For more information about AI service, see "Working With Install Services" in *Automatically Installing Oracle Solaris 11.4 Systems*.

The following table lists the EVS controller properties.

TABLE 5 EVS Controller Properties

EVS Controller Property	Description	Possible Values	Default Value
`l2-type`	Defines how an elastic virtual switch is implemented across physical machines. **Note -** When you change the `l2-type` property, the elastic virtual switches that are created prior to change are not affected. Only the elastic virtual switches that are	`flat`, `vlan`, or `vxlan`	`vlan`

EVS Controller Property	Description	Possible Values	Default Value
	created after the change have the updated `l2-type` property. This behavior means that L2 segments based on Flat, VLAN, and VXLAN can coexist in an EVS controller.		
`vlan-range`	A comma-separated list of VLAN ID ranges that are used for creating an elastic virtual switch. One VLAN ID is associated with each elastic virtual switch.	`1 - 4094`	`--`
`vxlan-range`	A comma-separated list of VXLAN segment number ranges that are used for creating an elastic virtual switch. One VXLAN segment number is associated with each elastic virtual switch.	`0 - 16777215`	`--`
`vxlan-addr`	Specifies the IP address over which the VXLAN datalink must be created. You can also set the `vxlan-addr` property to a subnet.	`--`	`--`
`vxlan-mgroup`	Specifies the multicast address that you need to use while creating the VXLAN datalinks.	`--`	If you do not specify the multicast address, the VXLAN datalink uses the `All Host` address.
`vxlan-ipvers`	Specifies the IP version of the address that you need to use for the IP interface that hosts VXLAN datalinks.	`v4` or `v6`	`v4`
`uplink-port`	Specifies the datalink that you need to use for the network types: Flat, VLAN, or VXLAN.	`--`	`--`
`uuid`	Specifies an unique ID to identify an EVS controller in the data center or multitenant cloud environment. `uuid` is a read-only property whose value is automatically generated when you set up an EVS controller.	`--`	`--`
`uri-template`	Specifies the template from which the RAD URI scheme is computed by the EVS controller. The computed RAD URI is used between EVS controller and EVS nodes.	`ssh://` [*username*`@`] or `unix://` [*username*`@`]	`ssh://`

Controller properties are applicable to the entire data center or multitenant cloud environment. However, you can override the values of the controller properties `uplink-port` and `vxlan-addr` by specifying the host to which those property settings should apply. For example, the following syntax sets the uplink port `net1` as applicable only to `host1`:

```
$ evsadm set-controlprop -h host1 -p uplink-port=net1
```

EVS Clients

EVS clients are systems that are serviced by the EVS controller.

Clients can be physical systems or virtual machines (VMs), which in Oracle Solaris, are synonymous with non-global zones. The typical commands for configuring EVS clients are `dladm`, `evsadm` and `zonecfg`.

When VNICs are created for the elastic virtual switch, the configuration information for VNICs is retrieved from the EVS controller.

EVS Nodes

A connection between the elastic virtual switch and a VNIC constitutes a node. EVS clients are EVS nodes if their VNICs are configured and connected to the switch. You can use commands such as `dladm` and `zonecfg` to specify VNICs that need to be connected to an elastic virtual switch. For more information, see "Configuring EVS Nodes" on page 150.

Restrictions in Configuring EVS Components

If you deploy an EVS on your site, some restrictions are apply when you administer the following components on the switch:

- VNICs – When these are connected to the switch, then you cannot modify them, change their names, or change their properties.
- VXLANs – When you implement VXLANs on elastic virtual switches, VXLAN datalinks are automatically created on the EVS nodes that hosts VNICs for the EVS. These datalinks are identified as `evs-vxlan`*segment-ID*, such as `evs-vxlan200`, to indicate that EVS was the creator of the datalink.

  You cannot rename or delete these type of datalinks. However, you can temporarily set their properties by using the appropriate `dladm` command options.

Mandatory Packages for Using EVS

You need to install the following packages before using EVS:

- `pkg:/service/network/evs`

  You need to install the core package `pkg:/service/network/evs` on the EVS manager, EVS controller, and EVS nodes.

  When you install the `pkg:/service/network/evs` package, a new user, `evsuser` is created. The `evsuser` is a specific user with the Elastic Virtual Switch Administration rights profile. This profile provides all the required authorizations and privileges to perform EVS operations.

- `pkg:/system/management/rad/module/rad-evs-controller`

  You need to install this package only on the system that acts as an EVS controller. You must use only one controller to manage all the elastic virtual switches in a data center or multitenant cloud environment.

Security Requirements for Using EVS

In a multitenant EVS setup, individual tenants cannot manage their own elastic virtual switches and their resources because per-tenant user authorization for each user is not supported. The entire EVS domain must have a single administrator who manages resources of all the tenants.

The role that this administrator assumes must have the appropriate rights profile as described in "Using Rights Profiles to Perform Network Configuration" on page 34.

As a preferred alternative, you can use `evsuser` to perform EVS-related tasks. This user is created automatically when you install the `pkg:/service/network/evs` package. `evsuser` has all the required authorizations to configure EVS as well as observe EVS resources and statistics. For more details, see "About SSH Authentication and the `evsuser`" on page 138.

♦ ♦ ♦ CHAPTER 6

Administering Elastic Virtual Switches

This chapter describes tasks for administering elastic virtual switches and their resources. It contains the following topics:

Note - Becoming `evsuser` is the recommended approach to administer elastic virtual switches and issue commands described in this chapter. This user is described further in "About SSH Authentication and the `evsuser`" on page 138. You might need additional rights profiles to issue other privileged commands indirectly related to EVS configuration. See "Using Rights Profiles to Perform Network Configuration" on page 34.

Overview of Elastic Virtual Switch Configuration

Configuring the elastic virtual switch (EVC) involves the following general steps:

1. Preparing to set up EVS.
   a. Installing the mandatory packages on the all the hosts containing the EVS controller, EVS manager, and EVS nodes.
   b. Setting up the SSH authentication with the preshared public key for `evsuser` among all the components in the EVS setup.
2. Configuring the EVS controller.

3. Creating and configuring EVS switches.
4. Configuring the EVS nodes.

To configure EVS, the main command to use is `evsadm`. Refer to the `evsadm`(8) man page for details about its subcommands and options. See also the `dladm`(8) and `zonecfg`(8) man pages.

Preparing to Set Up EVS

This section describes two tasks in the preparation for configuring EVS.

Installing the Required Packages

Begin your EVS network setup by installing the required EVS packages.

1. On the host where the EVS controller will reside, do the following:
   a. Install the required packages.

      ```
      controller$ pkg install evs
      controller$ pkg install rad-evs-controller
      ```

   b. Restart the local RAD service.

      ```
      controller$ svcadm restart rad:local
      ```

2. On all the other nodes, install the EVS package.

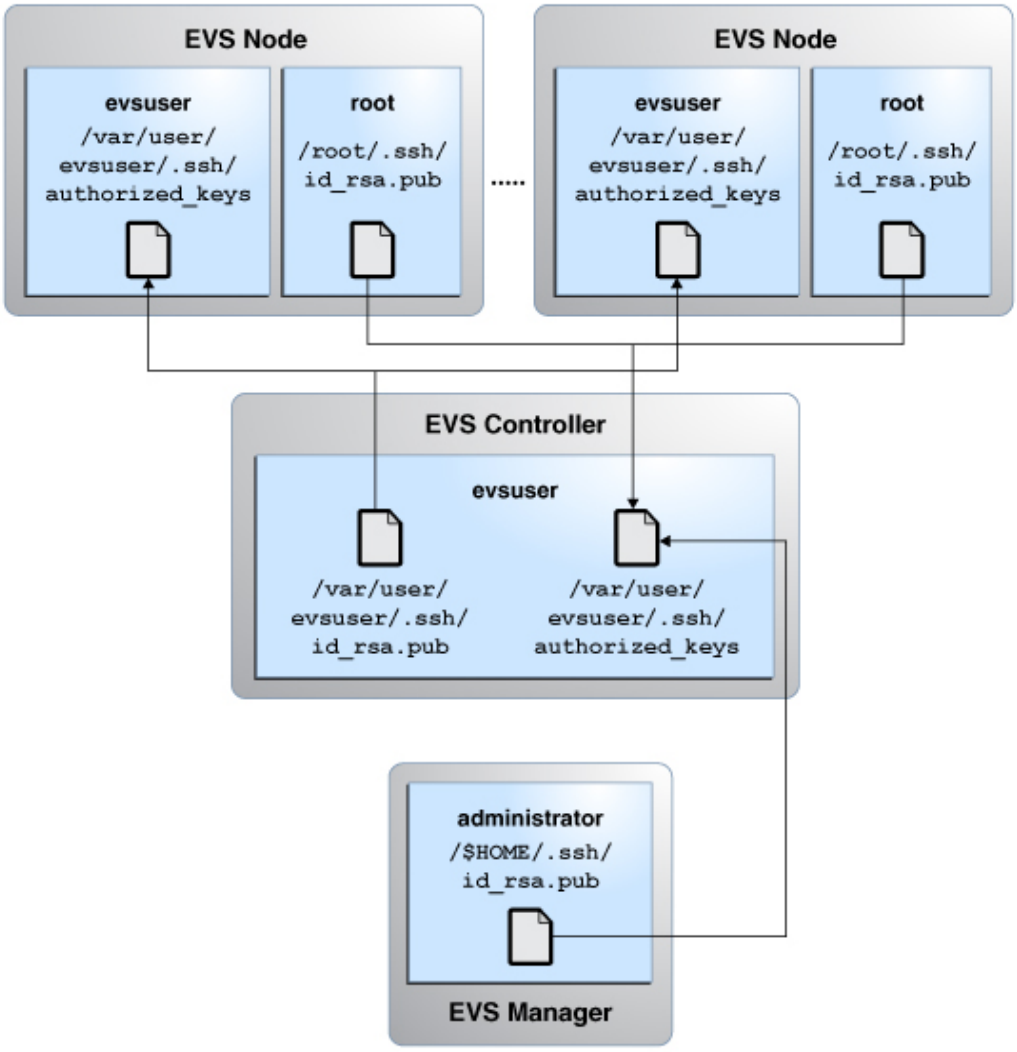
   ```
   evs-node$ pkg install evs
   ```

About SSH Authentication and the `evsuser`

For flexibility in configuring all EVS nodes, the `evsuser` is created automatically when you install the EVS package. This user is assigned the Elastic Virtual Switch Administration rights profile for configuring EVS. By becoming that user, you obtain the necessary authorization to perform the procedures contained in this chapter.

To use `evsuser`, you need to configure SSH authentication first to enable connectivity between the controller and the different nodes. You must perform authentication on **all** the nodes in the EVS setup. The following figure illustrates the authentication process:

FIGURE 19 SSH Authentication in the EVS Setup



How to Set Up SSH Authentication Between the EVS Controller and an EVS Node

Before You Begin Install the mandatory packages. See "Installing the Required Packages" on page 138.

1. **On the EVS controller, become `evsuser`.**

```
evs-controller$ su - evsuser
```

2. **Generate a RSA key pair in the EVS controller for `evsuser`.**

```
evsuser@evs-controller$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/var/user/evsuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/user/evsuser/.ssh/id_rsa.
Your public key has been saved in /var/user/evsuser/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1e:17:04:7d:bf evsuser@evs-controller
```

3. **Copy the public key from the controller's `/var/user/evsuser/.ssh/id_rsa.pub` file to an EVS node's `/var/user/evsuser/.ssh/authorized_keys` file.**

Note - You must perform this step for every EVS node that is in the EVS setup.

How to Set Up SSH Authentication Between an EVS Node and the EVS Controller

Note - You must perform this procedure on every EVS node in the EVS setup.

Before You Begin Install the mandatory packages. See "Installing the Required Packages" on page 138.

1. **On the EVS node, generate a RSA key pair.**

```
evs-node$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
```

```
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1d:17:04:7d:bf root@evs-node
```

2. **Copy the public key from the `/root/.ssh/id_rsa.pub` file in the EVS node to the `/var/user/evsuser/.ssh/authorized_keys` file in the EVS controller.**

▼ How to Set Up SSH Authentication Between the EVS Manager and the EVS Controller

You perform this procedure if the EVS Manager and EVS Controller reside on different hosts.

Before You Begin Install the mandatory packages. See "Installing the Required Packages" on page 138.

1. **On the EVS manager, generate a RSA key pair.**

```
evs-manager$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1d:17:04:7e:bf root@evs-manager
```

2. **Copy the public key from the `/root/.ssh/id_rsa.pub` file in the EVS manager to the `/var/user/evsuser/.ssh/authorized_keys` file in the EVS controller.**

Testing the SSH Authentication

To check whether SSH authentication completed successfully for all the nodes, issue the following commands from the specific nodes:

- Between controller and EVS node

  From the controller, connect to the node and issue this command:

```
evsuser@evs-controller$ ssh evsuser@evs-node
The authenticity of host 'evs-node (192.0.2.20)' can't be established.
RSA key fingerprint is 73:66:89:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'evs-node' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:40:28 2014 from evs-node
Oracle Corporation      SunOS 5.11      11.2    April 2014
evsuser@evs-node$
```

The output shows that you can log in to the EVS node as `evsuser` without a password from the EVS controller.

- Between EVS node and controller

  From the EVS node, connect to the controller and issue this command:

```
evs-node$ ssh evsuser@evs-controller
The authenticity of host 'evs-controller (192.0.2.10)' can't be established.
RSA key fingerprint is 73:66:81:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-controller' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:36:28 2014 from evs-controller
Oracle Corporation      SunOS 5.11      11.2    April 2014
evsuser@evs-controller$
```

  The output shows that you can log in to the EVS controller as `evsuser` without a password from the EVS node.

- Between EVS manager and EVS controller

  From the EVS manager, connect to the controller and issue this command:

```
evs-manager$ ssh evsuser@evs-controller
The authenticity of host 'evs-controller (192.0.2.10)' can't be established.
RSA key fingerprint is 73:66:81:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-controller' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:38:28 2014 from evs-controller
Oracle Corporation      SunOS 5.11      11.2    April 2014
evsuser@evs-controller$
```

  The output shows that you can log in to the EVS controller as `evsuser` without a password from the EVS manager.

Configuring the EVS Controller

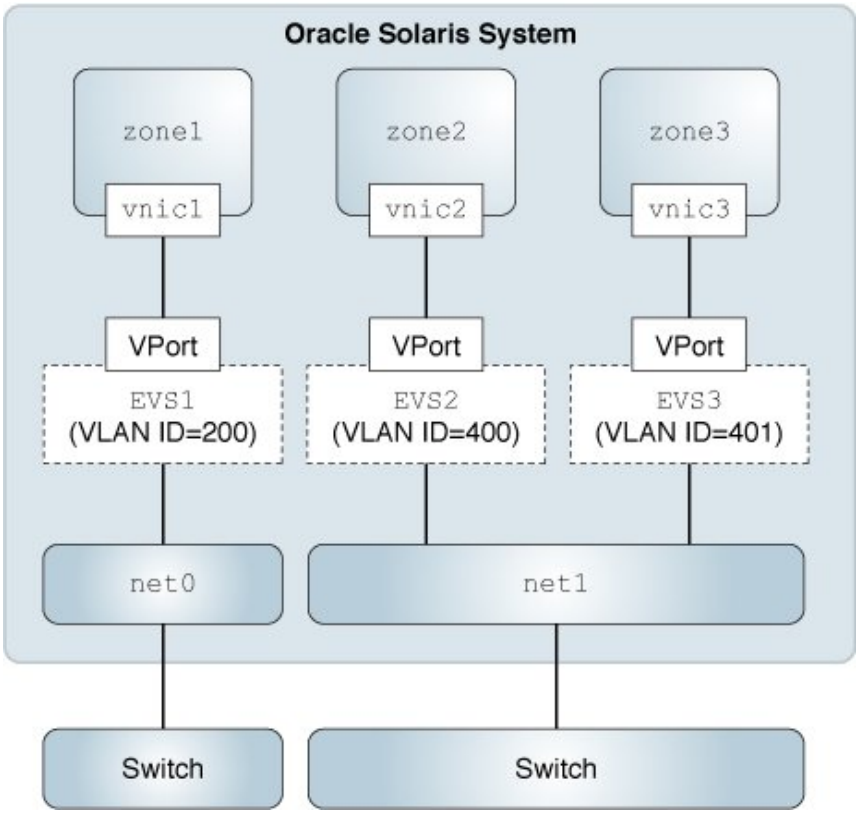
This section discusses important controller properties that you need to set.

About Layer 2 Segment Types

One priority group of controller properties to configure pertains to the type of layer 2 network segment you want to implement. For example, for a pure VLAN setup, you would only configure VLAN-related properties and disregard VXLAN properties. The amount of configuration also depends on the number of uplink ports.

The following figure shows a setup of 3 VLANs with their respective VLAN IDs and two uplink ports (`net0` and `net1`).

FIGURE 20 Multiple Uplink Ports in a Host



A different setup might use only pure VXLAN, or combine both VLANs and VXLANs in the configuration.

About Setting RAD Per-EVS Node Connections

To determine how RAD connections are established between the controller and the nodes, you set the `uri-template` property. The property's default value is `ssh://`.

If all the EVS nodes are on the same physical system, then using local connection is sufficient. You would set the property as follows:

```
$ evsadm set-controlprop -p uri-template=unix://
```

In a multinode environment, you should use SSH to connect. Further, because `evsuser` has been provided specifically for EVS configuration, then specify that user for RAD connections.

```
$ evsadm set-controlprop -p uri-template=ssh://evsuser
```

▼ How to Configure an EVS Controller

This procedure assumes that the steps are performed by you as `evsuser`. The `evsuser` has the necessary authorizations to perform EVS configuration and to run the `evsadm` command.

Before You Begin The following must be completed:

- Required EVS packages are installed. See "Installing the Required Packages" on page 138.
- SSH authentication for `evsuser` is completed. See "About SSH Authentication and the `evsuser`" on page 138.

1. **Specify the connection information for the controller.**
   If you are on a separate node from the controller, issue the following command to enable you to configure the EVS controller remotely.

   ```
   $ evsadm set-prop -p controller=value
   ```

   The property value can be one of the following:

   - `ssh://evsuser@evs-controller-`*host-name*
   - `ssh://evsuser@evs-controller-`*IP-address*
   - `unix://`

Specify this setting if all the EVS components are hosted on the same system.

2. **Become `evsuser` and connect to the controller.**

```
$ su - evsuser
evsuser@host ssh controller
```

3. **Configure controller properties.**

   a. **Set the Layer 2 type and the uplink ports.**

```
evsuser@controller$ evsadm set-controlprop [-h host] \
   -p l2-type=value,uplink-port=value
```

-h *host*	Host name for which the properties are being set.
l2-type	Layer 2 segment type, which can be `vxlan`, `vlan`, or `flat`.
uplink-port	Datalink to which EVS nodes connect. You can configure multiple uplink ports.

   b. **Configure other properties relevant to the `l2-type` setting.**

```
evsuser@controller$ evsadm set-controlprop -p property=value[,property=value,…]
```

   For example, for a VXLAN L2 segment, you would also configure `vxlan-range`, `vxlan-addr`, and so on.

   To list all configurable properties, type the command `evsadm show-controlprop`. Or, refer to Table 5, “EVS Controller Properties,” on page 132.

   See the examples for configuring specific L2 segment types.

4. **(Optional) Display the configuration information.**

```
evsuser@controller$ evsadm show-controlprop [-p property]
```

Example 50 Configuring an EVS Controller With a VXLAN L2 Segment Type

The following example shows how to configure the host `s11-server` as the EVS controller, whose L2 segments are created by using a VXLAN. Note that the displayed controller properties are only an extract of the entire output.

The example assumes that you have already specified the controller connections and you are connected to `s11-server` as `evsuser`.

```
evsuser@s11-server$ evsadm set-controlprop -p l2-type=vxlan
evsuser@s11-server$ evsadm set-controlprop -p vxlan-range=10000-20000
evsuser@s11-server$ evsadm set-controlprop -p vxlan-addr=192.0.2.0/24
evsuser@s11-server$ evsadm set-controlprop -h s11-server -p uplink-port=net3
evsuser@s11-server$ evsadm set-controlprop -h s11-client -p uplink-port=net4

evsuser@s11-server$ evsadm show-prop
PROPERTY            PERM   VALUE                      DEFAULT
controller          rw     ssh://evsuser@s11-server   --

evsuser@s11-server$ evsadm show-controlprop
PROPERTY            PERM VALUE                   DEFAULT     HOST
l2-type             rw   vxlan                   vlan        --
uplink-port         rw   --                      --          --
uplink-port         rw   net3                    --          s11-server
uplink-port         rw   net4                    --          s11-client
uri-template        rw   ssh://                  ssh://      --
uuid                r-   b3fda654-c14c-11e4      --          --
                         -ae16-5f67bed8a8e9
vlan-range          rw   --                       --         --
vlan-range-avail    r-   --                       --         --
vxlan-addr          rw   192.0.2.0/24 0.0.0.0     --
vxlan-ipvers        rw   v4                       v4         --
vxlan-mgroup        rw   0.0.0.0                  0.0.0.0    --
vxlan-range         rw   10000-20000              --         --
vxlan-range-avail   r-   10000-20000              --         --
```

Example 51 Configuring an EVS Controller With a VLAN L2 Segment Type

The following example shows how to configure the controller in `s11-server` to host a VLAN L2 segment.

The example assumes that you have already specified the controller connections and you are connected to `s11-server` as `evsuser`.

```
evsuser@s11-server$ evsadm set-controlprop -p l2-type=vlan
evsuser@s11-server$ evsadm set-controlprop -p vlan-range=200-300,400-500
evsuser@s11-server$ evsadm set-controlprop -p uplink-port=net2
evsuser@s11-server$ evsadm set-controlprop -h host2.example.com -p uplink-port=net3
evsuser@s11-server$ evsadm set-controlprop -h host3.example.com -p uplink-port=net4
```

Example 52 Configuring an EVS Controller With a Flat L2 Segment Type

This example shows how to configure a flat EVS network. All the EVS components are on `s11-server`. Note that in this example, the flat L2 type segment is simultaneously configured with the uplink port by specifying `flat=yes`.

The example assumes that you have become evsuser.

```
evsuser@s11-server$ evsadm set-prop -p controller=unix://
evsuser@s11-server$ evsadm set-controlprop -p uplink-port=net4,flat=yes

evsuser@s11-server$ evsadm show-controlprop -p uplink-port
PROPERTY          VALUE       FLAT VLAN_RANGE  VXLAN_RANGE  HOST
uplink-port       net4        yes   --         --           --

evsuser@s11-server$ evsadm create-evs -p l2-type=flat evs0

evsuser@s11-server$ evsadm show-evsprop -p l2-type
EVS     TENANT        PROPERTY   PERM VALUE     DEFAULT      POSSIBLE
evs0    sys-global    l2-type    r-   flat      --           --

evsuser@s11-server$ evsadm show-evs -L
EVS           TENANT        L2TYPE VID  VNI
evs0          sys-global    flat   --   --
```

Configuring Elastic Virtual Switches

Before creating elastic virtual switches, understand your virtual topology first. Determine the L2 segments you need, network related information such as the subnet and the default router, and virtual port configuration.

An EVS controller can support multiple switches. Thus, in a switch configuration, you can specify property settings that differ from what have been defined for the controller.

▼ How to Configure an Elastic Virtual Switch

Before You Begin The following must be completed:

- Required EVS packages are installed. See “Installing the Required Packages” on page 138.
- SSH authentication for evsuser is completed. See “About SSH Authentication and the evsuser” on page 138.
- The controller node is configured. See “Configuring the EVS Controller” on page 142.

1. **If you are on a different host than the controller, do the following:**

a. If you have not done so on this host yet, specify the controller connection information.

```
$ evsadm set-prop -p controller=value
```

See Step 1 of *How to Configure an EVS Controller* for an explanation of this step.

b. Become `evsuser` and then connect to the controller.

2. Create an elastic virtual switch.

```
evsuser@controller$ evsadm create-evs [-T tenant-name] \
   [-p {prop=value[,...]}[,..]] EVS-switch-name
```

`-T` *tenant-name*	Name of the tenant. If unspecified, the default tenant is the global zone and assigned the name `sys-global`.
`-p` *prop*	Comma-separated list of other properties such as `maxbw` and `priority`. Properties that you set are inherited by the virtual ports connected to this switch, unless you configure those virtual ports differently. For a list of a switch's configurable properties, type `evsadm show-evsprop`.
EVS-switch-name	Name of the elastic virtual switch.

3. Add a subnet to the switch.

```
evsuser@controller$ evsadm add-ipnet [-T tenant-name] \
   -p subnet=value[{,prop=value[,...]}[,...]] EVS-switch-name/IPnet-name
```

You must specify a subnet. Other properties, including the tenant, are optional.

For a list of a subnet's configurable properties, type `evsadm show-ipnetprop`.

4. (Optional) Add a virtual port to the switch.

```
evsuser@controller$ evsadm add-vport [-T tenant-name] \
    [-p {prop=value[,...]}[,...]] EVS-switch-name/VPort-name
```

`-p` *prop* specifies a comma-separated list of VPort properties. For a list of Vport properties, see Table 4, “Virtual Port Properties,” on page 129. Or, type `evsadm show-vportprop`.

Typically, you do not need to add a virtual port. When you configure a VNIC to connect to a switch, the EVS controller generates a system virtual port. These virtual ports follow the

naming convention `sys-`*vportname*, such as `sys-vport0`. The system virtual port inherits the elastic virtual switch properties.

5. **(Optional) Display the configured elastic virtual switch.**

```
evsuser@controller$ evsadm
```

Example 53 Configuring an Elastic Virtual Switch for the Global Tenant

The following example shows how to configure an elastic virtual switch with an IP subnet and a virtual port.

```
evsuser@controller$ evsadm create-evs ORA
evsuser@controller$ evsadm add-ipnet -p subnet=192.0.2.2/27 ORA/ora_ipnet
evsuser@controller$ evsadm add-vport ORA/vport0
evsuser@controller$ evsadm
NAME          TENANT        STATUS VNIC         IP               HOST
ORA           sys-global    idle   --           ora_ipnet        --
   vport0     --            free   --           192.0.2.2/27     --
```

Example 54 Configuring an Elastic Virtual Switch for a Tenant

The following example shows how to create the elastic virtual switch `ORA` with a subnet and a virtual port for a tenant.

```
evsuser@controller$ evsadm create-evs -T tenantA ORA
evsuser@controller$ evsadm add-ipnet -T tenantA -p subnet=192.0.2.0/27 ORA/ora_ipnet
evsuser@controller$ evsadm add-vport -T tenantA ORA/vport0
evsuser@controller$ evsadm
NAME          TENANT        STATUS VNIC         IP               HOST
ORA           tenantA       idle   --           ora_ipnet        --
   vport0     --            free   --           192.0.2.2/27     --
```

Example 55 Configuring a Tenant's Switch With an Address Pool

The following example shows how to add the IPnet `ora_ipnet` to `ORA`. In this example, you restrict the block from which the IP address is automatically allocated to a VPort. The IP address is allocated from the specified pool of IP addresses instead of the entire subnet.

```
evsuser@controller$ evsadm add-ipnet -T ABC -p subnet=192.0.2.0/27, \
   pool=192.0.2.10-192.0.2.15,192.0.2.20-192.0.2.25 ORA/ora_ipnet
evsuser@controller$ evsadm show-ipnetprop -p pool ORA/ora_ipnet
NAME          TENANT PROPERTY  PERM VALUE                   DEFAULT   POSSIBLE
ORA/ora_ipnet ABC    pool      rw   192.0.2.10-192.0.2.15,   --        --
                                    192.0.2.20-192.0.2.25
```

In this example, the IP addresses that are allocated to the virtual ports are within the pools `192.0.2.10-192.0.2.15` and `192.0.2.20-192.0.2.25`. The number of virtual ports must not exceed the available addresses in the pool.

Configuring EVS Nodes

EVS hosts that connect to the EVS switch are also called compute nodes. The connection is established through the host's VNIC. The host itself can contain virtual machines which connect to the switch through VNIC anets. Thus, zones can become EVS nodes as well.

▼ How to Set Up the Connection Between an EVS Node and the Controller

This procedure describes steps to configure a node to become part of the EVS topology. It contains steps that also involve non-global zone configuration. The information related to non-global zone is limited to what is relevant to EVS setups. For complete details about how to create, install, and configure zones, see *Creating and Using Oracle Solaris Zones*.

Before You Begin The following must be completed:

- Required EVS packages are installed. See "Installing the Required Packages" on page 138.
- SSH authentication for `evsuser` is completed. See "About SSH Authentication and the `evsuser`" on page 138.
- The controller node is configured. See "Configuring the EVS Controller" on page 142.
- The elastic virtual switch is configured. See "How to Configure an Elastic Virtual Switch" on page 147.

The host itself can already contain existing zones. Or, you can simultaneously create the zone and configure it to connect to the switch.

1. **Create a temporary VNIC over the elastic virtual switch.**

```
evsnode$ dladm create-vnic -t -c EVS-switch-name[/VPort-name] [-T tenant-name] VNIC-name
```

`-t` Specifies that the VNIC is temporary, which is required for VNICs created for EVS.

`-c` *EVS-switch-name*[*/VPort-name*]	Name of the elastic virtual switch to which the VNIC will connect. If you do not specify the VPort name, the system automatically generates a VPort and assigns the VPort to the VNIC. After the VNIC is connected to an elastic virtual switch, the VNIC either inherits the properties from the specified elastic virtual switch or VPort.
`-T` *tenant-name*	Name of the tenant that owns the EVS. If the tenant is not specified, then the system assumes the owner is the default `sys-global` tenant.
VNIC-name	The name of the VNIC.

2. (Optional) After you have created the VNIC, obtain the VNIC configuration information with this command:

```
evsnode$ dladm show-vnic -c
```

The `-c` option displays the information specifically about VNICs connected to an elastic virtual switch.

3. If you are using a non-global zone, configure its `anet` resource.

You use the `zonecfg` command to perform this step. Refer to the example for the command usage.

The `anet` resource has the following properties to set for a zone:

`tenant`	Specifies the name of the tenant. If a value is not specified when configuring a zone, the system assigns the default value, `sys-global` tenant.
`vport`	Specifies the name of the virtual port. If a value is not specified when configuring a zone, a system VPort is automatically generated for the elastic virtual switch and the port inherits the elastic virtual switch properties.
`evs`	Specifies the name of an elastic virtual switch to which you must connect the `anet` resource.

Example 56 Creating a VNIC `anet` Resource for an Elastic Virtual Switch

This example shows how to create a zone with a VNIC `anet` resource to connect to the switch and has the following details:

- Zone: `evszone`

- Tenant: `tenantA`
- Elastic Virtual Switch: `ORA`
- Host's datalink: `net1`
- VNIC `anet` for the zone: `evszone/net1`

```
evsnode$ zonecfg -z evszone
Use 'create' to begin configuring a new zone
zonecfg:evszone> create
create: Using system default template 'SYSdefault'
zonecfg:evszone> set zonepath=/export/zones/evszone
zonecfg:evszone> set tenant=tenantA
zonecfg:evszone> add anet
zonecfg:evszone:net> set evs=ORA
zonecfg:evszone:net> set vport=vport0
zonecfg:evszone:net> end
zonecfg:evszone> exit
evsnode$ zoneadm -z evszone install
evsnode$ zoneadm -z evszone boot
evsnode$ zlogin -C evszone
evsnode$ dladm show-vnic -c
LINK          TENANT   EVS  VPORT   OVER  MACADDRESS       IDS
evszone/net1  tenantA  ORA  vport0  net2  2:8:20:89:a1:97  VID:200
```

When `evszone` boots, the VNIC `anet` `evszone/net1` is associated with the MAC address, IP address, and SLA properties of the VPort `ORA/vport0`. For more information about configuring a zone's VNIC `anet` resources for an elastic virtual switch, see "Use Case: Configuring an EVS Network Topology" on page 171.

Administering Elastic Virtual Switches, IPnets, and VPorts

This section describes how to administer an elastic virtual switch, an IPnet, and a VPort. For more information about how to configure an elastic virtual switch, IPnet, and VPort, see "Configuring Elastic Virtual Switches" on page 147.

Administering an Elastic Virtual Switch

This section describes how to perform the following tasks for an elastic virtual switch:

- Displaying information about an elastic virtual switch
- Setting properties for an elastic virtual switch
- Displaying elastic virtual switch properties

Displaying Elastic Virtual Switch Information

You use the `evsadm show-evs` command to display elastic virtual switch information. The command syntax is:

`$ evsadm show-evs [-f {`*fname*`=`*value*`[,...]}[,...]] [-L] [[-c] -o` *field*`[,...]] [`*EVS-switch-name*`]`

`-f` {*fname*=*value*[,...]} [,...]
: A comma-separated name-value pair used to filter the output (row selection). If multiple filters are specified, then the displayed output is a result of an AND operation among the filters. If the filter value is multivalued, then the displayed output is a result of an OR operation among the filter values. The supported filters are:

- `tenant`
- `evs`
- `host`
- `ipnet`
- `vport`

`-L`
: Displays the L2 segment associated with an elastic virtual switch. Additionally, the VLAN IDs or VXLAN segment IDs associated with an elastic virtual switch is displayed.

`-o` *field*[,...]
: Specifies a case-insensitive, comma-separated list of output fields to display. You can specify the following fields, which appear as columns in the output:

  `all`
  : Displays all the output fields.

  `EVS`
  : Name of the elastic virtual switch.

  `TENANT`
  : Name of the tenant that owns the elastic virtual switch.

  `STATUS`
  : Status of the elastic virtual switch, whether it is idle or busy. The elastic virtual switch is busy if it has at the least one VPort that has a VNIC connected to it.

  `NVPORTS`
  : Number of virtual ports associated with the elastic virtual switch.

IPNETS	The list of IP networks associated with the EVS. Currently only one IP network can be associated with an elastic virtual switch.
HOST	The list of hosts that the elastic virtual switch spans across multiple systems.

EXAMPLE 57 Displaying Elastic Virtual Switch Information

The following example displays information for the elastic virtual switch ORA.

```
$ evsadm show-evs ORA
EVS           TENANT        STATUS NVPORTS IPNETS      HOST
ORA           sys-global    busy   1       ora_ipnet   s11-client
```

The following example displays the VLAN ID associated with the elastic virtual switch ORA.

```
$ evsadm show-evs -L
EVS           TENANT      L2TYPE  VID  VNI
ORA           tenantA     VLAN    200  --
```

The output shows the following information:

EVS	Name of the elastic virtual switch
TENANT	Name of the tenant that owns the elastic virtual switch
L2TYPE	Type of the L2 network
VID	VLAN ID used to implement the elastic virtual switch
VNI	VXLAN segment ID used to implement the elastic virtual switch

Setting Properties for an Elastic Virtual Switch

You use the `evsadm set-evsprop` command to set properties for an elastic virtual switch. The command syntax is:

```
$ evsadm set-evsprop [-T tenant-name] -p prop=value[,...] EVS-switch-name
```

-p *prop*	Sets the values of a property on the specified elastic virtual switch. EVS supports the following properties:

- `maxbw` – Sets the full-duplex bandwidth for all the virtual ports that connect to the specified elastic virtual switch. The bandwidth is specified as an integer with a scale suffix (`K`, `M`, or `G` for Kbps, Mbps, and Gbps). If no units are specified, the input value is read as Mbps. The default is no bandwidth limit.
- `priority` – Sets the default priority for all the virtual ports that connect to the specified elastic virtual switch. The possible values are `high`, `medium`, or `low`. The default value is `medium`. The priority is not reflected in any protocol priority fields on the wire but is used for packet processing scheduling within the system. A VPort with a high priority offers a better latency depending on the availability of system resources.

EXAMPLE 58 Setting Properties for an Elastic Virtual Switch

This example shows how to set properties for the elastic virtual switch `ORA`.

```
$ evsadm set-evsprop -p maxbw=200 ORA
$ evsadm set-evsprop -p priority=high ORA
```

Displaying Properties of an Elastic Virtual Switch

You use the `evsadm show-evsprop` command to display the properties of an elastic virtual switch. The command syntax is:

```
$ evsadm show-evsprop [-f {fname=value[,...]}[,...] [[-c] -o field[,...]] \
[-p prop[,...]] [EVS-switch-name]
```

`-f {`*fname*`=`*value*`[,...]} [,...]`

A comma-separated name-value pair used to filter the output (row selection). If multiple filters are specified, then the displayed output is a result of an AND operation among the filters. If the filter value is multivalued, then the displayed output is a result of an OR operation among the filter values. The supported filters are:

- `tenant` – Filter the elastic virtual switch properties by the tenant name
- `evs` – Filter the elastic virtual switch properties by the elastic virtual switch name
- `host` – Filter the elastic virtual switch properties by the host name

Example 59, “Displaying Elastic Virtual Switch Properties,” on page 156 shows output based on the filter value.

Note - You can filter elastic virtual switches by using their property values. See Example 67, "Displaying VPort Properties," on page 164.

`-o` *field*[,...] Specifies a case-insensitive, comma-separated list of output fields to display. You can specify the following fields, which appear as columns in the output:

`all`	Displays all the output fields.
`EVS`	Name of the elastic virtual switch.
`TENANT`	Name of the tenant that owns the elastic virtual switch.
`PROPERTY`	Name of the elastic virtual switch property.
`PERM`	The read or write permissions of the property. The value shown is either `r-` or `rw`.
`VALUE`	The current property value. If the value is not set, it is shown as `--`. If the value is unknown, it is shown as `?`.
`DEFAULT`	The default value of the property. If the property has no default value, `--` is shown.
`POSSIBLE`	A comma-separated list of possible values for the property. If the possible values are unknown or unbounded, `--` is shown.

EXAMPLE 59 Displaying Elastic Virtual Switch Properties

The following example displays the properties configured for the elastic virtual switch `ORA`.

```
$ evsadm show-evsprop ORA
EVS      TENANT        PROPERTY  PERM VALUE     DEFAULT   POSSIBLE
ORA      sys-global    maxbw     rw   200       --        --
ORA      sys-global    priority  rw   high      medium    low,medium,high
ORA      sys-global    tenant    r-   --        --        --
```

The following example displays the output for the elastic virtual switches `HR` and `ORA`. In this example, the `evs` filter is specified to obtain the output for elastic virtual switches `HR` and `ORA`.

```
$ evsadm show-evsprop -f evs=HR,ORA
EVS         TENANT        PROPERTY  PERM VALUE     DEFAULT   POSSIBLE
HR          tenantA       maxbw     rw   300       --        --
HR          tenantA       priority  rw   --        medium    low,medium,high
HR          tenantA       tenant    r-   --        --        --
ORA         sys-global    maxbw     rw   --        --        --
ORA         sys-global    priority  rw   --        medium    low,medium,high
ORA         sys-global    tenant    r-   --        --        --
```

EXAMPLE 60 Displaying the UUID of an Elastic Virtual Switch

This example shows how to display the UUID of the elastic virtual switch `evs1`.

```
$ evsadm show-evsprop -p uuid -o evs,tenant,property,perm,value evs1
EVS       TENANT        PROPERTY   PERM VALUE
evs1      sys-global    uuid       r-   5c5b7120-95cc-11e4-ab91-171c32874415
```

Administering an IPnet Configuration

This section describes how to perform the following tasks for an IPnet after you add an IPnet for an elastic virtual switch:

- Setting properties for an IPnet
- Displaying properties associated with an IPnet
- Removing an IPnet configured for an elastic virtual switch
- Displaying information about IPnets

Setting Properties for an IPnet

You use the `evsadm-setipnetprop` command to set properties for an IPnet. The command syntax is:

```
$ evsadm set-ipnetprop [-T tenant-name] -p
 prop=[value[,...]]\
EVS-switch-name/IPnet-name
```

The property associated with an IPnet is reset to the default value, if you do not specify any value for the property. For more information about the properties that can be set for an IPnet, see “How to Configure an Elastic Virtual Switch” on page 147.

EXAMPLE 61 Setting Properties for an IPnet

This example shows how to set the `pool` property for the IPnet `ora_ipnet`.

```
$ evsadm set-ipnetprop -T ABC -p pool=192.0.2.10-192.0.2.15 ORA/ora_ipnet
```

In this example, `ABC` is the name of the tenant and `ORA` is the name of the EVS.

Displaying Properties of an IPnet

You use the `evsadm-showipnetprop` command to display the properties associated with an IPnet. The command syntax is:

```
$ evsadm show-ipnetprop [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
[-p prop[,...]] [IPnet-name]
```

`-f` {*fname*=*value*[,...]} [,...]

A comma-separated name-value pair used to filter the output (row selection). If multiple filters are specified, then the displayed output is a result of an AND operation among the filters. If the filter value is multivalued, then the displayed output is a result of an OR operation among the filter values. The supported filters are:

- `tenant` – Filter the IPnet list by the tenant name
- `evs` – Filter the IPnet list by the elastic virtual switch name
- `ipnet` – Filter the IPnet list by the IPnet name
- `host` – Filter the IPnet list by the host name

Note - You can filter IPnets by using their property values. See Example 67, “Displaying VPort Properties,” on page 164.

`-p` *prop*

Specifies the properties that are associated with an IPnet. For information, see “How to Configure an Elastic Virtual Switch” on page 147.

`-o` *field*[,...]

Specifies a case-insensitive, comma-separated list of output fields to display. You can specify the following fields, which appear as columns in the output:

`all`	Displays all the output fields.
`NAME`	Name of the IPnet with the name of the elastic virtual switch with which it is associated in the format *EVS-switch-name*/*IPnet-name*.

`IPnet`	Name of the IPnet.
`EVS`	Name of the elastic virtual switch.
`TENANT`	Name of the tenant that owns the elastic virtual switch.
`PROPERTY`	Name of the IPnet property.
`PERM`	Permission of the property, which is either `rw` or `r-`.
`VALUE`	The current value of the property. If you have not set a value, it is shown as `--`. If the value is unknown, it is shown as `?`.
`DEFAULT`	The default value of the property. If the property does not a have default value, it is shown as `--`.
`POSSIBLE`	A comma-separated list of possible values for the property. If the possible values are unknown or unbounded, `--` is shown.

EXAMPLE 62 Displaying Properties of an IPnet

This example shows how to display properties for the IPnet `ora_ipnet`.

```
$ evsadm show-ipnetprop ora_ipnet
NAME              TENANT   PROPERTY  PERM VALUE                   DEFAULT   POSSIBLE
ORA/ora_ipnet     ABC      evs       r-   ORA                     --        --
ORA/ora_ipnet     ABC      subnet    r-   192.0.2.0/27            --        --
ORA/ora_ipnet     ABC      defrouter r-   192.0.2.1               --        --
ORA/ora_ipnet     ABC      pool      rw   192.0.2.10-192.0.2.15,  --        --
                                          192.0.2.20-192.0.2.25
ORA/ora_ipnet     ABC      tenant    r-   ABC                     --        --
```

EXAMPLE 63 Displaying the UUID of an IPnet

This example shows how to display the UUID for the IPnet `evs1/ipnet1`.

```
$ evsadm show-ipnetprop -p uuid -o name,tenant,property,perm,value evs1/ipnet1
NAME         TENANT      PROPERTY  PERM VALUE
evs1/ipnet1  sys-global  uuid      r-   d2698f0c-96ba-11e4-ab94-171c32874415
```

Removing an IPnet

You use the `evsadm remove-ipnet` command to remove an IPnet configured for the elastic virtual switch. The command syntax is:

```
$ evsadm remove-ipnet [-T tenant-name] EVS-switch-name/IPnet-name
```

This command removes the specified IPnet from the specified elastic virtual switch. You cannot remove an IPnet if any one of the VPorts is in use. A VPort is in use if it has a VNIC connected to it.

EXAMPLE 64 Removing an IPnet Configured for an Elastic Virtual Switch

This example shows how to remove the IPnet `ora_ipnet` from the elastic virtual switch `ORA`.

```
$ evsadm remove-ipnet ORA/ora_ipnet
```

Displaying IPnets

You use the `evsadm show-ipnet` command to display information about IPnets managed by the EVS controller or for the specified IPnet. The command syntax is:

```
$ evsadm show-ipnet [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] [IPnet-name]
```

`-f` {*fname*=*value*[,...]} [,...]
: A comma-separated name-value pair used to filter the output (row selection). If multiple filters are specified, then the displayed output is a result of an AND operation among the filters. If the filter value is multivalued, then the displayed output is a result of an OR operation among the filter values. The supported filters are `tenant`, `evs`, `ipnet`, and `host`.

`-o` *field*[,...]
: Specifies a case-insensitive, comma-separated list of output fields to display. You can specify the following fields, which appear as columns in the output:

  `all`
  : Displays all the output fields.

  `NAME`
  : Name of the IPnet along with the name of the elastic virtual switch with which it is associated.

  `IPNET`
  : Name of the IPnet.

EVS	Name of the elastic virtual switch.
TENANT	The name of the tenant that owns the elastic virtual switch.
SUBNET	Represents the subnet (either IPv4 or IPv6) for this IPnet.
START	Start address of the IP address range.
END	End address of the IP address range.
DEFROUTER	The IP address of the default router for the given IPnet.
AVAILRANGE	A comma-separated list of available IP addresses that can be assigned to VPort.

EXAMPLE 65 Displaying IPnet for an Elastic Virtual Switch

This example displays the IPnet configured for the elastic virtual switch ORA.

```
$ evsadm show-ipnet
NAME          TENANT     SUBNET          DEFROUTER    AVAILRANGE
ORA/ora_ipnet sys-global 192.0.2.0/27    192.0.2.1    192.0.2.3-192.0.2.30
```

Administering VPort Configuration

This section describes how to perform the following tasks for a VPort:

- Setting properties for a VPort
- Displaying properties associated with a VPort
- Displaying information about VPorts
- Resetting a VPort
- Removing a VPort

Setting Properties for a VPort

You use the `evsadm set-vportprop` command to set properties for a VPort. The command syntax is:

```
$ evsadm set-vportprop [-T tenant-name] -p prop=value[,...] EVS-switch-name/VPort-name
```

`-T` *tenant-name* — Specifies the name of the tenant.

`-p` *prop=value*[...,] — Specifies the values of a property for the specified VPort. If the VPort has a VNIC connected to it, then setting the property on that VPort results in change of VNIC's property. For information about VPort properties, see Table 4, “Virtual Port Properties,” on page 129.

Note - You cannot change the property of the system VPort. For more information about the system VPort, see “How to Configure an Elastic Virtual Switch” on page 147.

EVS-switch-name/*VPort-name* — Specifies the name of the elastic virtual switch or the VPort for which the properties are set.

Note - You cannot modify the `ipaddr`, `macaddr`, `evs`, and `tenant` properties after you have created the VPort.

EXAMPLE 66 Setting a Property for a VPort

This example shows how to set the maximum bandwidth property to `1G` for `HR/vport0`.

```
$ evsadm set-vportprop -p maxbw=1G HR/vport0
```

Displaying Properties of a VPort

You use the `evsadm show-vportprop` command to display properties of a VPort. The command syntax is:

```
$ evsadm show-vportprop [-f {fname=value[,...]}[,...] [[-c] -o field[,...]] \
[-p prop[,...]] [[EVS-switch-name]/[VPort-name]]
```

This command shows the current values of one or more properties for either all VPorts or the specified VPort. If VPort properties are not specified, then all available VPort properties are

displayed. For information about the VPort properties, see Table 4, “Virtual Port Properties,” on page 129.

[-f {*fname*=*value*[,...]} [,...]

A comma-separated name-value pair used to filter the output (row selection). If multiple filters are specified, then the displayed output is a result of an AND operation among the filters. If the filter value is multivalued, then the displayed output is a result of an OR operation among the filter values. The supported filters are:

- `tenant` – Filter the VPort properties by the tenant name
- `EVS` – Filter the VPort properties by the elastic virtual switch name
- `vport` – Filter the VPort properties by the VPort name
- `host` – Filter the VPort properties by the host name

Note - You can filter VPorts by using their property values. See Example 67, “Displaying VPort Properties,” on page 164.

-o *field*[,...]

Specifies a case-insensitive, comma-separated list of output fields to display. You can specify the following fields, which appear as columns in the output:

`all`	Displays all the output fields.
`NAME`	Name of the VPort with the name of the elastic virtual switch with which the VPort is associated in the format *EVS-switch-name*/*VPort-name*.
`TENANT`	Name of the tenant that owns the elastic virtual switch.
`PROPERTY`	Name of the VPort property.
`PERM`	The read or write permissions of the property. The value shown is either `r-` or `rw`.
`VALUE`	The current property value. If the value is not set, it is shown as `--`. If it is unknown, the value is shown as `?`.
`DEFAULT`	The default value of the property. If the property has no default value, `--` is shown.

POSSIBLE	A comma-separated list of possible values for the property. If the values span a numeric range, min - max might be shown as shorthand. If the possible values are unknown or unbounded, -- is shown.

EXAMPLE 67 Displaying VPort Properties

The following example displays the VPort properties for the VPort `vport0`.

```
$ evsadm show-vportprop ORA/vport0
NAME          TENANT      PROPERTY  PERM VALUE          DEFAULT    POSSIBLE
ORA/vport0    sys-global  cos       rw   --              0         0-7
ORA/vport0    sys-global  maxbw     rw   --              --        --
ORA/vport0    sys-global  priority  rw   --              medium    low,medium,high
ORA/vport0    sys-global  ipaddr    r-   192.0.2.2/24    --        --
ORA/vport0    sys-global  macaddr   r-   2:8:20:b0:6e:63 --        --
ORA/vport0    sys-global  evs       r-   ORA             --        --
ORA/vport0    sys-global  tenant    r-   sys-global      --        --
```

The following example shows how to filter the virtual ports by using the values of their property.

```
$ evsadm show-vportprop -p priority
NAME              TENANT      PROPERTY  PERM VALUE         EFFECTIVE POSSIBLE
evs1/vport0       sys-global  priority  rw   --            medium    low,
                                                                     medium,
                                                                     high
evs1/vport1       sys-global  priority  rw   high          high      low,
                                                                     medium,
                                                                     high
evs1/vport2       sys-global  priority  rw   --            medium    low,
                                                                     medium,
                                                                     high
$ evsadm show-vport -f priority=high
NAME               TENANT        STATUS VNIC         HOST
evs1/vport1        sys-global    free   --           --
```

The output shows only the VPort `evs1/vport1` whose `priority` property is set to `high`.

EXAMPLE 68 Displaying the UUID for a VPort

This example shows how to display the UUID for the VPort `evs1/vport1`.

```
$ evsadm show-vportprop -p uuid -o name,tenant,property,perm,value evs1/vport1
NAME          TENANT      PROPERTY  PERM VALUE
```

```
evs1/vport1   sys-global  uuid      r-   7d4c90e0-96bb-11e4-ab96-171c32874415
```

Displaying VPorts

You use the `evsadm show-vport` command to display VPorts. The command syntax is:

```
$ evsadm show-vport [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
[[EVS-switch-name/][VPort-name]]
```

`-f {`*fname*`=`*value*`[,...]}[,...]`

A comma-separated name-value pair used to filter the output (row selection). If multiple filters are specified, then the displayed output is a result of an AND operation among the filters. If the filter value is multivalued, then the displayed output is a result of an OR operation among the filter values. The supported filters are:

- `tenant` – Filter the VPort list by the tenant name
- `EVS` – Filter the VPort list by the elastic virtual switch name
- `vport` – Filter the VPort list by the VPort name
- `host` – Filter the VPort list by the host name

`-o` *field*`[,...]`

Specifies a case-insensitive, comma-separated list of output fields to display. You can specify the following fields, which appear as columns in the output:

`all`	Displays all the output fields.
`NAME`	Name of the VPort with the name of the elastic virtual switch with which it is associated in the format *EVS-switch-name*/*VPort-name*.
`TENANT`	Name of the tenant that owns the elastic virtual switch.
`STATUS`	Displays whether the VPort is in use or free. A VPort is in use if the VPort is associated with a VNIC. Otherwise, the VPort is free.
`VNIC`	Name of the VNIC associated with the VPort.
`HOST`	Name of the host that has the VNIC associated with the VPort.

EXAMPLE 69 Displaying VPort Information

This example displays information about the VPort `vport0`.

```
$ evsadm show-vport
NAME            TENANT        STATUS VNIC      HOST
ORA/vport0      sys-global    used   vnic1     s11-client
```

Resetting a VPort

When you delete a VNIC associated with a VPort, the state of the VPort is `free`. The VPort can be in the `used` state even if you delete the VNIC that is associated with the VPort in the following situations:

- The EVS node is unable to reach the EVS controller when you delete the VNIC in the EVS node.
- The VNIC associated with the VPort is not deleted before you reboot the EVS node.

To reset the state of a VPort to `free`, use the `evsadm reset-vport` command. The command syntax is:

```
$ evsadm reset-vport [-T tenant-name] EVS-switch-name/VPort-name
```

Removing a VPort

If a VNIC is associated with the VPort, then the removal of the VPort fails. Therefore, you must first check whether a VNIC is associated with the VPort that you want to remove by using the `evsadm show-vport` command. You use the `evsadm remove-vport` command to remove a VPort from an elastic virtual switch. The command syntax is:

```
$ evsadm remove-vport [-T tenant-name] EVS-switch-name/VPort-name
```

This command removes the specified VPort. When a VPort is removed, the IP address and the MAC address associated with the VPort are released.

EXAMPLE 70 Removing a VPort

This example shows how to remove the VPort `vport0` configured for the elastic virtual switch `ORA`.

```
$ evsadm remove-vport -T tenantA ORA/vport0
```

Deleting an Elastic Virtual Switch

This section describes how to delete an elastic virtual switch. You can delete an elastic virtual switch only when all the VPorts of an elastic virtual switch are free. Therefore, VPorts must not be associated with VNICs.

▼ How to Delete an Elastic Virtual Switch

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. Check whether VPorts are used by the elastic virtual switch.

```
$ evsadm show-evs
```

You cannot delete an elastic virtual switch if a VPort is in use. A VPort is in use if a VNIC is connected to the VPort. The `STATUS` field in the `evsadm show-evs` command output displays whether an elastic virtual switch is busy or idle.

If a VPort is in use, you need to delete the VNIC associated with the VPort as follows:

```
$ dladm delete-vnic VNIC
```

2. Delete the elastic virtual switch.

```
$ evsadm delete-evs [-T tenant-name] EVS-switch-name
```

This command deletes the specified elastic virtual switch and all the VPorts and the IPnet associated with the elastic virtual switch.

Example 71 Deleting an Elastic Virtual Switch

The following example shows how to delete the elastic virtual switch `ORA`.

```
$ evsadm show-evs
EVS           TENANT        STATUS NVPORTS IPNETS      HOST
ORA           sys-global    idle   0       ora_ipnet   --
$ evsadm delete-evs ORA
$ evsadm show-evs ORA
evsadm: failed to show EVS(s): evs not found
```

The following example shows how to delete the elastic virtual switch `EVS1`, which is busy.

```
$ evsadm show-evs EVS1
EVS           TENANT        STATUS NVPORTS IPNETS      HOST
```

```
EVS1          sys-global    busy   1       evs1_ipnet  s11-server
$ evsadm show-vport EVS1/vport1
NAME                TENANT        STATUS VNIC         HOST
EVS1/vport1         sys-global    used   vnic1        s11-server
$ dladm delete-vnic vnic1
$ evsadm show-evs EVS1
EVS           TENANT        STATUS NVPORTS IPNETS      HOST
EVS1          sys-global    idle   1       evs1_ipnet  --
$ evsadm delete-evs EVS1
$ evsadm show-evs EVS1
evsadm: failed to show EVS(s): evs not found
```

Monitoring Elastic Virtual Switches

You can monitor network traffic statistics for the virtual ports of an elastic virtual switch to obtain the following information:

- The amount of network traffic that is sent and received by a VM, which provides information about the workload on the VM.
- The number of packets that are dropped inbound (`idrops`) and outbound (`odrops`). These values provide information about faulty networks.
- The amount of network traffic that is sent and received by all the VMs on a compute node, which helps you to perform capacity planning.

You use the `evsstat` command to monitor elastic virtual switches. The `evsstat` command reports runtime statistics for each VPort of the elastic virtual switch. It also reports the statistics of VNICs associated with the VPorts. For more information about EVS and virtual ports, see the `evsadm`(8) man page.

The `evsstat` command is a Remote Administration Daemon (RAD) client, and it communicates with a remote EVS controller to run all the `evsstat` subcommands. Before using the `evsstat` command, you must specify a resolvable hostname or the IP address of the EVS controller by using the `evsadm set-prop` command. The command syntax is:

```
$ evsadm set-prop -p controller=ssh://[username@]hostname-or-IP-address
```

In addition, you must set up SSH authentication by using the preshared public key between the host where you run the `evsstat` command and the EVS controller. You need SSH authentication with the preshared public key for the `evsstat` command to communicate with the EVS controller non-interactively and securely. For more information, see “About SSH Authentication and the `evsuser`” on page 138.

The command syntax for `evsstat` is:

```
$ evsstat [-f {fname=value[,...]}[,...] [[-c] -o field[,...]] [-u R|K|M|G|T|P] \
[EVS-switch-name[/VPort-name]] [interval] [count]
```

EVS-switch-name
: Specifies the name of the elastic virtual switch whose statistics you want to monitor. If the name of the elastic virtual switch is not specified, statistics for all elastic virtual switches are displayed.

VPort-name
: Specifies the name of the VPort whose statistics you want to monitor. The statistics are displayed only for the VNIC connected to the specified VPort. You must specify the name of the elastic virtual switch and then specify the name of the VPort.

`-f` {*fname=val*[,...]} [,...]
: A comma-separated name-value pair used to filter the output (row selection). If multiple filters are specified, then the displayed output is a result of an AND operation among the filters. If the filter value is multivalued, then the displayed output is a result of an OR operation among the filter values. The supported filters are `tenant`, `evs`, and `host`.

`-o` *field*[,...]]
: Specifies a case-insensitive, comma-separated list of output fields to display. You can specify the following fields, which appear as columns in the output:

- `vport`
- `evs`
- `tenant`
- `vnic`
- `host`
- `ipkts`
- `rbytes`
- `opkts`
- `idrops`
- `odrops`

`-u R|K|M|G|T|P`
: Specifies the unit in which the statistics are displayed. If not specified, then different units, as appropriate, are used to display the statistics, using the format `xy.zU`, where `x`, `y`, and `z` are numbers and `U` is the appropriate unit. The supported units are:

- `R` – Raw count
- `K` – Kilobits
- `M` – Megabits
- `G` – Gigabits

- `T` – Terabits
- `P` – Petabits

interval Specifies the time in seconds at which you want to refresh the network statistics.

count Specifies the number of times to refresh the statistics. You must specify the interval and then specify the count.

EXAMPLE 72 Monitoring Elastic Virtual Switches

The following example displays statistics for all elastic virtual switches.

```
$ evsstat
VPORT        EVS      TENANT       IPKTS     RBYTES      OPKTS   OBYTES
sys-vport0   ORA      sys-global   101.88K   32.86M      40.16K  4.37M
sys-vport2   ORA      sys-global   4.50M     6.78G       1.38M   90.90M
sys-vport0   HR       sys-global   132.89K   12.25M      236     15.82K
sys-vport1   HR       sys-global   144.47K   13.32M      247     16.29K
```

The following example displays statistics for the specified elastic virtual switch, `evs0`.

```
$ evsstat ORA
VPORT         EVS      TENANT       IPKTS     RBYTES    OPKTS    OBYTES
sys-vport0    ORA      sys-global   101.88K   32.86M    40.16K   4.37M
sys-vport2    ORA      sys-global   4.50M     6.78G     1.38M    90.90M
```

The following example displays statistics for the specified VPort, `evs0/sys-vport2`.

```
$ evsstat ORA/sys-vport2
VPORT        EVS     TENANT       IPKTS   RBYTES   OPKTS  OBYTES
sys-vport2   ORA     sys-global   4.50M   6.78G    1.38M  90.90M
```

The following example shows the statistics of a VPort with an interval value of `1` second and count value of `3`. The statistics are refreshed three times with an interval of one second.

```
$ evsstat ORA/sys-vport2 1 3
VPORT         EVS      TENANT       IPKTS   RBYTES   OPKTS  OBYTES
sys-vport2    ORA      sys-global   4.50M   6.78G    1.38M  90.90M
sys-vport2    ORA      sys-global   4.50M   6.78G    1.38M  90.90M
sys-vport2    ORA      sys-global   4.50M   6.78G    1.38M  90.90M
```

The following example shows the statistics for the specified output fields.

```
$ evsstat -o vport,evs,vnic,host,ipkts,opkts
VPORT          EVS      VNIC      HOST    IPKTS     OPKTS
sys-vport0     ORA      vnic0     host1   101.88K   40.16K
```

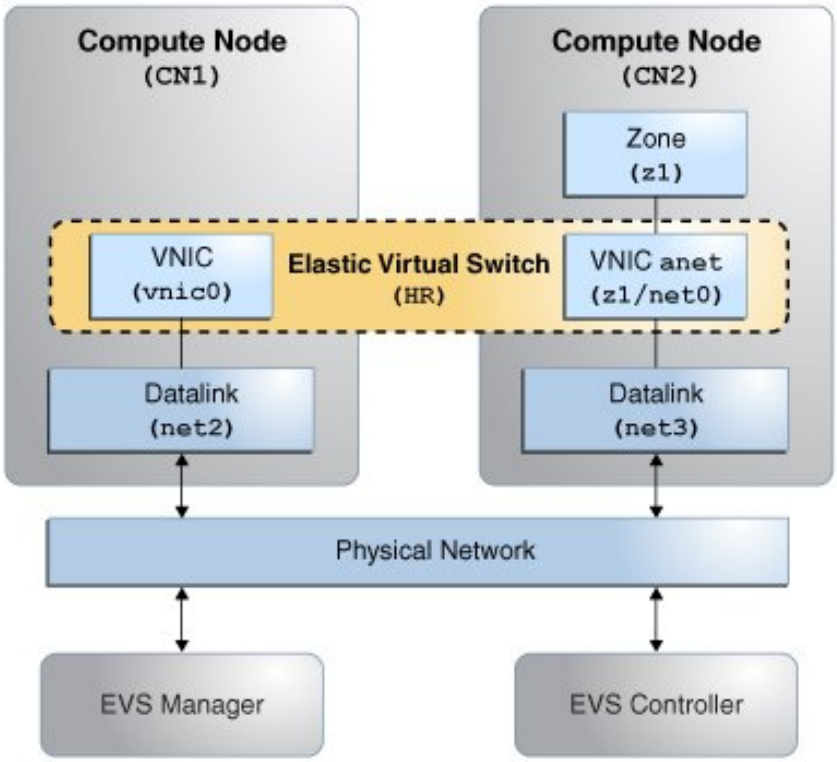
```
sys-vport2     ORA      vnic0  host2   4.50M     1.38M
sys-vport0     HR       vnic1     host1   132.89K   236
sys-vport1     HR       vnic1     host2   144.47K   247
```

Use Case: Configuring an EVS Network Topology

Objective – This use case shows how to set up a network topology consisting of EVS components for a specific tenant.

The network is shown in the following figure:

FIGURE 21 Elastic Virtual Switch Configuration for a Tenant



The network consists of the following components:

- One EVS controller node
- One EVS Manager node
- Two compute nodes `CN1` and `CN2`, each with its own datalink
- A virtual machine `z1` to be hosted by `CN2`

The following configurations, not shown in the figure, will be implemented:

- The controller's hostname is `evs-controller.example.com`.
- The controller is configured as a VLAN.
- The elastic virtual switch (`HR`) is created for a tenant, `tenantA`.
- The subnet for `tenantA` is called `HR/hr_ipnet`.

In this sample case, the necessary EVS packages are already properly installed on all nodes. Further, SSH authentication for `evsuser` has been completed on all nodes. The configuration is performed as `evsuser` on the EVS Manager node.

Configure the EVS Manager node to connect with the controller.

```
manager$ su - evsuser
evsuser@manager$ evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

Connect to the controller. The configuration consists of setting up a VLAN with a range of VLAN IDs and creating uplink ports for the compute nodes. The resulting configuration is displayed.

```
evsuser@controller$ evsadm set-controlprop -p l2-type=vlan
evsuser@controller$ evsadm set-controlprop -p vlan-range=200-300
evsuser@controller$ evsadm set-controlprop -h CN1 -p uplink-port=net2
evsuser@controller$ evsadm set-controlprop -h CN2 -p uplink-port=net3

evsuser@controller$ evsadm show-controlprop -p l2-type,vlan-range,uplink-port
NAME             VALUE         DEFAULT        HOST
l2-type          vlan          vlan           --
vlan-range       200-300       --             --
uplink-port      net2          --             CN1
uplink-port      net3          --             CN2
```

Create the switch for the tenant and specify the tenant's subnet. The resulting configuration is displayed.

```
evsuser@controller$ evsadm create-eps -T tenantA HR
evsuser@controller$ evsadm add-ipnet -T tenantA -p subnet=192.0.2.0/27 HR/hr_ipnet

evsuser@controller$ evsadm
```

```
NAME         TENANT     STATUS   VNIC    IP             HOST
HR           tenantA    --       --      hr_ipnet       --
   vport0    --         free     --      192.0.2.2/27   --

evsuser@controller$ evsadm show-vportprop -p macaddr,ipaddr HR/vport0
NAME         TENANT     PROPERTY  PERM  VALUE            DEFAULT POSSIBLE
HR/vport0    tenantA    ipaddr    r-    192.0.2.2/27     --      --
HR/vport0    tenantA    macaddr   r-    2:8:20:d8:da:10  --      --

evsuser@controller$ evsadm show-evs -L
EVS   TENANT     VID   VNI
HR    tenantA    200   --
```

Connect to `CN1` to configure it. The configuration consists of a VNIC to enable CN1 to connect to the switch. The IP address to assign is obtained by checking the allowed IP address for the VNIC.

```
evsuser@CN1$ evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com

evsuser@CN1$ dladm create-vnic -t -T tenantA -c HR vnic0
evsuser@CN1$ dladm show-linkprop -p allowed-ip vnic0
LINK     PROPERTY     VALUE      EFFECTIVE  DEFAULT   POSSIBLE
vnic0    allowed-ips  192.0.2.2  192.0.2.2  --        --

evsuser@CN1$ ipadm create-ip -t vnic0
evsuser@CN1$ ipadm create-addr -t -a 192.0.2.2 vnic0
```

Connect to `CN2` to configure it. The configuration consists of creating the zone for the tenant. The zone is configured with a VNIC's `anet` resource to provide connectivity to the switch. Note that you still need to complete zone configuration by logging in after the zone has booted. The remaining configuration is not related to EVS and is not covered in this example.

```
evsuser@CN2$ evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com

evsuser@CN2$ zonecfg -z z1
zonecfg:z1> create
create: Using system default template 'SYSdefault'
zonecfg:z1> set zonepath=/export/zones/z1
zonecfg:z1> set tenant=tenantA
zonecfg:z1> select anet linkname=net0
zonecfg:z1:anet> set evs=HR
zonecfg:z1:anet> end
zonecfg:z1> commit
zonecfg:z1> exit
evsuser@CN2$

evsuser@CN2$ zoneadm -z z1 install
...
```

```
evsuser@CN2$ zoneadm -z z1 boot
...
evsuser@CN2$ zlogin -C z1
...

evsuser@CN2.z1$ dladm show-vnic -c
LINK          TENANT   EVS  VPORT   OVER  MACADDRESS       IDS
z1/net0       tenantA  HR   vport0  net3  2:8:20:d8:da:10  VID:200

evsuser@CN2.z1$ ipadm
NAME        CLASS/TYPE   STATE   UNDER   ADDR
lo0         loopback     ok      --      --
   lo0/v4   static       ok      --      127.0.0.1/8
   lo0/v6   static       ok      --      ::1/128
net0        ip           ok      --      --
   net0/v4  inherited    ok      --      192.0.2.3/27
```

♦ ♦ ♦ CHAPTER 7

Managing Network Resources

This chapter explains how to implement IP quality of service (QoS) by allocating network resources through datalink properties and flows. It contains the following topics:

- "Managing Network Resources by Using Datalink Properties"
- "Managing NIC Rings"
- "Managing Pools and CPUs"
- "Using the Large Receive Offload Feature in Oracle Solaris"
- "Managing Network Resources by Using Flows"
- "Use Case: Managing Network Resources by Setting Datalink and Flow Properties"

For an introduction to network resource management, see "Overview of Network Resource Management" on page 30. For a demonstration, see Managing Network Resources Using Oracle Solaris.

To manage network resources and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 34.

Note - In Oracle Solaris 11.4, the Service Management Facility (SMF) manages persistent network configuration. Consequently, some datalink property names have changed. See Appendix A, "Datalink Properties," in *Configuring and Managing Network Components in Oracle Solaris 11.4*.

Managing Network Resources by Using Datalink Properties

Network resources are regulated in datalinks through the following properties:

- `max-bw` – Maximum amount of bandwidth for a datalink.
- `rx-rings` and `tx-rings` – Number of receive rings (Rx rings) and transmit rings (Tx rings) of a NIC assigned to a datalink.

- `pool` – Name of the CPU pool containing sets of CPUs for a datalink.
- `cpus` – Name of the CPUs you assign to a datalink.
- `lro` – Status of the large receive offload (LRO) feature for a datalink.

To configure network resource properties of datalinks, use one of the following commands:

- `$ dladm create-vnic -l` *link* `-p` *prop=value*`[,...]` *VNIC*

  link Name of the link which can be either a physical link or a virtual link.

  prop Datalink property.
- `$ dladm set-linkprop -p` *prop=value*`[,...]` *link*

For reference, see the `dladm`(8) man page.

Managing NIC Rings

Receive (Rx) rings and transmit (Tx) rings are NIC hardware resources through which the system receives and sends network packets, respectively. These rings enable you to tune the system's packet processing to increase efficiency, for example, by increasing the number of receive rings of a link that is receiving a large volume of packets.

Note - The current Oracle Solaris release supports kernel network data path bypass for the User Datagram Protocol (UDP) but only on physical, not virtual, interfaces. Also, a bypass socket can only be associated with one kernel data path bypass-capable network interface for both send and receive operations. See "Using the Network Data Path Bypass Capability for UDP" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Ring-Related Properties

The following properties are related to rings.

- `rx-rings-available` – Number of Rx rings you can allocate to hardware-based clients on the physical datalink.
- `rx-rings` – Number of Rx rings exclusively used by the datalink.
- `tx-rings-available` – Number of Tx rings you can allocate to hardware-based clients on the physical datalink.

- `tx-rings` – Number of Tx rings exclusively used by the datalink.
- `ring-group` – indicates whether the driver supports assigning of dedicated ring groups for the NICs.

Configuring Clients and Allocating Rings

An entity configured over a NIC is called a client. Clients enable communication between a system and other network nodes. From the perspective of ring allocation, a client can be one of two types:

- Hardware-based clients have exclusive use of one or more NIC rings. Assigning rings for exclusive use depends on the ring allocation supported by the NICs.
- Software-based clients do not have exclusive use of NIC rings. Rings are shared with other existing software-based clients or with the primary client. The rings that these clients use depend on the number of hardware-based clients that have priority in ring allocation.

Ring allocation to VLAN clients differs based on the command you use to create the VLAN.

- `$ dladm create-vlan -l` *link* `-v` *vid VLAN*

  This command creates a VLAN that shares the same MAC address and consequently, Rx and Tx rings, as the underlying datalink.
- `$ dladm create-vnic -l` *link* `-v` *vid VNIC*

  This command creates a *VLAN-VNIC* with a unique MAC address. Therefore, assuming that the NIC supports hardware-based clients, the VNIC can be assigned its own dedicated rings.

▼ How to Configure Clients and Allocate Rings

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. Display the properties of the underlying physical datalink.

```
$ dladm show-linkprop -p rx-rings-available,tx-rings-available link
```

Determine the following information from the output of the command:

- Whether the NIC supports hardware-based clients
- The availability of rings to allocate to hardware-based clients
- The availability of hardware-based clients that you can configure on the link

2. **Depending on the information from the previous step, perform one of the following:**

   - **Create the hardware-based client with the following syntax:**

     ```
     $ dladm create-vnic -p rx-rings=value[,tx-rings=value] -l link VNIC
     ```

     where *value* can be one of the following:

     - `hw` - Indicates that you are configuring a hardware-based client.
     - *number* - Indicates that you are configuring a hardware-based client only. The number refers to the quantity of rings that you can allocate to the client for its exclusive use.

   - **Create the software-based client with the following syntax:**

     ```
     $ dladm create-vnic -p rx-rings=sw[,tx-rings=sw] -l link VNIC
     ```

   Alternatively, if the client was previously created, you can use the `dladm set-linkprop` command to set the ring properties.

3. **(Optional) Verify the ring information of the client that you created.**

   ```
   $ dladm show-linkprop -p rx-rings,tx-rings VNIC
   ```

4. **(Optional) Verify the link's rings that are distributed among different clients.**

   ```
   $ dladm show-phys -H link
   ```

Creating VNICs With Dedicated Ring Groups

Assigning dedicated ring groups to VNICs helps in better resource isolation. Enforcing hardware SLAs on the VNICs enables hardware anti-spoofing for the VNICs.

For physical links, `ring-group`, which is a read-only property, indicates whether the driver supports the dedicated ring group feature. You cannot enable or disable this feature for physical datalinks.

For VNICs, the `ring-group` property can be specified only during the VNIC's creation, as described in the following procedure. You cannot set the property on already existing VNICs.

▼ How to Create VNICs to Have Their Own Ring Groups

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See “Using Rights Profiles to Perform Network Configuration” on page 34.

1. **Check if the physical link supports exclusive ring groups.**

   ```
   $ dladm show-linkprop -p ring-group linkname
   ```

2. **Create a VNIC by specifying the `ring-group` property.**

   ```
   $ dladm create-vnic -l linkname -p ring-group=value VNIC
   ```

   `ring-group` — Specifies whether the VNIC can be assigned their own ring group. The valid values are:

   `auto` – Specifies whether `exclusive` or `shared` is used on a particular physical link is decided by the system. The default value is `auto`.

   `exclusive` – Specifies that the VNIC creation must fail if `exclusive` ring group is not available.

   `shared` – Specifies that the dedicated resources are not allocated.

3. **Check if the VNIC has exclusive ring groups.**

   ```
   $ dladm show-linkprop -p ring-group VNIC
   ```

4. **Check the exclusive ring groups of the VNIC.**

   ```
   $ dladm show-phys -H linkname
   ```

Example 73 Creating a VNIC With Ring Groups

This example shows how to create a VNIC that has exclusive ring group.

```
$ dladm show-linkprop -p ring-group net1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net1     ring-group      r-   exclusive    exclusive    --        shared,
                                                                  exclusive

$ dladm create-vnic -l net1 -p ring-group=exclusive vnic5

$ dladm show-linkprop -p ring-group vnic5
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
vnic5    ring-group      r-   exclusive    exclusive    auto      auto,shared,
                                                                  exclusive

$ dladm show-phys -H net3
LINK               RINGTYPE RINGS              CLIENTS
net1               RX       0-31               <default,mcast>
net1               RX       33-36              vnic5
net1               TX       0-31               <default>
```

```
net1                TX       33-36               vnic5
```

Example 74 Displaying the Rx Rings and Tx Rings for a VNIC With Exclusive Ring Group

This example displays the Rx and Tx rings in the `EFFECTIVE` field for a VNIC that has exclusive ring group.

```
$ dladm show-linkprop -p rx-rings,tx-rings vnic1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
vnic1    rx-rings        r-   --           4            --        --
vnic1    tx-rings        r-   --           4            --        --
```

Example 75 Creating a VNIC With the `ring-group` and `bw-share` Properties

This example shows the creation of a VNIC by specifying the ring group along with the share of bandwidth.

```
$ dladm show-linkprop -p ring-group net1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net1     ring-group      r-   exclusive    exclusive    --        shared,
                                                                  exclusive
$ dladm create-vnic -l net2 -p ring-group=exclusive,bw-share=80 vnic1
$ dladm create-vnic -l net2 -p ring-group=exclusive,bw-share=20 vnic2
$ dladm show-linkprop -p ring-group,bw-share vnic1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
vnic1    ring-group      r-   exclusive    exclusive    auto      auto,shared,
                                                                  exclusive
vnic1    bw-share        rw    80           80%          --        1-100
```

VNICs With Ring Group in SR-IOV Mode

SR-IOV mode supports VNICs with the `ring-group` property set to `exclusive`. You can create a VNIC in the SR-IOV mode only if the `iov` property of the VNIC is either `auto` or `off`. If the `iov` property is set to `on`, then the VNIC creation fails. For more information about SR-IOV VNICs, see "Using Single Root I/O Virtualization With VNICs" on page 69.

Zone's `anet` Resource With Ring Group

The `anet` resource of Oracle Solaris zones supports the `ring-group` property. The value that can be set for this property is the same as that of a VNIC. Similar to VNIC creation, you cannot create an `anet` resource if the `iov` property is set to `on` and `ring-group` is set to `exclusive`. For more information, see *Oracle Solaris Zones Configuration Resources*.

Note - Any value other than `auto` takes precedence when the value of `iov` or `ring-group` is set a value other than `auto`.

Hardware Offload Support for the `ring-group` Property

When the `ring-group` property of a VNIC has the value `exclusive`, you can offload the SLA implementation in MAC to the NIC only if the NIC supports hardware SLAs. The offloading helps to save the CPU cycles. For more information, see "Setting Hardware SLA Properties for VF VNICs" on page 76.

Support of Hardware Flows

During VNIC creation, if the `ring-group` property is set to `auto` or `exclusive`, the VNIC tries to get a hardware flow from the underlying link. If the flow allocation is successful, the VNIC uses the existing ring group interface to initialize the Tx or Rx ring groups for this flow. Then, the ring groups are usable and exclusively owned by this VNIC.

Displaying Ring Use and Ring Assignments

Use the following commands to display information about rings, depending on the specific information:

- Display all possible values, configured values, and effective values of Rx rings and Tx rings.

  ```
  $ dladm show-linkprop -p rx-rings,tx-rings link
  ```

- Display how the rings of a physical datalink are currently being used by clients.

  ```
  $ dladm show-phys -H link
  ```

- Display ring group resource information for the underlying physical device.

  ```
  $ dladm show-phys -G link
  ```

EXAMPLE 76 Ring Use and Ring Assignments on a Datalink

The following example shows the ring assignments on the datalink `net4`.

```
$ dladm show-linkprop net4
LINK     PROPERTY              PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
```

```
...
net4     rx-rings               rw   1           --          --        sw,hw,<1-7>
net4     tx-rings               rw   1           --          --        sw,hw,<1-11>
net4     tx-rings-available     r-   10          10          --        --
net4     rx-rings-available     r-   7           7           --        --
...
```

The output shows that `net4` has exclusive use of one Rx ring and one Tx ring. The datalink has seven Rx rings and ten Tx rings that are available for allocation to the clients. You can create three hardware-based Rx clients and three hardware-based Tx clients over the datalink `net4`.

The following example shows the ring use for the datalink `net0`.

```
$ dladm show-phys -H net0
LINK   RINGTYPE   RINGS   CLIENTS
net0   RX         0-1     <default,mcast>
net0   TX         0-7     <default>net0
net0   RX         2-3     net0
net0   RX         4-5     --
net0   RX         6-7     --
```

Based on the output, the two Rx rings allocated to `net0` are rings `2` and `3`. For Tx rings, `net0` uses rings `0` through `7`.

The following example shows the ring groups used by UMACs (Message Authentication Code by using universal hashing) and ring groups used by the VNICs.

```
$ dladm show-phys -G net2
LINK              RG-AVAIL  RG-INUSE-UMAC  RG-INUSE-VNIC   RG-INUSE-FLOW
net2              309       0              1                0
```

Managing Pools and CPUs

Oracle Solaris supports two properties, `pool` and `cpus`, to manage resources for tasks beyond network processes. Use the `cpus` property to assign specific CPUs to a datalink.

If you are managing resources while working with non-global zones, use the `pool` property instead. This property enables you to integrate network resource management with CPU allocation and zone administration.

For example, you use the zone commands `zonecfg` and `poolcfg` to configure a pool of resources for zones. To dedicate that same pool to also manage network processes, then you configure the datalink's `pool` property. When the datalink with the configured `pool` is assigned to a non-global zone's network interface, then the datalink is bound to the zone's pool. If you set

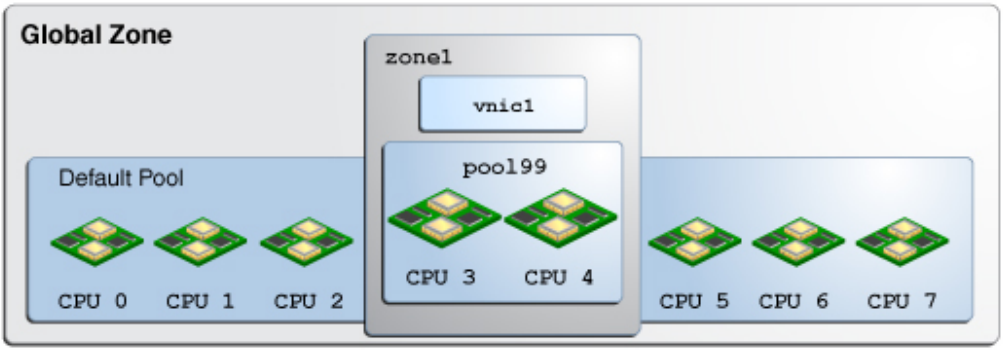
the zone to be exclusive, then CPU resources in the pool can no longer be used by other links that are not assigned to the zone.

See Chapter 13, “Creating and Administering Resource Pools Tasks” in *Administering Resource Management in Oracle Solaris 11.4* and the `poolcfg`(8) man page.

Note - The `cpus` and `pool` properties are mutually exclusive. You cannot set both properties for a given datalink.

The following figure shows how pools work when the `pool` property is assigned to a datalink.

FIGURE 22 `pool` Property of a VNIC Assigned to a Zone



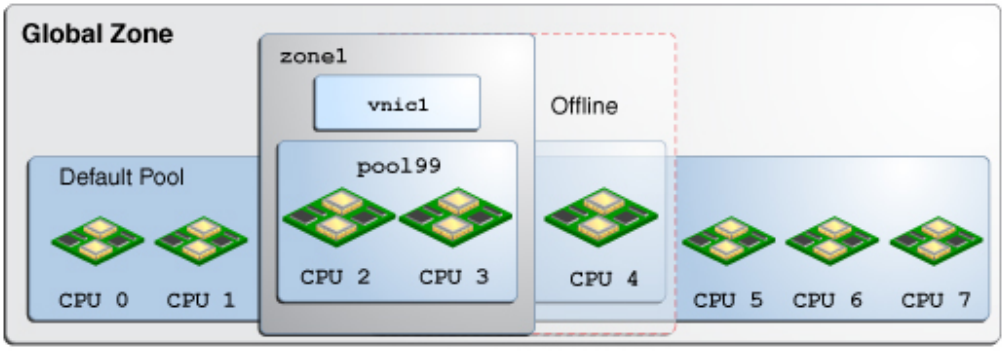
The figure shows a system with eight CPUs. When no pools are configured, all the CPUs belong to the *default pool* and are used by the global zone. In the figure, `pool99` is created and consists of `CPU 3` and `CPU 4`. The pool is associated with `zone1`, an exclusive zone. If `pool99` is set as a property of `vnic1`, then `pool99` becomes dedicated to also manage `vnic1`'s networking processes. After `vnic1` is assigned to be `zone1`'s network interface, the CPUs in `pool99` are reserved to manage both networking and non-networking processes of `zone1`.

The `pool` property is dynamic in nature. Zone pools can be configured with a range of CPUs, and the kernel determines which CPUs are assigned to the pool's CPU set. Changes to the pool are automatically implemented for the datalink. In contrast, assigning specific CPUs to the link by using the `cpu` property requires you to specify the CPU to be assigned. You have to set the `cpu` property every time you want to change the CPU components of the pool.

For example, suppose that the system `CPU 4` in Figure 22, “`pool` Property of a VNIC Assigned to a Zone,” on page 183 is taken offline. Because the `pool` property is dynamic, the

software automatically associates an additional CPU with the pool. Hence, the pool's original configuration of two CPUs is preserved. For `vnic1`, the change is transparent. The updated configuration is shown in the following figure.

FIGURE 23 Automatic Reconfiguration of the `pool` Property



When you display datalink property information for `pool` and `cpus`, the `EFFECTIVE` column in the output shows the corresponding pool and CPU resources that are used for network processes, as shown in this example:

```
$ dladm show-linkprop -p cpus
LINK  PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net0  cpus      rw    --     0-63       --       --

$ dladm show-linkprop -p pool
LINK   PROPERTY  PERM  VALUE  EFFECTIVE     DEFAULT  POSSIBLE
vnic1  pool      rw    --     pool99         --       --
```

If you are configuring a non-global zone's resources, use the appropriate `zonecfg` or `poolcfg` commands rather than directly configuring the datalink's `cpus` or `pool` properties. When you bind the datalink to the zone and then reboot the zone, these datalinks are automatically set according to the zone configurations. These settings are then reported under the `EFFECTIVE` column, even though their `VALUE` column shows empty fields.

▼ How to Configure a CPU Pool for a Datalink

Before You Begin You must have completed the following tasks:

- Created a processor set with its assigned number of CPUs
- Created a pool with which the processor set will be associated
- Associated the pool with the processor set

Note - For the instructions to complete these prerequisites, see "How to Modify a Pools Configuration" in *Administering Resource Management in Oracle Solaris 11.4*.

Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Set the link's `pool` property to the pool of CPUs that you created for the zone.**

   - **If the VNIC has not yet been created, use the following syntax:**

     ```
     $ dladm create-vnic -l link -p pool=pool VNIC
     ```

   - **If the VNIC exists, use the following syntax:**

     ```
     $ dladm set-linkprop -p pool=pool VNIC
     ```

2. **Set the zone to use the VNIC.**

   ```
   global$ zonecfg -z zone
   zonecfg:zone> add net
   zonecfg:zone:net> set physical=VNIC
   zonecfg:zone:net> end
   ```

3. **Verify and commit the changes you have implemented and then exit the zone.**

   ```
   zonecfg:zone> verify
   zonecfg:zone> commit
   zonecfg:zone> exit
   ```

Example 77 Assigning a Link's CPU Pool to a Zone

This example is based on the configuration in Figure 22, "`pool` Property of a VNIC Assigned to a Zone," on page 183 where `pool99` has already been configured for the zone. The pool is then assigned to a VNIC. Finally, the non-global zone `zone1` is set to use the VNIC as the network interface.

```
$ dladm create-vnic -l net1 -p pool=pool99 vnic1

$ zonecfg -z zone1
```

```
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
```

▼ How to Allocate CPUs to a Datalink

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **Verify the CPU assignments for the interface.**

   ```
   $ dladm show-linkprop -p cpus link
   ```

2. **Assign CPUs to the link.**

   A list of CPUs that process packets for the datalink. Interrupts for the datalink might also be targeted to one of the CPUs in the list.

   ```
   $ dladm set-linkprop -p cpus=cpu1,cpu2,... link
   ```

   cpu1,cpu2,... Refers to the CPU number that you want to assign to the link. You can dedicate multiple CPUs to the link.

3. **(Optional) Display the CPUs that are associated with the link.**

   ```
   $ dladm show-linkprop -p cpus link
   ```

Example 78 Allocating CPUs to a Datalink

This example shows how to dedicate specific CPUs to the datalink `net0`.

```
$ dladm show-linkprop -p cpus net0
LINK     PROPERTY     PERM    VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     cpus         rw      --           0-2          --        --
```

The output shows that the system has implicitly assigned three CPUs (`0-2`) to the datalink `net0`. However, the CPUs are not exclusively allocated to the datalink `net0`.

```
$ dladm set-linkprop -p cpus=0,1 net0
$ dladm show-linkprop -p cpus net0
LINK     PROPERTY     PERM    VALUE        EFFECTIVE    DEFAULT   POSSIBLE
```

```
net0     cpus         rw      0-1          0-1          --        --
```

The output shows that you have explicitly assigned two CPUs (`0-1`) to the datalink `net0`. The allocated CPUs will process packets for the datalink `net0`.

Using the Large Receive Offload Feature in Oracle Solaris

In Oracle Solaris, you can use the large receive offload (LRO) feature to merge successive incoming packets into a single packet before the packets are delivered to the IP layer. The incoming packets must share the same transport protocol, local or remote IP address, and port number. This set of attributes are also known as five-tuple. If most of the packets share the same five-tuple, the packet processing overhead in the IP layer and the layers above it reduces, thereby improving network throughput. Typically, the TCP links under heavy load contain packets that share the same five-tuple.

Benefits of Using the LRO Feature

In Oracle Solaris, the merging of the packets is implemented in the MAC layer. The NIC delivers the packets to the networking stack and the MAC layer merges the successive incoming packets that share the same five-tuple information.

The LRO feature in Oracle Solaris provides the following benefits:

- Significantly improves the system's receive-side TCP performance. This improvement is higher in the kernel zone environment.
- Allows you to enable LRO on a physical NIC that supports the LRO feature.
- Allows you to selectively enable or disable LRO for each datalink in the host that includes a physical NIC, VNIC or `anet` resource, and paravirtualized (PV) NIC or SR-IOV VF within the Oracle Solaris Kernel Zone.

Enabling LRO for Datalinks

You can administer the LRO feature on both physical NICs and VNICs. You can use the `lro` property to enable or disable the LRO feature on a per-VNIC basis. By default, the `lro` property of the VNIC is inherited from the underlying datalink. Because the default value for the `lro`

datalink property of a physical NIC is `off`, the `lro` property that is inherited by the VNICs from the physical NIC is disabled by default.

You can enable LRO on a datalink in the following ways:

- When you enable the `lro` link property on a physical NIC, LRO is enabled for the primary MAC client of the NIC. If the `lro` property is set to `auto` for other MAC clients such as VNICs configured on the NIC, the effective value of the `lro` property for the MAC clients is inherited from the NIC.
- When you enable the `lro` link property on a MAC client such as a VNIC, the `lro` property is enabled for only the VNIC.

Similarly, you can disable LRO on a datalink by using the `lro` link property.

You can enable or disable the `lro` link property on any network device such as a VNIC, a physical NIC, a SR-IOV VF, a link aggregation, or a PV NIC in a guest domain. You can enable or disable LRO for a network device by setting the `lro` property. You use the following command syntax to set the `lro` property for a datalink:

```
$ dladm set-linkprop -p lro=value link
```

For a VNIC, you can specify the following values for the `lro` property:

`on` Enables LRO for the VNIC MAC client.

`off` Disables LRO for the VNIC MAC client.

`auto` Inherits the effective value of the lower datalink and enables merging of the packets if LRO is effective on the lower datalink. This value is the default value.

For a physical NIC, you can specify the following values for the `lro` property:

`on` Enables LRO for the primary MAC client of the NIC. The other MAC clients of the NIC inherit the state of the `lro` property if the value of the MAC client's `lro` property is `auto`.

`off` Disables LRO for the primary MAC client of the NIC. The other MAC clients of the NIC inherit the state of the `lro` property if the value of the MAC client's `lro` property is `auto`.

`auto` Default value of the `lro` property. When you set the `lro` property to `auto`, the effective value of the `lro` property is `off`.

For a PV NIC, you can specify the following values for the `lro` property:

`on`	Enables LRO for the primary MAC client of the PV NIC.
`off`	Disables LRO for the primary MAC client of the PV NIC.
`auto`	Inherits the effective value of the lower shadow VNIC and enables merging of packets on the shadow VNIC if LRO is enabled.

Note - The SR-IOV VF in a guest domain is considered to be like a physical NIC. When you set the `lro` property to `auto`, the effective value of the property is `off`.

EXAMPLE 79 Enabling LRO for a Physical NIC

The following example shows how to enable LRO for the physical NIC `net0` and check the status of the `lro` property.

```
$ dladm set-linkprop -p lro=on net0
$ dladm show-linkprop -p lro net0
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     lro             rw   on           on           auto      on,off,auto
```

EXAMPLE 80 Enabling LRO for PV NICs and `anet` Resources

The following example shows how to enable LRO for the PV NIC `net0`.

```
$ dladm set-linkprop -t -p lro=on net0
```

The following example shows the LRO status of a PV VNIC for a kernel zone after the value for the `lro` property is set to `on`.

```
$ dladm show-linkprop -p lro zone1/net0
LINK            PROPERTY        PERM    VALUE   EFFECTIVE       DEFAULT POSSIBLE
zone1/net0      lro             rw      on      on              auto    on, off, auto
```

The following example shows the default LRO status of a PV VNIC for a kernel zone if the effective value is `on` for the `lro` property of the lower shadow VNIC.

```
$ dladm show-linkprop -p lro zone1/net0
LINK            PROPERTY        PERM    VALUE   EFFECTIVE       DEFAULT POSSIBLE
zone1/net0      lro             rw      auto    on              auto    on, off, auto
```

The following example shows how to enable LRO for the `anet` resource `z1/net0` of the native zone `z1`.

```
$ dladm set-linkprop -t -p lro=on z1/net0
```

LRO Support for Link Aggregations

The LRO feature is supported for link aggregations. For information about link aggregations, see Chapter 2, "Configuring High Availability by Using Link Aggregations" in *Managing Network Datalinks in Oracle Solaris 11.4*.

Use the following command syntax to set the `lro` property for a link aggregation:

```
$ dladm set-linkprop -p lro=value  aggr
```

For a link aggregation, you can specify the following values for the `lro` property:

`on`	Enables LRO for the primary MAC client of the link aggregation.
`off`	Disables LRO for the primary MAC client of the link aggregation.
`auto`	Default value of the `lro` property. When you set the `lro` property to `auto`, the effective value of the `lro` property is `off`.

Note - The default setting on the link aggregation disables LRO on the aggregated datalinks even though LRO is enabled. To enable LRO on the aggregated datalinks, you need to explicitly set the `lro` property to `on`.

LRO Support for Zones

You can configure the `lro` link property for the `anet` resource of a kernel zone by setting the `lro` property by using the `zonecfg` command.

For the `anet` resource of a native zone, you can set the `lro` property by using the `dladm` command. See Example 80, "Enabling LRO for PV NICs and `anet` Resources," on page 189.

EXAMPLE 81 Enabling LRO for a Kernel Zone

This example shows how to enable LRO for the `anet` resource of the kernel zone `kzone1`.

```
$ zonecfg -z kzone1
zonecfg:kzone1> select anet id=1
```

```
zonecfg:kzone1:anet> set lro=on
```

To show the status of the `lro` property for the `anet` resource `kzone1/net1` of the kernel zone `kzone1`:

```
$ dladm show-linkprop -p lro kzone1/net1
LINK        PROPERTY   PERM VALUE    EFFECTIVE    DEFAULT   POSSIBLE
kzone1/net1 lro        rw   on       on           auto      on,off,auto
```

Managing Network Resources by Using Flows

A flow is a customized way of categorizing network packets based on a single attribute or a combination of attributes. Flows enable you to further allocate network resources. For an overview of flows, see "Managing Network Resources Through Flows" on page 31.

Using flows for managing network resources involves the following steps:

1. Creating the flow.

   A flow is created based on attributes that are derived from the information in a packet's header, such as the IP address, transport protocol, and DS field. Starting with Oracle Solaris 11.3, you can create flows that have different combination of attributes on a datalink.
2. Customizing the flow's use of resources by setting properties that pertain to network resources. Currently, bandwidth, priority, and rank properties can be associated with flows.

Commands for Resource Allocation in Flows

- To simultaneously create a flow and add resources to it, use the following command syntax:

  ```
  $ flowadm add-flow -l link \
      -a attribute=value[,attribute=value] \
      -p prop=value[,...] flow
  ```

  The set of defined attributes that characterizes the flows constitutes the system's *flow control policy*.
- To set the property of an existing flow, use the following command syntax:

  ```
  $ flowadm set-flowprop -p prop=value[,...] flow
  ```

  prop refers to the flow properties that can be assigned to a flow. The flow properties are the same as the properties that are assigned directly to a link. However, only the bandwidth and priority properties can be associated with flows.

For more information, see the `flowadm`(8) man page.

Configuring Flows

This section describes how to create flows and set flow properties to implement resource control.

▼ How to Configure Flows

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 34.

1. **(Optional) List the available links to determine the link on which you will configure flows.**

   ```
   $ dladm show-link
   ```

2. **Verify that IP interfaces over the selected link are properly configured with IP addresses.**

   ```
   $ ipadm show-addr
   ```

3. **Create flows according to the attribute you have determined for each flow.**

   ```
   $ flowadm add-flow -l link -a attribute=value[,attribute=value] flow
   ```

   link — Refers to the link on which you are configuring the flow.

   attribute — A single attribute or combination of attributes that organizes network packets into a flow.

   The direction attribute for a flow enables the classification of packets and implementing properties for flows on inbound or outbound packets. The `direction` attribute supports the values `in` for inbound only, `out` for outbound only, and `bi` for bidirectional. The default value for this attribute is `bi`. See Example 82, "Creating a Flow With the `direction` Attribute," on page 193.

   flow — Refers to the name that you assign to the flow.

   For more information about flows and flow attributes, see the `flowadm`(8) man page.

4. **(Optional) Display the possible range of values for the link's bandwidth.**

```
$ dladm show-linkprop -p max-bw link
```

link Refers to the datalink on which the flow is configured.

The range of values is listed under the `POSSIBLE` field of the command's output.

5. **Implement resource controls on the flows by setting the appropriate flow properties.**

```
$ flowadm set-flowprop -p prop=value[,...] flow
```

For information about the properties that you can set for flows, see "Setting Properties for Flows" on page 194.

6. **(Optional) Display the flows that you have created over the datalink.**

```
$ flowadm
```

Note - The `flowadm` command, if used without any subcommand, provides the same information as the `flowadm show-flow` command.

7. **(Optional) Display the property values for a specified flow.**

```
$ flowadm show-flowprop flow
```

This command displays `max-bw`, `priority`, and `hw-flow` properties.

For an example about configuring flows and setting flow properties, see "Use Case: Managing Network Resources by Setting Datalink and Flow Properties" on page 201.

Example 82 Creating a Flow With the `direction` Attribute

The following example shows how to create the flows `http-in` and `http-out` by specifying the `direction` attribute.

```
$ flowadm add-flow -l net4 -a transport=tcp,local_port=80,direction=in http-in
$ flowadm add-flow -l net4 -a transport=tcp,local_port=80,direction=out http-out
$ flowadm
FLOW        LINK     PROTO LADDR   LPORT  RADDR   RPORT DIR
http-out    net4     tcp   --      80     --      --    out
http-in     net4     tcp   --      80     --      --    in
$ flowadm show-flow -o flow,link,dir
FLOW        LINK     DIR
http-out    net4     out
http-in     net4     in
```

The `direction` value of `bi` is incompatible with the values `in` or `out`. If you try create a flow with the same attributes, the flow creation fails.

```
$ flowadm add-flow -l net4 -a transport=tcp,local_port=80 http-flow
flowadm: add flow failed: a flow with identical attributes but with
incompatible direction exists
```

You can create a flow with a different attribute and it will succeed.

```
$ flowadm add-flow -l net4 -a transport=tcp,local_port=443 ssl-flow
$ flowadm
FLOW        LINK     PROTO LADDR   LPORT RADDR  RPORT DIR
http-out    net4     tcp   --      80    --     --    out
http-in     net4     tcp   --      80    --     --    in
ssl-flow    net4     tcp   --      443   --     --    bi
```

Setting Properties for Flows

You implement resource controls over flows by setting flow properties. Flows support the following properties:

- `max-bw` – The maximum bandwidth of the datalink that the packets identified with a flow can use. You must set a value within the allowed range of values for the bandwidth of a datalink. See Example 83, “Setting Maximum Bandwidth and Priority for a Flow,” on page 195.
- `priority` – The priority with which packets belonging to the specified flow will be processed. The allowed values for the `priority` property are `high`, `medium`, and `low`. If the priority of a flow is set to `high`, all the packets belonging to that flow will be processed ahead of other packets on the same link. This property is used to create a flow for applications that are latency sensitive. The default value of this property is `medium`. See Example 83, “Setting Maximum Bandwidth and Priority for a Flow,” on page 195.

  Note - Currently, setting the `priority` property to `low` from `medium` has no effect.

- `hw-flow` – Specifies whether the flow is offloaded to the underlying NIC. The possible values of this property are `auto`, `on`, and `off`.
  - `on` means that the flow has been offloaded to the NIC and packet classification for the flow is conducted at the hardware level.
  - `auto`, the default setting, means that the system decides if the flow can be offloaded to the NIC based on the capability of the NIC.
  - `off` means the flow is not offloaded.

Note - If the `hw-flow` property is set to `on`, offloading might fail in some cases such as VNIC migration or VNIC failover on a DLMP aggregation if the destination port does not support flow offload. Retaining the default `auto` setting is a safer configuration.

Currently, only the flows that are defined by specifying all the transport protocols, local or remote IP address, and local or remote port can be assigned the value `on` for `hw-flow`. Also, not all NICs support the `hw-flow` property.

- `rank` – The rank for a flow which you can optionally set. A flow with no rank setting is lower in the lookup order than one with a rank. Rank values range from 1 to 65535. Low numbers place flows higher in the lookup order than high numbers. If multiple flows have overlapping rank values, the tie is broken by following the default system policy. See "Overlapping Flows" on page 196.

EXAMPLE 83 Setting Maximum Bandwidth and Priority for a Flow

This example shows how to set the maximum bandwidth to `2G` and a `high` priority for the flow `http`.

```
$ flowadm set-flowprop -p max-bw=2G http
$ flowadm set-flowprop -p priority=high http
```

EXAMPLE 84 Setting `hw-flow` for a Flow

This example shows how to set the `hw-flow` property for a flow when the NIC is capable and also when the NIC is not capable of offloading the flows.

```
$ flowadm show-flowprop -p hw-flow
FLOW        PROPERTY   PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
ssh-flow    hw-flow    rw   auto         off          auto      auto,off

# flowadm set-flowprop -p hw-flow=on ssh-flow
flowadm: failed: cannot set flow property 'hw-flow' on 'ssh-flow': operation not
 supported
```

The `POSSIBLE` field shows only the values `auto` and `off`, which means that the NIC is not capable of offloading the flows. Therefore, you cannot set the `hw-flow` property to `on`.

```
$ flowadm show-flowprop -p hw-flow
FLOW        PROPERTY   PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
http-flow   hw-flow    rw   auto         on           auto      auto,on,off

$ flowadm set-flowprop -p hw-flow=on http-flow
```

```
$ flowadm show-flowprop -p hw-flow
FLOW        PROPERTY   PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
http-flow   hw-flow    rw   on           on           auto      auto,on,off
```

Overlapping Flows

When multiple flows are configured on a datalink with different attributes, the flows might overlap. In this case, you can use the `flowadm show-flow` command to display a list of flows on a datalink based on a default ranking order. That is, the first flow in the output is searched first for a given packet and then the next flow is searched. You can change the ranking order of a flow by using the `rank` property.

For example, assume that you have created the flow `solaris` to limit the traffic from a remote IP address as follows.

```
$ flowadm add-flow -l net4 -a remote_ip=192.0.2.3 solaris
$ flowadm set-flowprop -p max-bw=10K solaris
```

If you want a packet from the IP address `192.0.2.0` to port 80 to match `solaris` instead of the `http` flow, you can set a high rank for the `solaris` flow as follows:

```
$ flowadm set-flowprop -p rank=1 solaris
$ flowadm show-flowprop -p rank solaris
FLOW     PROPERTY  PERM VALUE  EFFECTIVE   DEFAULT    POSSIBLE
solaris  rank      rw    1         1         --       1-65535
```

Use the `flowadm match-flow` command to check whether a flow that you want to create overlaps with other existing flows. If flows overlap, check the ranking order. Also, if you have a policy in place to disallow the creation of overlapping flows, check before adding a flow.

```
$ flowadm match-flow [-P] [[-p] -o  field[,...]] \
   [-l link] \
   -a attr=value[,...]
```

`-l` *link* — Limits the match to flows on the specified link. If you do not specify a link, flows on all the links are used.

`-a` *attr*`=`*value*`[,...]` — A comma-separated list of attributes that are used as the key for the lookup for a matching flow or flows.

EXAMPLE 85 Checking the Overlapping Flows

This example shows how to check whether an added flow overlaps with other flows.

The following example displays a flow configured on a system.

```
$ flowadm
FLOW     LINK     PROTO LADDR   LPORT RADDR  RPORT DIR
http     net4     tcp   --      80    --     --    bi
```

When you want to add a another flow `backup` with the remote IP address `203.0.113.117` on the datalink `net4`, you can check whether the `backup` flow overlaps with other flows as follows.

```
$ flowadm match-flow -l net4 -a remote_ip=192.0.2.4
FLOW   LINK     PROTO LADDR  LPORT RADDR  RPORT DIR
http   net4     tcp   --     80    --     --    bi
```

The output shows that the flows `http` and `backup` can overlap for certain packets.

Configuring Flows With Dedicated Ring Groups

Flows share the resources such as rings and interrupts of the underlying datalink. Flows can also have their own dedicated ring groups to perform flow classification by using the underlying physical datalink. Currently, only the Intel XL710 10/40 Gigabit Ethernet controller NIC supports configuring flows with dedicated ring groups. The dedicated ring groups provides the following benefits for the flows:

- Provides hardware SLA enforcement
- Improves data path performance
- Saves CPU utilization of flows
- Improves resource isolation

▼ How to Create a Flow With a Dedicated Ring Group

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See “Using Rights Profiles to Perform Network Configuration” on page 34.

1. Check if the underlying datalink supports dedicated (exclusive) ring group.

a. Display the `ring-group` property of a datalink to ensure that the datalink supports exclusive ring group.

```
$ dladm show-linkprop -p ring-group datalink
```

If the underlying datalink supports dedicated ring group, the value of the `ring-group` property is displayed as `exclusive`. If the underlying link does not support dedicated ring group, you cannot add a flow with a dedicated ring group to the underlying datalink.

b. Check the ring group resource availability for the underlying datalink.

```
$ dladm show-phys -G datalink
```

2. Create a flow by setting the `hw-flow` property to `on` and setting the other required attributes.

```
$ flowadm add-flow -l datalink -a transport=trans-protocol,local_ip=IPaddr, \
    local_port=port,remote_ip=IPaddr,remote_port=port -phw-flow=on flowname
```

Note - You can only create flows with the following combination of attributes for the NIC, XL710:

- With the transport protocol `udp`, you need to set the attributes `local_ip` and `local_port`.
- With the transport protocol `tcp`, you need to set the attributes `local_ip`, `local_port`, and `remote_port`.

For more information, see the `i40e`(4D) man page.

---

3. Display the flow and the flow properties.

```
$ flowadm
```

```
$ flowadm show-flowprop flowname
```

Example 86 Creating a Flow With Dedicated Ring Group

The following example shows how to create a flow with dedicated ring group.

```
$ dladm show-linkprop -p ring-group net2
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net2     ring-group      r-   exclusive    exclusive      --      shared
                                                                  exclusive
```

You can add flows with exclusive ring group to a datalink only if the underlying datalink supports exclusive ring group.

Check the resource availability for the ring group of the underlying datalink.

```
$ dladm show-phys -G net2
LINK            RG-AVAIL    RG-INUSE-UMAC RG-INUSE-VNIC RG-INUSE-FLOW
net2            268         0             0             0

$ flowadm add-flow -l net3 -a transport=tcp,local_ip=203.0.113.2/27, \
local_port=5002,remote_ip=203.0.113.35/27,remote_port=5001 -p hw-flow=on tcpflow1
```

```
$ flowadm
FLOW        LINK     PROTO LADDR             LPORT RADDR             RPORT  DIR
tcpflow1    net3     tcp   203.0.113.2/27    5002  203.0.113.35/27   5001   bi

$ flowadm show-flowprop tcpflow1
FLOW        PROPERTY   PERM VALUE      EFFECTIVE    DEFAULT   POSSIBLE
tcpflow1    max-bw     rw   --         --           --        --
tcpflow1    dscp       rw   --         --           --        0-63
tcpflow1    hw-flow    rw   on         on           auto      auto,off
tcpflow1    bw-share   rw   --         --           --        1-100
tcpflow1    ofaction   rw   --         --           --        --
```

Configuring Bandwidth Share for the Flows With Dedicated Ring Group

You can set the bandwidth share for a flow only if the underlying datalink supports offloading of SLAs and the flow has a dedicated ring group. You can specify the bandwidth share by using the `bw-share` property while creating the flow. All the hardware resource consumers of the physical datalink such as VF VNIC, ring group VNIC, and hardware flows share the total bandwidth of the datalink.

Note - Datalinks such as i40e support only the transmit side of the hardware SLA offloading.

EXAMPLE 87 Setting Bandwidth Share for the Flows With Dedicated Ring Groups

The following examples show how to set the bandwidth share for the hardware flows with dedicated ring groups.

Check whether the underlying datalink support offloading of hardware SLAs by using the `dladm show-linkprop` command.

```
$ dladm show-linkprop -H -pbw-share net3
LINK     PROPERTY        MODE HWPOSSIBLE    HWFLAGS SWPOSSIBLE    SWFLAGS
net3     bw-share        none 1-100         o       --            --
$ dladm show-linkprop -H -pmax-bw net3
LINK     PROPERTY    MODE   HWPOSSIBLE    HWFLAGS SWPOSSIBLE    SWFLAGS
net3     max-bw      sw     50-10000:50   o       0-10000:0.001 oi
```

Set the bandwidth share for the flow while creating the flow by using the `bw-share` property.

```
$ flowadm add-flow -lnet3 -a transport=tcp,local_ip=203.0.113.2/27, \
local_port=5002,remote_ip=203.0.113.35/27,remote_port=5001 -p hw-flow=on,bw-share=10
 tcpflow1
```

You can set the bandwidth share for the existing flows by using the `flowadm set-flowprop` command.

```
$ flowadm set-flowprop -p bw-share=10 tcpflow1

 $ flowadm set-flowprop -p bw-share=40 udpflow1

$ flowadm show-flowprop -p bw-share
FLOW        PROPERTY   PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
tcpflow1    bw-share    rw   10           20%           --        1-100
udpflow1    bw-share    rw   40           80%           --        1-100
```

Configuring a Flow With a Group of Flow Filters

In addition to configuring a flow that consists of one flow filter (tuple of L3/L4 attribute), you can now configure a flow that consist of multiple flow filters. The resource control properties apply only to the flow. You must define all the filters for a flow with the same combination of attributes. You use the `flowadm` and `flowstat` commands to manage and display the flow filters. Typically, flow filters are created from an application by using the flow socket APIs. Hence, the CLI support is mainly useful for observability.

Viewing Flow Filter Information

To view a flow filter, use any of the following command syntaxes:

- `# flowadm show-filter flow-name`
- `# flowadm show-flow -v`

The following example shows how to display the filters that are added by an application.

```
$ nc -M flow=webserver 203.0.113.10/27 80 &
$ nc -M flow=webserver 203.0.113.10/27 8080 &
```

Use any of the following commands to view the flow filters:

- `$ flowadm show-filter webserver`

  ```
  FILTER      PROTO LADDR           LPORT RADDR           RPORT DIR  _HWFILTER
  webserver@2 tcp   203.0.113.65/27 38016 203.0.113.10/27 8080  bi    off
  webserver@1 tcp   203.0.113.65/27 36172 203.0.113.10/27 80    bi    off
  ```

- `$ flowadm show-flow -v`

  ```
  FLOW           LINK PROTO LADDR           LPORT RADDR           RPORT DIR
  webserver      net4 --     --         --    --          --    --
  ```

```
webserver@2 net0 tcp   203.0.113.65/27 38016 203.0.113.10/27 8080  bi
webserver@1 net0 tcp   203.0.113.65/27 36172 203.0.113.10/27 80    bi
```

You can also use the `flowstat` command to display the statistics of the flow filters.

For example,

```
$ flowstat -v
FLOW        IPKTS RBYTES IDROPS OPKTS OBYTES ODROPS
webserver   26    1.74K   25     24    1.61K  0
webserver@2 16    1.07K   13     15    1.00K  0
webserver@1 10     674     12     9     602   0
```

For information about how to add or remove a flow filter, see the `flowadm`(8) man page. For more information about flow statistics, see the `flowstat`(8) man page.

Creating a Flow Without a Filter

You might need to create a flow without any filter, that is, without any attributes for some applications. The application adds the flow filters dynamically and you can configure SLAs for the flow.

This example consists of the creation of a filterless flow and the subsequent setup of a flow property.

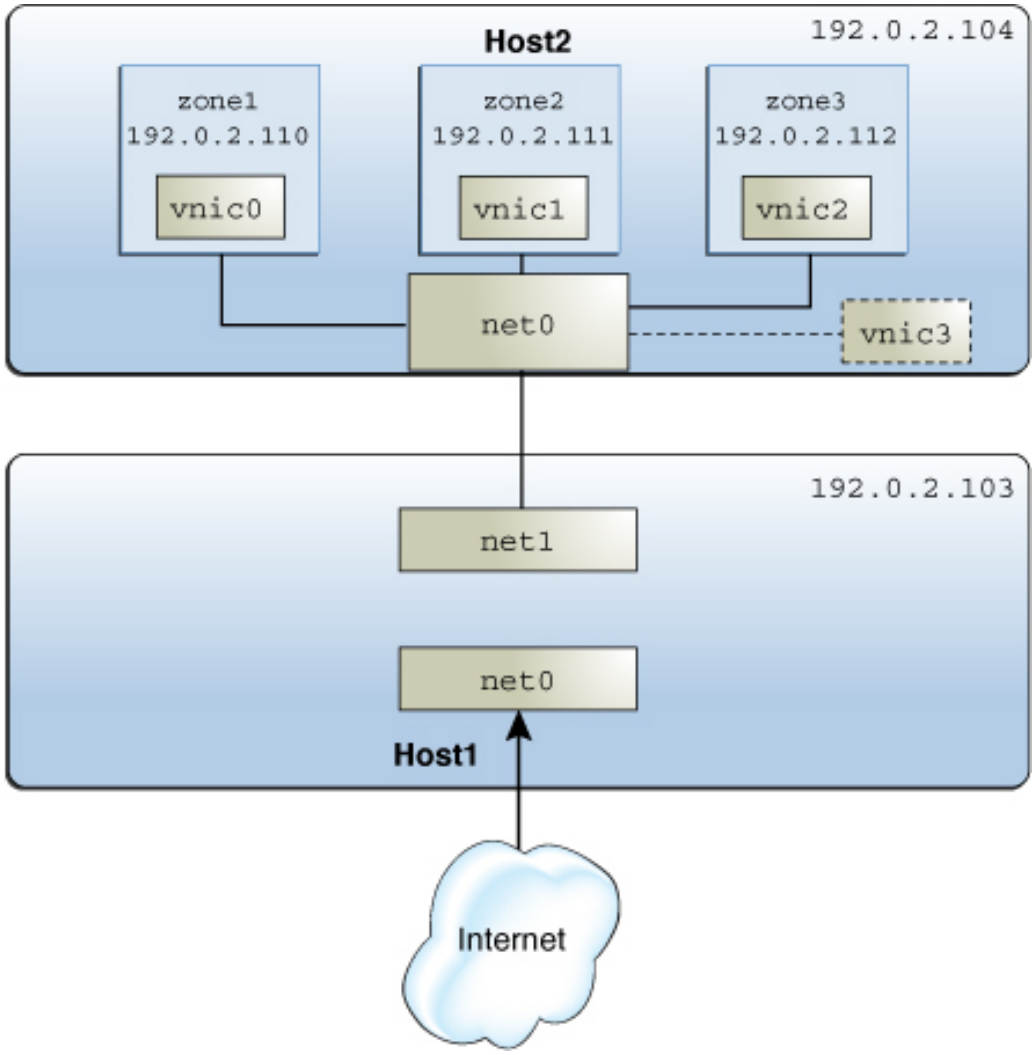
```
$ flowadm add-flow -l net4 webserver
$ flowadm
FLOW LINK PROTO LADDR LPORT RADDR RPORT DIR
webserver net4   --   --    --    --    bi

$ flowadm set-flowprop -p max-bw=1G webserver
$ flowadm show-flowprop -p max-bw
FLOW     PROPERTY PERM VALUE EFFECTIVE  DEFAULT POSSIBLE
ebserver max-bw   rw   1000  1000       --      --
```

Use Case: Managing Network Resources by Setting Datalink and Flow Properties

The following use case is based on a scenario in which you increase a system's efficiency by setting both datalink and flow properties. This use case is based on the configuration shown in the following figure.

FIGURE 24 System Configuration for Managing Resources on Datalinks and Flows



The figure shows the following two physical hosts that are connected to each other:

- `Host1` has the following configuration:

  - One non-global zone that functions as a server and router. Two interfaces are assigned to the zone: the `net0` interface connects to the Internet and the `net1` interface connects to the internal network including the `Host2`.
  - Flows are configured over `net1` to isolate the traffic and implement control over how packets belonging to the flows use resources. For information about configuring flows, see "Managing Network Resources by Using Flows" on page 191.
- `Host2` has the following configuration:
  - Three non-global zones and their respective VNICs. The VNICs are configured over `net0`, whose NIC card supports ring allocation. For more information about ring allocation, see "Managing NIC Rings" on page 176.
  - Each zone's network processing load is different. In this example, `zone1` functions as the HTTP client. The remaining zones, `zone2` and `zone3`, function as the secure shell (SSH) client that tries to access `Host1` through SSH protocol. The network traffic for `zone1` is higher than `zone2` and `zone3` and is not time sensitive. However, the network traffic for `zone2` and `zone3` is low and time sensitive. Therefore, to process the network traffic faster for `zone2` and `zone3`, you need to limit the bandwidth allocated to the network traffic for `zone1`. If the bandwidth allocated for `zone1` is not limited, it will use all the available bandwidth. This leads to the denial of bandwidth to the remaining zones: `zone2` and `zone3`.
  - A separate VNIC is configured as a software-based client. For an overview of client types, see "Configuring Clients and Allocating Rings" on page 177.

The tasks in this use case involve the following actions:

- Creating a flow and configuring flow control – Flows are created over `net1` to create a separate resource control over packets belonging to the flows that are received by `net1` of `Host1`.
- Configuring network resource properties for the VNICs on `Host2` – Based on the processing load, each zone's VNIC is configured with a set of dedicated rings. A separate VNIC is also configured without dedicated rings as an example of a software-based client.

Note - The use case does not include any procedures for zone configuration. To configure zones, see Chapter 2, "Setting Up a Non-Global Zone" in *Creating and Using Oracle Solaris Zones*.

```
$ ipadm
NAME              CLASS/TYPE  STATE    UNDER      ADDR
lo0               loopback    ok       --         --
   lo0/v4         static      ok       --         127.0.0.1/8
   lo0/v6         static      ok       --         ::1/128
net1              ip          ok       --         --
   net1/v4        static      ok       --         192.0.2.103/24
```

```
net0              ip          ok       --         --
   net0/v4        static      ok       --         203.0.113.129/24

$ flowadm add-flow -l net1 -a transport=tcp,local_ip=192.0.2.103, \
   local_port=80,remote_ip=192.0.2.110 httpflow

$ flowadm add-flow -l net1 -a transport=tcp,local_ip=192.0.2.103, \
   local_port=22 sshflow

$ flowadm set-flowprop -p max-bw=500M httpflow

$ flowadm set-flowprop -p priority=high sshflow

$ flowadm
FLOW        LINK     PROTO LADDR         LPORT  RADDR         RPORT DSFLD
httpflow    net1     tcp   192.0.2.103   80     192.0.2.110   --    --
sshflow     net1     tcp   192.0.2.103   22     --            --    --

$ flowadm show-flowprop
FLOW         PROPERTY        PERM    VALUE        DEFAULT        POSSIBLE
httpflow     maxbw           rw      500          --             --
httpflow     priority        rw      medium       medium         low,medium,high
httpflow     hwflow          r-      off          --             on,off
sshflow      maxbw           rw      --           --             --
sshflow      priority        rw      high         medium         low,medium,high
sshflow      hwflow          r-      off          --             on,off

$ dladm create-vnic -l net0 vnic0
$ dladm create-vnic -l net0 vnic1
$ dladm create-vnic -l net0 vnic2

$ dladm set-linkprop -p rx-rings=4,tx-rings=4 vnic0
$ dladm set-linkprop -p rx-rings=2,tx-rings=2 vnic1
$ dladm set-linkprop -p rx-rings=1,tx-rings=1 vnic2

$ zonecfg -z zone1
# zonecfg:zone1> add net
# zonecfg:zone1:net> set physical=vnic0
# zonecfg:zone1:net> end
# zonecfg:zone1> commit
# zonecfg:zone1> exit
$ zoneadm -z zone1 reboot

$ zonecfg -z zone2
# zonecfg:zone2> add net
# zonecfg:zone2:net> set physical=vnic1
# zonecfg:zone2:net> end
# zonecfg:zone2> commit
```

```
# zonecfg:zone2> exit
$ zoneadm -z zone2 reboot

$ zonecfg -z zone3
# zonecfg:zone3> add net
# zonecfg:zone3:net> set physical=vnic2
# zonecfg:zone3:net> end
# zonecfg:zone3> commit
# zonecfg:zone3> exit
$ zoneadm -z zone3 reboot

$ dladm create-vnic -p rx-rings=sw,tx-rings=sw -l net0 vnic3

$ dladm set-linkprop -p pool=pool1 vnic0
```

♦ ♦ ♦ CHAPTER 8

Monitoring Network Traffic and Resource Usage

This chapter describes tasks for monitoring network statistics about the use of network resources on datalinks and flows. You configure network accounting on a system to record network traffic statistics in a log file. This statistical information can help you analyze resource allocation for provisioning, consolidation, and billing purposes. This chapter introduces the two commands that you can use to display network traffic statistics: `dlstat` and `flowstat`.

This chapter contains the following topics:

For more information about the observing network traffic usage on various layers of the network protocol stack, see Chapter 2, "Using Observability Tools to Monitor Network Traffic Usage" in *Troubleshooting Network Administration Issues in Oracle Solaris 11.4*.

Note - To perform monitoring tasks that might involve issuing privileged commands, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 34.

Overview of Monitoring Network Traffic Statistics of Datalinks and Flows

Packets traverse a path when they flow into or out of a system. On a granular level, packets are received and transmitted through receive (Rx) rings and transmit (Tx) rings of a NIC. Inbound packets from these rings are passed up the network stack for further processing while outbound packets are sent to the network.

You can combine and allocate system resources to manage the network traffic. You can monitor the receive-side and transmit-side network traffic statistics for both datalinks and flows. This chapter focuses primarily on receive-side network traffic statistics on datalinks and flows.

You can configure receive rings, transmit rings, and other resources on datalinks by setting datalink properties. Depending on the network traffic on a datalink, you can assign dedicated hardware rings to a datalink to increase the system's efficiency to process packets. For example, you can allocate more rings to a datalink, where the network traffic is most heavy. For more information about how to allocate hardware rings to a datalink, see "Configuring Clients and Allocating Rings" on page 177.

A datalink might not have dedicated hardware rings because of the following reasons:

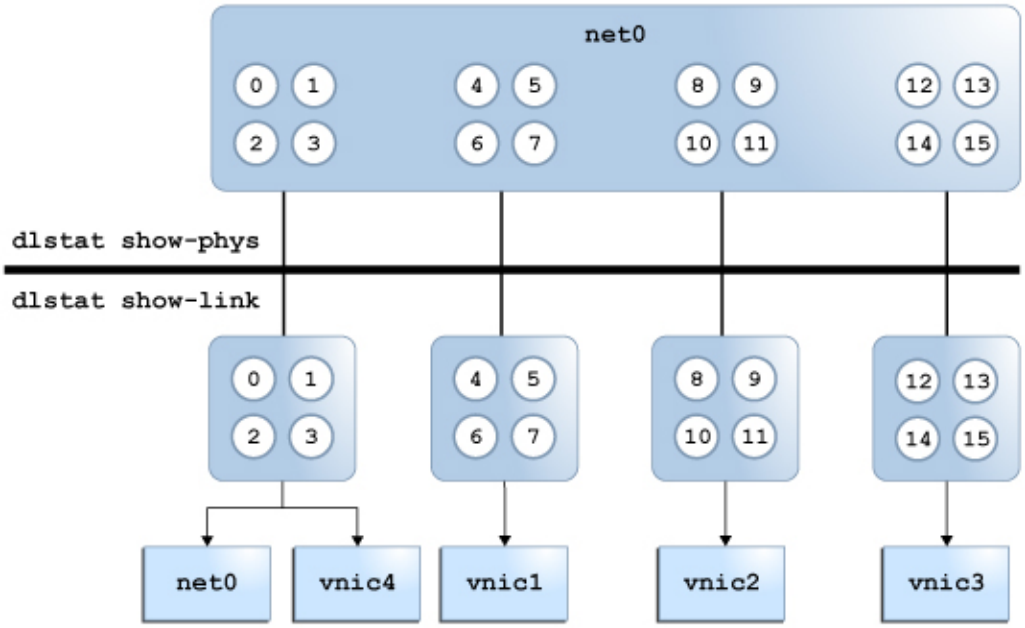
- Lack of hardware resources. For example, there might not be rings available that can be exclusively assigned to datalinks.
- Lack of hardware capabilities. For example, the NIC does not expose hardware rings.
- The datalink might not be tied to a lower hardware datalink. For example, when you create VNICs over etherstubs.

Some datalinks might be configured to share rings for the following reasons:

- The datalink might not be performing intensive processes that require dedicated rings.
- The NIC might not support ring allocation.
- The rings are no longer available to be assigned for exclusive use although the datalink supports ring allocation.

The following figure shows the allocation of hardware rings among datalinks.

FIGURE 25 Ring Allocation in Datalinks



The figure shows the following configuration:

- The `net0` datalink has 16 hardware rings (`0-15`) that can be allocated to other datalinks.
- The VNICs `vnic1`, `vnic2`, `vnic3`, and `vnic4` are configured over the datalink `net0`.
- The VNICs `vnic1`, `vnic2`, and `vnic3` are each assigned four dedicated hardware rings.
- The hardware rings (`0-3`) are shared between the datalink `net0` and the VNIC `vnic4`. The following example shows the ring allocation for the physical datalink `net0`.

```
$ dladm show-phys -H net0
LINK         RINGTYPE  RINGS              CLIENTS
net0         RX        0-3                <default,mcast>,vnic4
net0         RX        4-7                vnic1
net0         RX        8-11               vnic2
net0         RX        12-15              vnic3
net0         TX        0-7                <default>,vnic4,vnic3,vnic2,vnic1
```

- You use the `dlstat show-phys` command to display the network traffic statistics for the physical datalink `net0`. See Example 88, “Displaying Traffic Statistics for Physical Links on the System,” on page 212.
- You use the `dlstat show-link` command to display the network traffic statistics for the datalinks `net0`, `vnic1`, `vnic2`, `vnic3`, and `vnic4`. See Example 95, “Displaying Network Traffic Statistics for a Datalink With Dedicated Hardware Rings,” on page 215.

Commands for Monitoring Network Traffic Statistics

The `dlstat` and `flowstat` commands enable you to monitor network traffic statistics on datalinks and flows, respectively. These commands are equivalent to the `dladm` and `flowadm` commands. The following table compares the functions of the pair of administrative commands to the pair of monitoring commands.

Administrative Commands		Monitoring Commands	
Command	**Function**	**Command**	**Function**
`dladm`	Configures and administers datalinks	`dlstat`	Displays traffic statistics on datalinks
`flowadm`	Configures and administers flows	`flowstat`	Displays traffic statistics on flows

Displaying Network Traffic Statistics of Links

You can use the following variants of the `dlstat` command to display network traffic information.

Command	Information Provided
`dlstat` [*link*] `dlstat -rt` [*link*] `dlstat show-link` [*link*]	Displays inbound and outbound traffic statistics per datalink
`dlstat show-link -rt` [*link*]	Displays inbound and outbound traffic statistics per ring per datalink
`dlstat show-phys` [*link*]	Displays inbound and outbound traffic statistics per network physical device

Command	Information Provided
`dlstat show-phys -rt` [*link*]	Displays inbound and outbound traffic statistics per ring per network physical device
`dlstat show-aggr` [*link*] `dlstat show-aggr -rt` [*link*]	Displays inbound and outbound traffic statistics per port per aggregation
`dlstat show-bridge` [*bridge*] `dlstat show-bridge -rt` [*bridge*]	Displays inbound and outbound traffic statistics per bridge
`dlstat show-cap` [*link*]	Displays statistics for packets that are logged by a firewall

You can use the `-r` option to display receive-side statistics information or the `-t` option to display the transmit-side statistics information with the `dlstat` command. For more information about other options, see the `dlstat(8)` man page.

Displaying Network Traffic Statistics of Network Devices

The `dlstat show-phys` command provides statistics that refer to the physical network device. As shown in Figure 25, “Ring Allocation in Datalinks,” on page 209, the `dlstat show-phys` command operates on the hardware rings which are on the device layer of the network stack.

You can use the following command syntax to display the network traffic statistics on network devices:

```
$ dlstat show-phys [-r|-t] [-Tu|-Td] \
   [-o idrops[,idropbytes][,odrops][,odropbytes]] \
   [link] [interval [count]]
```

`-r`
: Displays receive-side network traffic statistics only. You should not specify the `-t` option with this option.

  If you do not specify the `-r` option or the `-t` option, both the transmit-side and receive-side network statistics are displayed.

`-t`
: Displays transmit-side network traffic statistics only. You should not specify the `-r` option with this option.

  If you do not specify the `-r` or the `-t` option, both the transmit-side and receive-side network statistics are displayed.

`-Tu` Displays the current time in internal representation.

`-Td` Displays the current time in standard date format.

`-o [idrops[, idropbytes] [,odrops][, odropbytes]]` Displays the input and output packet drops per physical datalink. In addition to the number of input and output packet drops, this option displays the number of bytes of the drops.

link Name of the datalink whose network statistics you want to monitor. If you do not specify the datalink, then the information about all the configured datalinks on the system are displayed.

interval Specifies the time in seconds at which you want to refresh the network statistics.

count Specifies the number of times you want the displayed network traffic statistics to be refreshed. If you do not specify the count value, the statistics are refreshed indefinitely.

EXAMPLE 88 Displaying Traffic Statistics for Physical Links on the System

In this example, both incoming and outgoing network traffic on each link on the system is displayed. The number of packets and their byte sizes are displayed.

```
$ dlstat show-phys
LINK    IPKTS   RBYTES     OPKTS    OBYTES
net5        0        0         0         0
net6        0        0         0         0
net0   25.57K    5.10M     1.93K  226.05K
net0      179   26.63K       161   22.75K
net3        0        0         0         0
net4        0        0         0         0
net2        0        0         0         0
net8      238  137.16K       191    8.41K
net1        0        0         0         0
...
```

The output shows the following information:

`LINK` Physical or virtual datalink, identified by a name

`IPKTS` Number of inbound packets on the link

`RBYTES` Number of bytes received on the link

OPKTS	Number of outbound packets on the link
OBYTES	Number of bytes sent on this link

EXAMPLE 89 Displaying Receive-Side Traffic Statistics for Network Devices

In this example, network traffic statistics that are being received are displayed with an interval value of 2 seconds and the count value of 3.

```
$ dlstat show-phys -r 2 3
LINK  TYPE  INDEX    IPKTS   RBYTES
net0    rx      0    8.03M   12.09G
net1    rx      0        0        0
net0    rx      0    8.79K   13.28M
net1    rx      0        0        0
net0    rx      0    8.50K   12.83M
net1    rx      0        0        0
```

Consider the datalinks, net0 and net1 as a set. The first set of datalinks, net0 and net1, show the total number of packets and bytes received. In this example, 8.03M is the total number of packets received and 12.09G is the total number of bytes received by net0. The second set of datalinks, net0 and net1, show the network traffic statistics in rates per second, also known as the normalized value. That is, 8.79K is the normalized value of the packets received by net0 in the interval of 2 seconds. Similarly, the third set of datalinks, net0 and net1, also show the normalized value for the network traffic statistics in the interval of 2 seconds.

EXAMPLE 90 Displaying Receive-Side Traffic Statistics for a Network Device

In this example, the incoming traffic statistics for the datalink net0 are displayed.

```
$ dlstat show-phys -r net0
LINK     TYPE     ID    INDEX    IPKTS     RBYTES
net0       rx   local      --        0          0
net0       rx      hw       1        0          0
net0       rx      hw       2    1.73M      2.61G
net0       rx      hw       3        0          0
net0       rx      hw       4    8.44M     12.71G
net0       rx      hw       5    5.68M      8.56G
net0       rx      hw       6    4.99M      7.38G
net0       rx      hw       7        0          0
```

In this example, the net0 datalink has eight receive rings, which are identified under the INDEX field. An even distribution of packets per ring is an ideal configuration that indicates that the rings are properly allocated to links according to the link's load. An uneven distribution indicates a disproportionate distribution of rings per link. The resolution of the uneven

distribution depends on whether the NIC supports dynamic ring allocation. If it does, you can redistribute rings per link to process packets more evenly. For more information, see “Managing NIC Rings” on page 176.

EXAMPLE 91 Displaying Transmit-Side Traffic Statistics for a Network Device

In this example, the usage of the transmit rings for `net0` as a network device is displayed.

```
$ dlstat show-phys -t net0
LINK  TYPE  INDEX    OPKTS   OBYTES
net0    tx      0       93    4.63K
net0    tx      1        0        0
net0    tx      2        0        0
net0    tx      3        0        0
net0    tx      4        0        0
net0    tx      5       47   11.02K
net0    tx      6       23    7.13K
net0    tx      7        0        0
```

EXAMPLE 92 Displaying Traffic Statistics for a Network Device With Time

The following example displays statistics about network traffic for `net0` as a network device with internal representation of the current time.

```
$ dlstat show-phys -Tu net0
1401652481
          LINK    IPKTS   RBYTES    OPKTS   OBYTES
          net0      184   27.14K      165   22.91K
```

The following example displays statistics about network traffic for `net0` as a network device with the current time in standard date format.

```
$ dlstat show-phys -Td net0
Sun Jun  1 12:54:47 PDT 2014
          LINK    IPKTS   RBYTES    OPKTS   OBYTES
          net0      184   27.14K      165   22.91K
```

EXAMPLE 93 Displaying Input and Output Packet Drops

The following example displays the input and output packet drop statistics for the datalink `net0`.

```
$ dlstat show-phys net0 -o idrops,idropbytes,odrops,odropbytes
  IDROPS IDROPBYTES   ODROPS ODROPBYTES
   399     42.52K        0       0
```

Displaying Network Traffic Statistics of Datalinks

You can use the `dlstat show-link` command to display the network traffic statistics for a datalink.

EXAMPLE 94 Displaying Network Traffic Statistics for a Datalink

This example shows the network traffic statistics for the datalink `vnic0`.

```
$ dlstat show-link vnic0
LINK    IPKTS   RBYTES    OPKTS   OBYTES
vnic0   3       180       0       0
```

EXAMPLE 95 Displaying Network Traffic Statistics for a Datalink With Dedicated Hardware Rings

This example shows the receive-side network traffic statistics for the datalink `vnic0` that has four dedicated Rx rings. The `hw` value under the `ID` column in the output indicates that the datalink `vnic0` has dedicated hardware rings.

```
$ dlstat show-link -r vnic0
LINK   TYPE      ID  INDEX    IPKTS   RBYTES    INTRS    POLLS   IDROPS
vnic0    rx   local     --        0        0        0        0        0
vnic0    rx   other     --       64    2.94K        0        0        0
vnic0    rx      hw      8        0        0        0        0        0
vnic0    rx      hw      9       53    7.97K       53        0        0
vnic0    rx      hw     10        4      392        4        0        0
vnic0    rx      hw     11  153.65K  220.68M  153.65K        0        0
```

EXAMPLE 96 Displaying Transmit-Side Network Traffic Statistics for a Datalink

This example shows the transmit-side network traffic statistics for the datalink `vnic0`.

```
$ dlstat show-link -t vnic0
LINK   TYPE      ID  INDEX    OPKTS   OBYTES   ODROPS
vnic0    tx   local     --        0        0        0
vnic0    tx   other     --       19      798        0
vnic0    tx      sw     --        0        0        0
```

EXAMPLE 97 Displaying Network Traffic Statistics for a Datalink Without Dedicated Hardware Rings

This example shows the network traffic statistics for the datalink `net6` that does not have dedicated Rx rings. The `sw` value under the `ID` column in the output indicates that the datalink `net6` is not configured with dedicated hardware rings.

```
$ dlstat show-link -r net6
LINK  TYPE      ID  INDEX    IPKTS   RBYTES INTRS POLLS   IDROPS
net6    rx   local     --        0   0        0     0        0
net6    rx   other     --        0   0        0     0        0
net6    rx      sw     --        0   0        0     0        0
```

Displaying Network Traffic Statistics of Link Aggregations

The `dlstat show-aggr` command shows network packet statistics for each aggregation's ports when traffic traverses the aggregation on the system.

EXAMPLE 98 Displaying Network Traffic Statistics for Link Aggregations

```
$ dlstat show-aggr
LINK      PORT    IPKTS   RBYTES    OPKTS   OBYTES
aggr0     --         13      832       13      780
aggr0     net0        0        0       13      780
aggr0     net3       13      832        0        0
```

In this example, the output indicates the configuration of a link aggregation `aggr0` with two underlying links, `net0` and `net3`. As network traffic is received or sent by the system through the aggregation, information about incoming and outgoing packets and their respective sizes is reported for every port. The ports are identified by the underlying links of the aggregation.

For information about link aggregations, see Chapter 2, “Configuring High Availability by Using Link Aggregations” in *Managing Network Datalinks in Oracle Solaris 11.4*.

EXAMPLE 99 Displaying Per-Ring Statistics for an Aggregation

When you specify `-r` or `-t` option with the `dlstat show-aggr` command, per-ring statistics of the aggregation are displayed.

```
$ dlstat show-aggr -r
```

```
LINK      PORT  INDEX      IPKTS   RBYTES
aggr1      --    --        7.93M    4.37G
aggr1     net4   0       541.37K  298.21M
aggr1     net4   1             0        0
aggr1     net4   2             0        0
aggr1     net4   3         1.79M   91.63M
aggr1     net4   4       780.47K  433.30M
aggr1     net4   5             1       98
aggr1     net4   6       530.05K  292.34M
aggr1     net4   7       548.36K  301.74M

$ dlstat show-aggr -t
LINK     PORT  INDEX      OPKTS    OBYTES
aggr1      --     --      2.23M   146.96M
aggr1    net4      0    221.94K    14.65M
aggr1    net4      1    147.37K     9.73M
aggr1    net4      2          0         0
aggr1    net4      3     83.04K     5.49M
aggr1    net4      4    429.10K    28.32M

$ dlstat show-aggr -rt
LINK    PORT  TYPE  INDEX    IPKTS   RBYTES    OPKTS   OBYTES
aggr1     --    --     --    7.93M    4.37G    2.23M  146.96M
aggr1   net4    tx      0       --       --  221.94K   14.65M
aggr1   net4    tx      1       --       --  147.37K    9.73M
aggr1   net4    tx      2       --       --        0        0
aggr1   net4    tx      3       --       --   83.04K    5.49M
aggr1   net4    tx      4       --       --  429.10K   28.32M
aggr1   net4    tx      5       --       --  185.15K   12.22M
aggr1   net4    tx      6       --       --   71.54K    4.72M
aggr1   net4    tx      7       --       --   67.62K    4.46M
aggr1   net4    rx      0  541.37K  298.21M     --         --
aggr1   net4    rx      1        0        0     --         --
aggr1   net4    rx      2        0        0     --         --
aggr1   net4    rx      3    1.79M  991.63M     --         --
aggr1   net4    rx      4  780.47K  433.30M     --         --
aggr1   net4    rx      5        1       98     --         --
aggr1   net4    rx      6  530.05K  292.34M     --         --
aggr1   net4    rx      7  548.36K  301.74M     --         --
```

EXAMPLE 100 Displaying Input and Output Packet Drops for an Aggregation

You can use the `dlstat show-aggr` command to display the input and output packet drops per aggregation. You can use the following options with the command to display the number of bytes of the drops in addition to the number of input and output packet drops.

```
# dlstat show-aggr [-o
          [idrops[,idropbytes][,odrops][,odropbytes]]
```

The following example displays the input and output packet drop statistics for `aggr1`.

```
$ dlstat show-aggr aggr1 -o idrops,idropbytes,odrops,odropbytes -u R
  IDROPS IDROPBYTES   ODROPS ODROPBYTES
   125     22.63K        0       0
```

Displaying Network Traffic Statistics of Bridges

The `dlstat show-bridge` command shows network statistics for each bridge and lists the statistics of the links connected to each bridge.

EXAMPLE 101 Displaying Network Traffic Statistics for Bridges

In this example, the network statistics for the bridges `rbblue0` and `stbred0` are displayed.

```
$ dlstat show-bridge
BRIDGE        LINK     IPKTS    RBYTES     OPKTS    OBYTES     DROPS  FORWARDS
rbblue0         --     1.93K   587.29K     2.47K     3.30M         0         0
             simblue1     72     4.32K     2.12K     2.83M         0        --
             simblue2  1.86K   582.97K       348   474.04K         0        --
stbred0         --       975   976.69K     3.44K     1.13M         0        38
              simred3    347   472.54K     1.86K   583.03K         0        --
              simred4    628   504.15K     1.58K   551.51K         0        --
```

Displaying Network Traffic Statistics of Flows

Statistics on flows help you to evaluate packet traffic on all the defined flows on a system. To display the statistics on flows, use the `flowstat` command. For more information, see the `flowstat(8)` man page.

Use the following command syntax to display network traffic statistics on flows:

```
$ flowstat [-r|-t] [-l link] \
   [-Tu | -Td] [flow] [interval [count]]
```

`-r`
: Displays receive-side network traffic statistics only. You should not specify the `-t` option with this option.

  If you do not specify the `-r` option or `-t` option, both the transmit-side and receive-side network statistics are displayed.

`-t`	Displays transmit-side network traffic statistics only. You should not the specify the `-r` option with this option. If you do not specify the `-r` option or the `-t` option, both the transmit-side and receive-side network statistics are displayed.
`-l` *link*	Name of the datalink whose network statistics you want to monitor. If you do not specify the datalink, then the information about all the configured flows on the system are displayed.
`-Tu`	Displays the current time in internal representation.
`-Td`	Displays the current time in standard date format.
flow	Name of the flow whose network statistics you want to monitor. If you do not specify the flow, then depending on the specified link, all the flow statistics are displayed.
interval	Specifies the time in seconds at which you want to refresh the network statistics. If you do not specify the interval value, then the total number of packets and bytes is displayed.
count	Specifies the number of times you want the displayed network traffic statistics to be refreshed. If you do not specify the count value, the statistics are refreshed indefinitely.

The following examples show different ways to display information about configured flows on the system.

EXAMPLE 102 Displaying Network Traffic Statistics for Flows

In this example, network traffic statistics for all the configured flows on the system are displayed with an interval value of `1` second and the count value of `2`.

```
$ flowstat 1 2
FLOW     IPKTS   RBYTES   IDROPS    OPKTS   OBYTES   ODROPS
flow1    1.78M    2.68G      443  889.57K   58.72M        0
flow2        0        0        0        0        0        0
flow1    8.31K   12.51M      243    4.22K  280.45K        0
flow2        0        0        0        0        0        0
```

Consider the flows, `flow1` and `flow2`, as a set. The first set of flows, `flow1` and `flow2`, show the total number of network traffic statistics received and transmitted by the flows. In this example, `1.78M` is the total number of packets received by `flow1`. The second set of flows, `flow1` and

flow2, show the network statistics in rates per second, also known as the normalized value. In this example, 8.31K is the normalized value of the packets received by flow1 in the interval of 1 second.

EXAMPLE 103 Displaying Transmit-Side Traffic Statistics for Flows

In this example, the network traffic statistics about outgoing traffic for all the configured flows on the system are displayed.

```
$ flowstat -t
FLOW     OPKTS   OBYTES   ODROPS
flow1   24.37M    1.61G        0
flow2        0        0        0
```

EXAMPLE 104 Displaying Receive-Side Traffic Statistics for Flows on a Datalink

In this example, incoming network traffic for all the configured flows on the datalink net0 are displayed with an interval value of 2 seconds and the count value of 5.

```
$ flowstat -r -l net0 2 5
FLOW     IPKTS   RBYTES   IDROPS
flow1    2.38M    3.59G   14.89K
flow2        0        0        0
flow1    8.24K   12.40M      180
flow2        0        0        0
flow1    8.94K   13.47M      206
flow2        0        0        0
flow1    7.43K   11.19M      161
flow2        0        0        0
flow1    8.38K   12.62M      213
flow2        0        0        0
```

Consider the flows, flow1 and flow2, as a set. The first set of flows, flow1 and flow2, show the total number of packets and bytes received by the flows. In this example, 2.38M is the total number of packets received and 3.59G is the total number of bytes received by flow1. The second set of flows, flow1 and flow2, show the network statistics in rates per second, also known as the normalized value. In this example, 8.24K is the normalized value of the packets received by flow1 in the interval of 2 seconds. Similarly, the succeeding sets of flows also show the normalized value for the network traffics statistics in the periodic interval of 2 seconds.

EXAMPLE 105 Displaying Traffic Statistics for Flows With Time

The following example displays statistics about incoming traffic on all the flows that are created over the datalink net0 with the internal representation of the current time.

```
$ flowstat -r -l net0 -Tu
1364380279
          FLOW        IPKTS   RBYTES    IDROPS
          tcp-flow  183.11K  270.24M        0
          udp-flow        0        0        0
```

The following example displays statistics about incoming traffic on all the flows that are created over the datalink `net0` with the current time in standard date format.

```
$ flowstat -r -l net0 -Td
Wednesday, March 27, 2013 04:01:011 PM IST
          FLOW        IPKTS   RBYTES    IDROPS
          tcp-flow  183.11K  270.24M        0
          udp-flow        0        0        0
```

Configuring Network Accounting for Network Traffic

You can use the extended accounting facility to set up network accounting on the system. Network accounting involves capturing statistics about network traffic in a log file. You can maintain records of traffic for tracking, provisioning, consolidation, and billing purposes. Later, you can see the log file to obtain historical information about network use over a period of time.

To set up network accounting, use the extended accounting facility's `acctadm` command. For more information, see the `acctadm`(8) man page. After you have completed setting up network accounting, use the `flowstat` command to record traffic statistics.

▼ How to Set Up Network Accounting

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See “Using Rights Profiles to Perform Network Configuration” on page 34.

1. **View the status of the accounting types that can be enabled by the extended accounting facility.**

   ```
   $ acctadm [process | task | flow | net]
   ```

   The extended accounting facility can enable four types of accounting. The optional operands of the `acctadm` command correspond to the following accounting types:

   - `process` – Process accounting
   - `task` – Task accounting

- `flow` – Flow accounting
- `net` – Network accounting

Note - Network accounting also applies to flows that are managed by the `flowadm` and `flowstat` commands as discussed in "Managing Network Resources by Using Flows" on page 191. Therefore, to set up accounting for these flows, use the `net` option with the `acctadm` command. Do *not* use the `flow` option, which enables flow accounting for IPQoS configurations.

Specifying `net` displays the status of network accounting. If `net` is not used, then the status of all four accounting types is displayed.

---

2. **Enable the extended accounting for network traffic.**

```
$ acctadm -e extended -f filename net
```

where *filename* includes the full path of the log file that captures network traffic statistics. The log file can be created in any directory that you specify.

3. **Verify that extended network accounting has been activated.**

```
$ acctadm net
```

Example 106 Setting Up Network Accounting on the System

This example shows how to configure network accounting to capture and display historical traffic information on the system.

View the status of all accounting types as follows:

```
$ acctadm
           Task accounting: inactive
       Task accounting file: none
     Tracked task resources: none
   Untracked task resources: extended
         Process accounting: inactive
    Process accounting file: none
  Tracked process resources: none
Untracked process resources: extended,host
           Flow accounting: inactive
       Flow accounting file: none
     Tracked flow resources: none
   Untracked flow resources: extended
            Net accounting: inactive
       Network accounting file: none
```

```
     Tracked Network resources: none
   Untracked Network resources: extended
```

The output shows that network accounting is not active. Therefore, you should enable extended network accounting.

```
$ acctadm -e extended -f /var/log/net.log net
$ acctadm net
            Net accounting: active
       Net accounting file: /var/log/net.log
     Tracked net resources: extended
   Untracked net resources: none
```

Displaying Historical Statistics on Network Traffic

After you have enabled network accounting, you can use the `dlstat` and `flowstat` commands to extract information from the log file.

You must enable extended accounting for the network before you can display historical data about the network. Further, to display historical data about traffic on flows, you must first configure flows on the system, as explained in “Managing Network Resources by Using Flows” on page 191.

Displaying Historical Network Traffic Statistics on Datalinks

You can display historical network traffic statistics on datalinks by using the following command syntax:

```
$ dlstat show-link -h [-a] -f filename [-d date] \
   [-F format] [-s start-time] \
   [-e end-time] [link]
```

`-h` Displays a summary of historical information about resource usage by incoming and outgoing packets on datalinks.

`-a` Displays resource usage on all datalinks, including those that have already been deleted after the data capture.

`-f` *filename* Specifies the log file that was defined when network accounting was enabled with the `acctadm` command.

`-d` *date* — Displays logged information for the specified date.

`-F` *format* — Displays the data in a specific format that can then be plotted for analysis. Currently, `gnuplot` is the only supported format.

`-s` *start-time* — Specifies the start time to display the logged information of the network statistics. Use the `MM/DD/YYY,hh:mm:ss` format. The `hour` (`hh`) must use 24-hour clock notation. If you do not include the date, then data for the specified time range for the current date is displayed.

`-e` *end-time* — Specifies the end time to display the logged information of the network statistics. Use the `MM/DD/YYY,hh:mm:ss` format. The `hour` (`hh`) must use 24-hour clock notation. If you do not include the date, then data for the specified time range for the current date is displayed.

link — Displays historical data for a specified datalink. If you do not use this option, then historical network data for all configured datalinks is displayed.

EXAMPLE 107 Displaying Historical Statistics About Resource Usage on Datalinks

In this example, the historical statistics about network traffic and its use of resources on all the datalinks in a system are displayed.

```
$ dlstat show-link -h -f /var/log/net.log
LINK  DURATION  IPKTS    RBYTES     OPKTS  OBYTES     BANDWIDTH
net0   80        1031    546908      0        0       2.44 Mbps
net1   100       2045    235977      0        0       9.67 Mbps
```

Displaying Historical Network Traffic Statistics on Flows

You can display historical network traffic statistics on flows by using the following command syntax:

```
$ flowstat -h [-a] -f filename [-d date] \
   [-F format] [-s start-time] \
   [-e end-time] [flow]
```

`-h` — Displays a summary of historical information about resource usage by incoming and outgoing packets on configured flows.

`-a` — Displays resource usage on all configured flows, including those that have already been deleted after the data capture.

`-f` *filename*	Specifies the log file that was defined when network accounting was enabled with the `acctadm` command.
`-d`	Displays logged information for the specified date.
`-F` *format*	Displays the data in a specific format. Currently, `gnuplot` is the only supported format.
`-s` *start-time*	Specifies the start time to display the logged information of the network statistics. Use the `MM/DD/YYY,hh:mm:ss` format. The `hour` (`hh`) must use 24-hour clock notation. If you do not include the date, then data for the specified time range for the current date is displayed.
`-e` *end-time*	Specifies the end time to display the logged information of the network statistics. Use the `MM/DD/YYY,hh:mm:ss` format. The `hour` (`hh`) must use 24-hour clock notation. If you do not include the date, then data for the specified time range for the current date is displayed.
flow	Displays historical data for a specified flow. If you do not use this option, then historical network data for all configured flows is displayed.

EXAMPLE 108 Displaying Historical Statistics About Resource Usage on Flows

The following example displays historical statistics of resource usage by traffic on the flows in a system.

```
$ flowstat -h -f /var/log/net.log
FLOW      DURATION  IPACKETS RBYTES      OPACKETS OBYTES    BANDWIDTH
flowtcp   100       1031     546908      0        0         43.76Kbps
flowudp   0         0        0           0        0          0.00Mbps
```

The following example displays historical statistics of resource usage by traffic on `flowtcp` over a given date and time range.

```
$ flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-f /var/log/net.log flowtcp

FLOW      START       END         RBYTES   OBYTES     BANDWIDTH
flowtcp   10:39:06    10:39:26    1546     6539         3.23 Kbps
flowtcp   10:39:26    10:39:46    3586     9922         5.40 Kbps
flowtcp   10:39:46    10:40:06    240      216        182.40 bps
flowtcp   10:40:06    10:40:26    0        0            0.00 bps
```

The following example displays historical statistics of resource usage by traffic on `flowtcp` over a given date and time range by using `gnuplot` format.

```
$ flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-F gnuplot -f /var/log/net.log flowtcp
# Time tcp-flow
10:39:06 3.23
10:39:26 5.40
10:39:46 0.18
10:40:06 0.00
```

Index

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

Z