Managing Network Datalinks in Oracle® Solaris 11.4

Part No: E60990
November 2020

ORACLE

Managing Network Datalinks in Oracle Solaris 11.4

Part No: E60990

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: E60990

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

Using This Documentation 13

1 Introduction to Managing Network Datalinks 15
Datalink Technologies for Improving the Network 15
Using Rights Profiles to Perform Network Configuration 16

2 Configuring High Availability by Using Link Aggregations 19
Overview of Link Aggregations 19
Deploying Link Aggregation Configurations 20
Trunk Aggregations 21
Using a Switch 21
Back-to-Back Trunk Aggregation Configuration 22
Using a Switch With the Link Aggregation Control Protocol 23
Defining Aggregation Policies for Load Balancing 24
Datalink Multipathing Aggregations 24
How DLMP Aggregation Works 25
Failure Detection in DLMP Aggregation 26
DLMP Aggregation of SR-IOV NICs 28
DLMP Aggregation of InfiniBand Host Channel Adapter (HCA) Ports 29
Configuring IPMP Over DLMP in a Virtual Environment 29
Requirements for Link Aggregations 31

3 Link Aggregation Configuration Tasks 33
Creating a Link Aggregation 33
▼ How to Create a Link Aggregation 34
▼ How to Configure DLMP Aggregation of SR-IOV NICs 36
▼ How to Configure DLMP Aggregation of InfiniBand HCA Ports 38
Displaying the Clients Associated With Each Port 39

Adding a Link to an Aggregation 40
▼ How to Add a Link to an Aggregation 40
Removing a Link From an Aggregation 41
Modifying a Trunk Aggregation Attributes 41
Configuring Probe-Based Failure Detection for DLMP Aggregation 42
▼ How to Configure Probe-Based Failure Detection 42
Monitoring Probe-Based Failure Detection 45
Deleting a Link Aggregation 47
▼ How to Delete a Link Aggregation 47
Use Case: Configuring a Link Aggregation 48

4 Configuring Virtual Networks by Using Virtual Local Area Networks 51
Deploying VLANs 51
When to Use VLANs 52
About VLAN IDs 52
VLAN Topology 52
Planning a VLAN Configuration 54
Configuring a VLAN 55
▼ How to Configure a VLAN 55
Using VLANs With Virtualization 57
Configuring VLANs Over a Link Aggregation 61
▼ How to Configure VLANs Over a Link Aggregation 61
Displaying VLAN Information 63
Modifying VLANs 64
Modifying the VLAN ID of a VLAN 64
Migrating a VLAN to Another Underlying Link 65
Deleting a VLAN 67
Using VLAN Tagging for SR-IOV-Enabled Ports 68
Use Cases in of VLAN Deployments 68
Combining Link Aggregation and VLAN Configurations 68
Configuring Flows for a VLAN 70
Configuring Bandwidth for VLAN VNICs 71

5 Configuring Private Virtual Local Area Networks 73
Overview of Private VLANs 73
PVLAN Ports 76
Tagging the Outgoing Traffic 76

PVLAN Configuration Requirements 77
PVLANs With Zones 77
Configuring a Private VLAN 79
Modifying Private VLANs 79
Deleting a Private VLAN 80
Assigning a PVLAN to a Zone 80
▼ How to Create a PVLAN and Assign to a Zone 81

6 Administering Bridging Features 83
Overview of Bridged Networks 83
Simple Bridged Network 84
How Oracle Solaris Bridges Are Implemented in the Network Stack 85
Bridged Network Ring 86
How a Bridged Network Works 87
Bridging Protocols 88
STP Daemon 89
TRILL Daemon 90
Creating a Bridge 90
Modifying the Protection Type for a Bridge 92
Adding Links to an Existing Bridge 92
Removing Links From a Bridge 93
Setting Link Properties for a Bridge 93
Displaying Bridge Configuration Information 94
Displaying Information About Configured Bridges 94
Displaying Configuration Information About Bridge Links 96
Deleting a Bridge From the System 96
▼ How to Delete a Bridge From the System 97
Administering VLANs on Bridged Networks 97
▼ How to Configure VLANs Over a Datalink That Is Part of a Bridge 97
VLANs and the STP and TRILL Protocols 98
Debugging Bridges 99

7 Exchanging Network Connectivity Information With Link Layer Discovery Protocol 101
Overview of LLDP 101
Components of an LLDP Implementation 102
Information Sources of the LLDP Agent 103

LLDP Agent Modes 103
Information the LLDP Agent Advertises 104
Mandatory TLV Units 104
Optional TLV Units 105
TLV Unit Properties 105
Enabling LLDP on the System 107
▼ How to Install the LLDP Package 108
▼ How to Enable LLDP Globally 108
▼ How to Enable LLDP for Specific Ports 109
Specifying TLV Units and Values for the LLDP Packet of an Agent 111
▼ How to Specify TLV Units for the LLDP Packet of an Agent 112
▼ How to Define TLV Units 113
Disabling LLDP 115
▼ How to Disable LLDP 115
Monitoring LLDP Agents 116
Displaying the Advertised Information 116
Displaying LLDP Statistics 119

8 Managing Converged Networks by Using Data Center Bridging 121
Overview of Data Center Bridging 121
Considerations When Using DCB 123
Priority-Based Flow Control 123
Enhanced Transmission Selection 124
Enabling DCBX 125
▼ How to Enable the Data Center Bridging Exchange Feature Manually 125
Setting the Mode of Operation for DCB 126
▼ How to Set the Mode of Operation for DCB 127
Customizing Priority-Based Flow Control for DCB 128
Setting the PFC-Related Datalink Properties 128
Setting the PFC TLV Units 129
Displaying PFC Configuration Information 130
Displaying Datalink Properties 130
Displaying the Capability of the Local Host to Synchronize PFC Information 131
Displaying PFC Mapping Information Between Host and Peer 132
Displaying Priority Definitions 132
Application Priority Configurations 133

Customizing Enhanced Transmission Selection for DCB 134
Setting the ETS-Related Datalink Properties 134
Setting ETS TLV Units 136
Recommending ETS Configuration to the Peer 136
Displaying ETS Configuration Information 138

A Comparison: Trunk Aggregations and DLMP 143

B Link Aggregations and IPMP: Feature Comparison 145

C Packet Format of Transitive Probes 147

Index 149

Using This Documentation

- **Overview** – Describes how to manage network datalinks to improve network performance. Specifically, describes how to combine links into aggregations by using trunk or DLMP aggregation, divide your network into subnetworks by using virtual local area networks, connect separate network segments by using bridges, exchange network connectivity information by using Link Layer Discovery Protocol, and manage converged networks by using data center bridging.
- **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 Managing Network Datalinks

This chapter introduces the advanced features in datalink management. The application of these features depends on specific circumstances as well as hardware configuration requirements. Therefore, not all the configuration procedures in this chapter have to be completed. Instead, select and deploy the technologies that address your site network setup.

Make sure you are familiar with basic network configuration as well as datalink fundamentals as described in *Configuring and Managing Network Components in Oracle Solaris 11.4*.

This chapter contains the following topics:

- "Datalink Technologies for Improving the Network"
- "Using Rights Profiles to Perform Network Configuration"

Datalink Technologies for Improving the Network

Different networking technologies can increase efficiency of your network setup. These technologies address different areas of network operations. However, all of them provide common benefits such as network availability, improved network performance, and ease of network administration. These layer 2 technologies are discussed in detail in subsequent chapters of this guide:

- Link aggregations
- Virtual Local Area Networks (VLANs)
- Private VLANs or PVLANs
- Bridged networks
- Link Layer Discovery Protocol (LLDP)
- Data Center Bridging (DCB)

To implement network security, see *Securing the Network in Oracle Solaris 11.4*.

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.

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

Configuring High Availability by Using Link Aggregations

This chapter provides an overview of link aggregations and covers the following topics:

- "Overview of Link Aggregations"
- "Deploying Link Aggregation Configurations"
- "Trunk Aggregations"
- "Datalink Multipathing Aggregations"
- "Configuring IPMP Over DLMP in a Virtual Environment"
- "Requirements for Link Aggregations"

Overview of Link Aggregations

Link aggregations enable you to pool multiple datalink resources to administer them as a single unit. The combined resources improve bandwidth, provide high network availability, and eases management.

When used with network virtualization, link aggregation has advantages over other high availability technology such as IPMP. For example, while IPMP needs to be configured in every non-global zone, link aggregation is configured in the global zone. The aggregation becomes the underlying link for the non-global zone. When the zone boots, it is assigned a virtual network interface card (VNIC) on the aggregation. Thus the one-time configuration automatically provides highly availability to non-global zones as well.

Because link aggregation groups multiple links into a single logical datalink, the aggregation also benefits from other datalink features, such as link protection and resource management. See Chapter 1, "Using Link Protection in Virtualized Environments" in *Securing the Network in Oracle Solaris 11.4* and Chapter 7, "Managing Network Resources" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

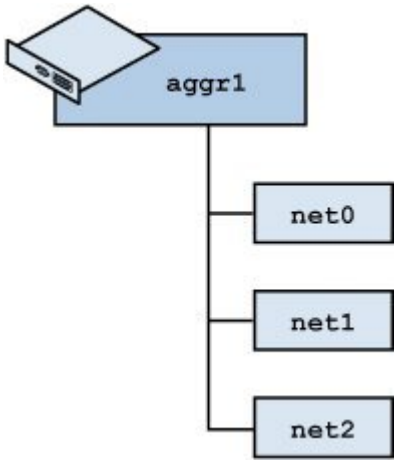
In Oracle VM server for SPARC, you can configure a virtual switch to use a link aggregation as a network device to connect to the physical network. This configuration enables the

virtual switch to avail of aggregation features set by the IEEE 802.3ad Link Aggregation standard. See "Using Link Aggregation With a Virtual Switch" in *Oracle VM Server for SPARC 3.6 Administration Guide*. The Oracle Enterprise Manager Ops Center can be the interface to manage the aggregations. See `http://www.oracle.com/pls/topic/lookup?ctx=oc122&id=OPCCM`.

Deploying Link Aggregation Configurations

The following figure shows an example of a simple link aggregation.

FIGURE 1 Link Aggregation Configuration



The aggregation `aggr1` consists of three underlying datalinks to serve network traffic. The underlying links are hidden from external applications. Instead, only the logical datalink `aggr1` is accessible.

Oracle Solaris supports the following types of link aggregations:

- Trunk aggregations
- Datalink multipathing (DLMP) aggregations

For a comparison, see Appendix A, "Comparison: Trunk Aggregations and DLMP".

Trunk Aggregations

Based on the IEEE 802.3ad standard, trunk aggregations work by enabling multiple flows of traffic to be spread across a set of aggregated ports. The clients configured over the aggregations acquire a consolidated bandwidth of the underlying links. Each network port is associated with every underlying datalink.

You might use this type in the following situations:

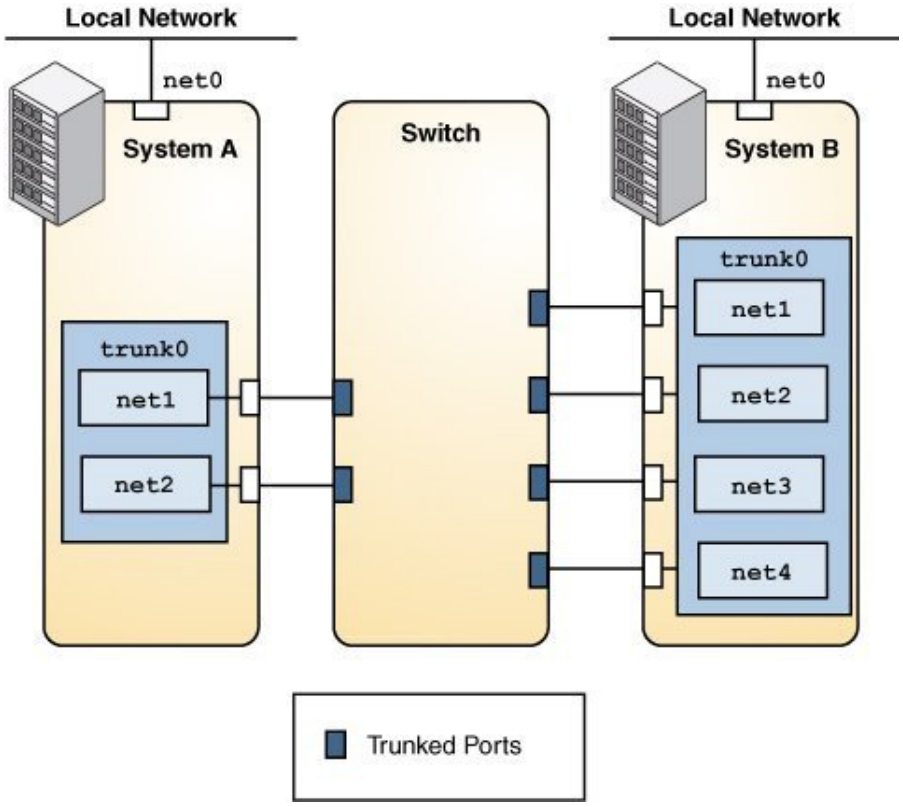
- For systems in the network that run applications with distributed heavy traffic, you can dedicate a trunk aggregation to that application's traffic to take advantage of the increased bandwidth.
- For sites with limited IP address space that require large amounts of bandwidth, you need only one IP address for the trunk aggregation of datalinks.
- For sites that need to hide any internal datalinks, the IP address of the trunk aggregation hides these datalinks from external applications.
- For applications that need reliable network connection, trunk aggregation protects network connections against link failure.

Unless you specify otherwise, an aggregation is automatically created as a trunk aggregation.

Using a Switch

Systems with trunk aggregations might use an external switch to connect to other systems. See the following figure.

FIGURE 2 Trunk Aggregation Using a Switch



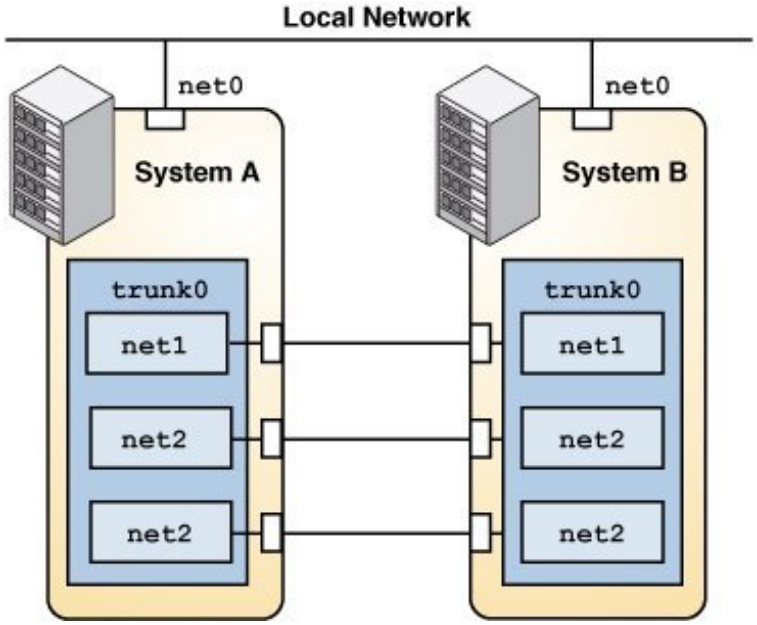
System A's aggregation consists of two datalinks, while System B's aggregation consists of four. All the links of both aggregations are connected to the aggregated ports on the switch.

If aggregations and switches are used together, the switch must support the IEEE 802.3ad standard and the switch ports must be configured for aggregation.

Back-to-Back Trunk Aggregation Configuration

Trunk aggregations can be set up as a back-to-back configuration. Two systems are directly connected to run parallel aggregations, as shown in the following figure.

FIGURE 3 Back-to-Back Trunk Aggregation Configuration



The back-to-back setup enables both systems to provide redundancy, high availability, and high-speed communication between both systems.

Back-to-back aggregations are commonly used in mirrored database server setups in large deployments such as data centers. Both servers must be updated together and therefore require significant bandwidth, high-speed traffic flow, and reliability.

Using a Switch With the Link Aggregation Control Protocol

If the switch supports Link Aggregation Control Protocol (LACP), such as Oracle Switch ES1-24 (see "Sun Ethernet Fabric Operating System" in *LA Administration Guide*), you can enable LACP for both the switch and the system.

LACP is more reliable in detecting datalink failures. Without LACP, to detect failures, an aggregation relies only on the link state reported by the device driver. With LACP, LACP data units (LACPDUs) are exchanged at regular intervals to ensure that the aggregated datalinks can send and receive traffic. LACP also detects some misconfiguration cases, for example, when the grouping of datalinks does not match between the two peers.

LACP can be set to one of three modes:

- `off` – Default mode for aggregations. The system does not generate LACPDUs.
- `active` – System generates LACPDUs at specified intervals.
- `passive` – System generates an LACPDU only when it receives an LACPDU from the switch. When both the aggregation and the switch are configured in `passive` mode, they do not exchange LACPDUs.

Defining Aggregation Policies for Load Balancing

To distribute load across the available aggregation links and establish load balancing, select from the following load specifiers:

- L2 – Determines the outgoing link by using the MAC (L2) header of each packet
- L3 – Determines the outgoing link by using the IP (L3) header of each packet
- L4 – Determines the outgoing link by using the TCP, User Datagram Protocol (UDP), or other Upper Layer Protocol (ULP) (L4) header of each packet

Any combination of these policies is also valid. The default policy is L4.

Datalink Multipathing Aggregations

Datalink multipathing (DLMP) aggregation is a type of link aggregation that does not require switch configuration. It also supports link-based failure detection and probe-based failure detection to ensure continuous availability of the network to send and receive traffic. DLMP aggregations can be created from the following components:

- Physical NICs
- VNICs
- Single root I/O virtualization (SR-IOV) VNICs on the aggregation of SR-IOV NICs
- IP over InfiniBand (IPoIB) VNICs on the aggregation of IPoIB partition datalinks

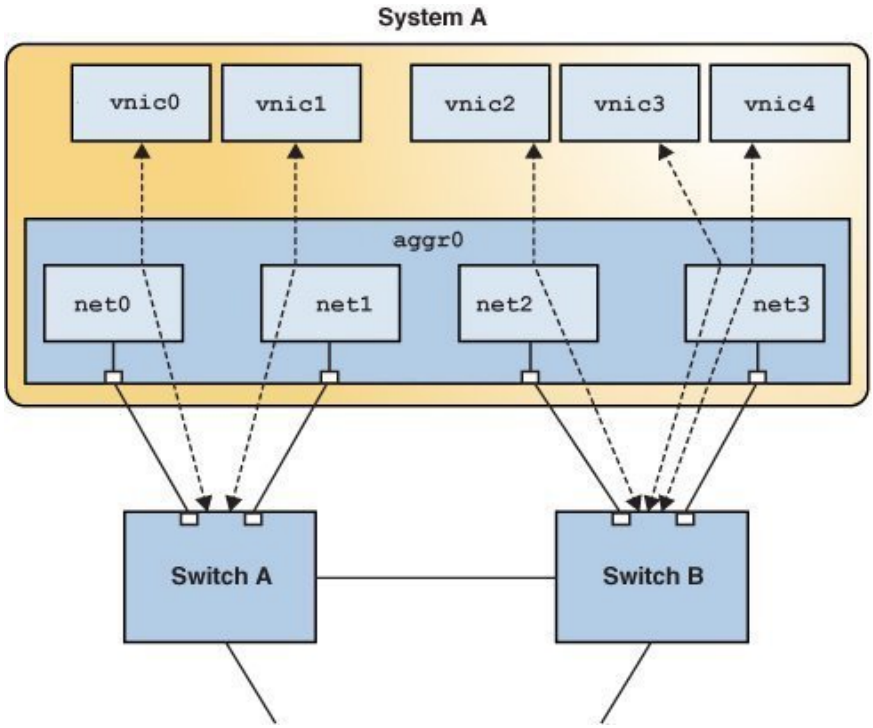
DLMP aggregations span multiple switches. As a Layer 2 technology, aggregations integrate well with other Oracle Solaris virtualization technologies.

How DLMP Aggregation Works

In a trunk aggregation, each port is associated with every configured datalink over the aggregation. In a DLMP aggregation, a port is associated with any of the aggregation's configured datalinks.

The following figure shows how a DLMP aggregation works.

FIGURE 4 DLMP Aggregation



System A's link aggregation `aggr0` consists of four underlying links. The aggregation is connected to both Switch A and Switch B through which the system accesses the wider network.

VNICs are associated with aggregated ports through the underlying links. Thus, `vnic0` through `vnic3` are associated with the aggregated ports through the underlying links `net0` through `net3`.

If an equal number of VNICs and underlying links exists, then each port has a corresponding underlying link. If more VNICs exist than underlying links, then one port is associated with multiple datalinks. Thus, in the figure, `vnic4` shares a port with `vnic3`.

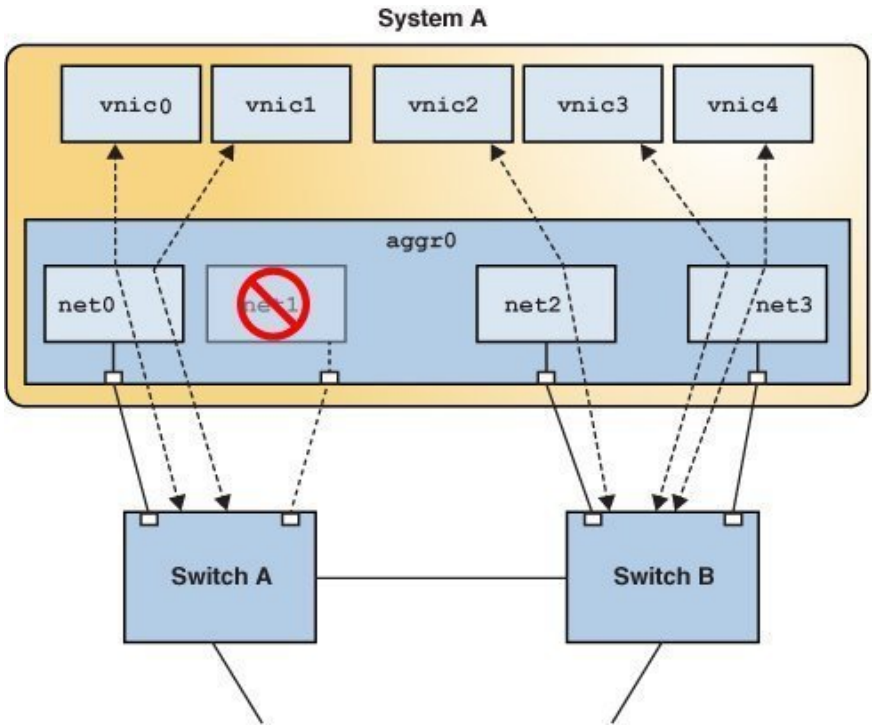
Failure Detection in DLMP Aggregation

Failure detection in DLMP aggregation detects failure in aggregated ports.

When a port fails, the clients associated with that port are failed over to an active port to maintain network connectivity. Failed aggregated ports remain unusable until they are repaired. The remaining active ports continue to function while any existing ports are deployed as needed. After the failed port recovers from the failure, clients from other active ports can be associated with it.

To understand the failover mechanism, compare the following figure with Figure 4, “DLMP Aggregation,” on page 25.

FIGURE 5 DLMP Aggregation When a Port Fails



The link `net1` has failed and its connection with the switch is down. Thus, `net0` becomes a shared port between `vnic0` and `vnic1`.

DLMP aggregation supports both link-based and probe-based failure detection.

Link-Based Failure Detection

Link-based failure detection detects failure arising from loss of direct connection between the datalink and the first-hop switch. Link-based failure detection is enabled by default when a DLMP aggregation is created.

Probe-Based Failure Detection

Probe-based failure detection detects failures between an end system and the configured targets. This method is useful when a default router is down or when the network becomes unreachable.

Probe-based failure detection works by sending and receiving probe packets. It uses two types of probes together to determine the health of the aggregated physical datalinks: Internet Control Message Protocol (ICMP (L3)) probes and transitive (L2) probes.

- **ICMP Probing**

  ICMP packets are sent from a source address to a target address. You can specify the source and target IP addresses manually or let the system choose them. If a source address is not specified, it is chosen from the IP addresses that are configured on the aggregation or on a VNIC on the aggregation. If a target address is not specified, it is chosen from one of the next hop routers on the same subnet as a selected source IP addresses.
- **Transitive Probing**

  L2 packets are sent from ports with no configured source addresses to targets with configured source addresses. Therefore, the source cannot receive ICMP probe replies. If the source receives an L2 reply from the target, then the ICMP target is reachable.

  Transitive probing is performed when the health state of all network ports cannot be determined by using only ICMP probing.

  Oracle Solaris includes proprietary protocol packets for transitive probes that are transmitted over the network. For more information, see Appendix C, “Packet Format of Transitive Probes”.

DLMP Aggregation of SR-IOV NICs

By grouping several SR-IOV-enabled NICs, you can create a SR-IOV-enabled DLMP aggregation on top of which you configure SR-IOV VNICs. The VNICs can be associated with the virtual functions (VFs) of the ports. Ports can be dynamically added or removed ports without disrupting the network connections or removing the existing configurations. See “How to Configure DLMP Aggregation of SR-IOV NICs” on page 36.

For information about SR-IOV VNICs, see “Using Single Root I/O Virtualization With VNICs” in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

DLMP Aggregation of InfiniBand Host Channel Adapter (HCA) Ports

By grouping multiple InfiniBand host channel adapter (HCA) ports, you can create a DLMP aggregation over which you configure IPoIB VNICs over this DLMP aggregation.

Note - Only the IPoIB ULP through IPoIB VNIC configured over DLMP aggregation of InfiniBand HCA ports can be highly available and not the other InfiniBand ULPs.

You cannot directly configure an IP address on the DLMP aggregation of InfiniBand HCA ports. Instead, you configure IP addresses on the IPoIB VNICs configured over the DLMP aggregation.

When a link status of an IPoIB datalink is changed to being `up` or `down`, individual HCA ports are automatically failed over and network high availability of IPoIB partition datalinks is achieved.

Note - For DLMP aggregations of InfiniBand HCA ports, only link-based failover is supported and probe-based failover is not supported.

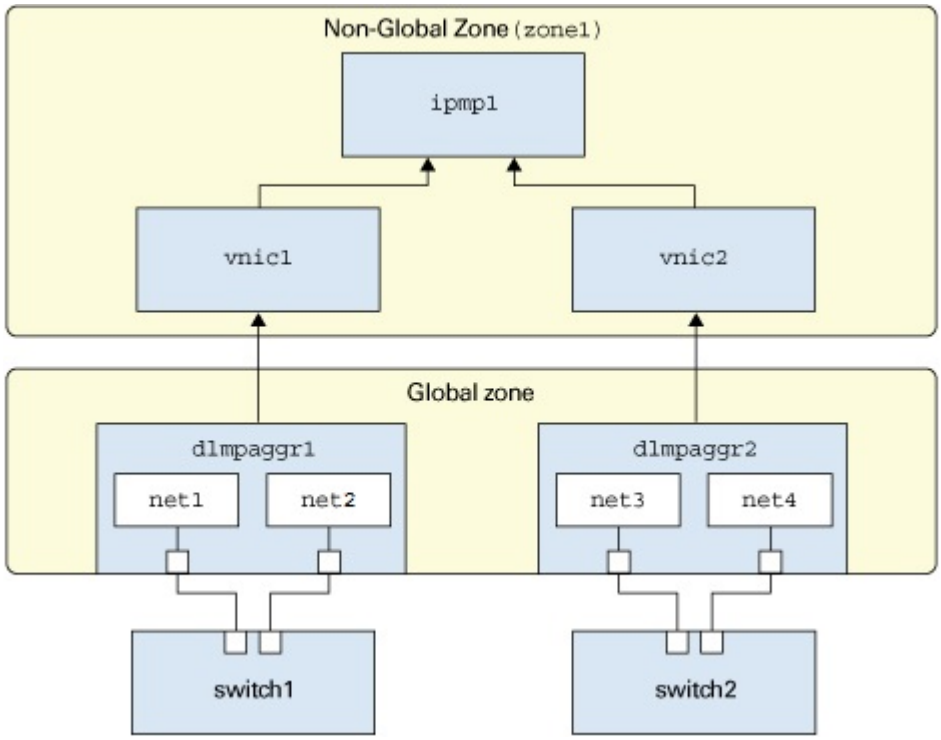
See "How to Configure DLMP Aggregation of InfiniBand HCA Ports" on page 38.

Configuring IPMP Over DLMP in a Virtual Environment

An IPMP group over a DLMP aggregation enhances network performance and availability. The underlying interfaces of the IPMP group must not be part of the same DLMP aggregation. Otherwise, both DLMP and IPMP provide high availability and load spreading over the same interfaces, which becomes redundant and error prone. Thus, IPMP groups can be created only from interfaces that are coming from different DLMP aggregations. See the examples in Figure 7, "Configuration of IPMP Over DLMP in Oracle VM Server for SPARC," on page 31 and Figure 6, "Configuration of IPMP Over DLMP in Zones," on page 30.

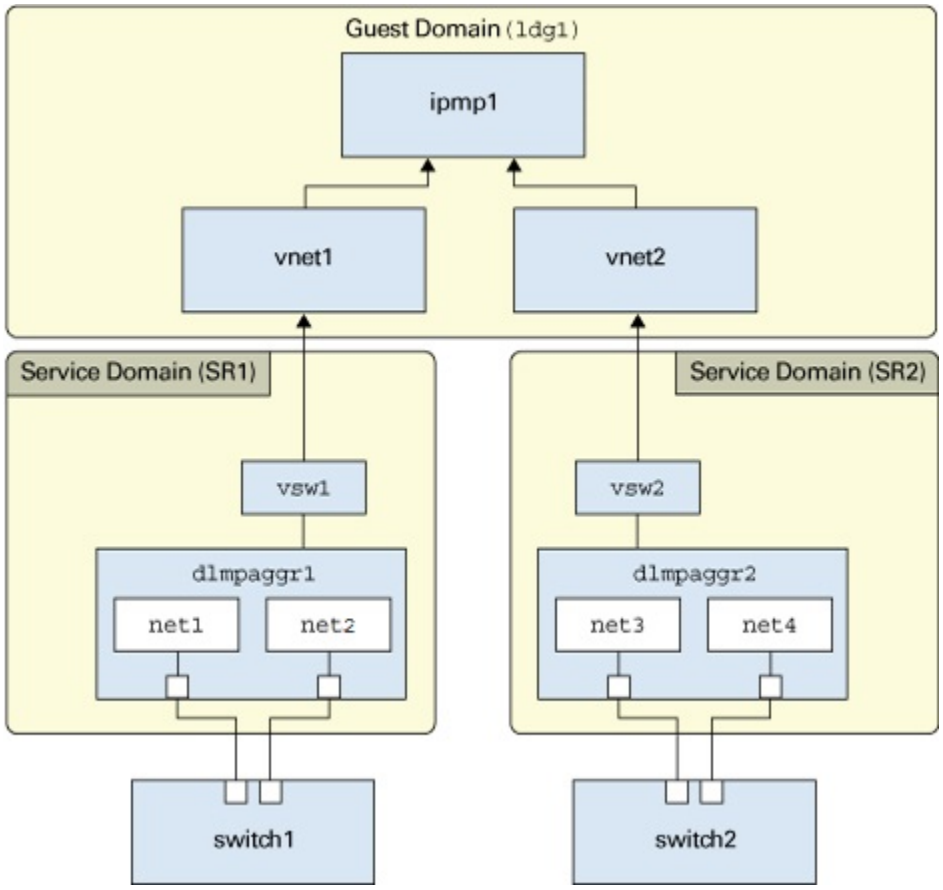
In virtual environments, creating an IPMP group over interfaces coming from non overlapping DLMP aggregations can offer better network performance. High availability is provided by the DLMP aggregation at the L2 layer and the load spreading is managed by the IPMP group at the L3 layer across multiple switches. The next two figures show examples of a combined IPMP and DLMP configuration. The following figure illustrates a configuration on a system with both the global and a non-global zone.

FIGURE 6 Configuration of IPMP Over DLMP in Zones



This next figure shows the same combined configuration of IPMP and DLMP but deployed on an Oracle VM Server for SPARC.

FIGURE 7 Configuration of IPMP Over DLMP in Oracle VM Server for SPARC



Requirements for Link Aggregations

Link aggregations must meet the following requirements:

- The datalinks to be configured into an aggregation should not have any IP interface configured over them.

- You can create link aggregations only from the global zone where the underlying links are physical NICs. By contrast, interfaces in non-global zones are virtual. Therefore, you cannot create a link aggregation from the non-global zone.
- All the datalinks in the aggregation must run at the same speed and in full-duplex mode.
- For DLMP aggregations, you must have at least one switch to connect the aggregation to the ports in the other systems. You cannot use a back-to-back setup when configuring DLMP aggregations.
- On SPARC based systems, each datalink must have its own unique MAC address. For information, see "How to Ensure That the MAC Address of Each Interface Is Unique" in *Configuring and Managing Network Components in Oracle Solaris 11.4*.
- (Trunk aggregation only) Aggregating devices must support link state notification as defined in the IEEE 802.3ad Link Aggregation Standard in order for a port to attach to a trunk aggregation or to detach from a trunk aggregation. Devices that do not support link state notification can be aggregated only by using the `-f` option of the `dladm create-aggr` command. For such devices, the link state is always reported as `UP`. For information about the use of the `-f` option, see "How to Create a Link Aggregation" on page 34.

♦ ♦ ♦ CHAPTER 3

Link Aggregation Configuration Tasks

This chapter describes different procedures to configure and administer link aggregations. It contains the following topics.

- "Creating a Link Aggregation"
- "Adding a Link to an Aggregation"
- "Removing a Link From an Aggregation"
- "Modifying a Trunk Aggregation Attributes"
- "Configuring Probe-Based Failure Detection for DLMP Aggregation"
- "Monitoring Probe-Based Failure Detection"
- "Deleting a Link Aggregation"
- "Use Case: Configuring a Link Aggregation"

Note - To administer link aggregations and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 16.

Creating a Link Aggregation

In a link aggregation, the underlying ports become a single logical group for the aggregation's exclusive use. Thus, the ports cannot be used for anything else. However, you can configure VNICs on top of the aggregation itself.

Before creating the link aggregation, you must remove any existing IP interface on these ports.

Note - Link aggregation works only on full-duplex, point-to-point links that operate at identical speeds. Make sure that the datalinks in your aggregation conform to this requirement.

How to Create a Link Aggregation

Before You Begin

If you are using a switch with a trunk aggregation, configure also the switch's ports to be used as an aggregation. If the switch supports LACP, configure LACP to either `active` or `passive` mode. See the switch manufacturer's documentation to configure the switch.

Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 16.

1. **Display the datalink information to identify the physical datalinks for the aggregation.**

   ```
   $ dladm show-phys
   ```

2. **Ensure that the datalinks that you intend to aggregate are not in use by any application.**

   For example, if an IP interface is created over the datalink, remove the IP interface first.

   a. **Determine the link state.**

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

   The output indicates that an IP interface `net0` currently exists over the `net0` datalink.

   Note - By default, an IP interface shares the same name as the datalink over which it is created.

   b. **Remove the IP interface.**

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

3. **Create the link aggregation.**

   ```
   $ dladm create-aggr [-f] [-m mode] [-P policy]  [-L LACP-mode] \
   [-T time] [-u address] -l link1 -l link2 [...] aggr
   ```

   `-f` — Forces the creation of the aggregation. Use this option when you are attempting to aggregate devices that do not support link state notification.

   `-m` *mode* — Mode must be set to one of the following values.

   - `trunk` – IEEE 802.3ad compliant link aggregation mode (default)

- `dlmp` – Datalink multipathing mode

`-P` *policy* — (Trunk aggregation only) Specifies the load balancing policy for the aggregation. Supported values are `L2`, `L3`, and `L4`. For more information, see "Defining Aggregation Policies for Load Balancing" on page 24.

`-L` *LACP-mode* — (Trunk aggregation only) Specifies the mode of LACP if it is used. Supported values are `off`, `active`, or `passive`. For information about the modes, see "Using a Switch" on page 21.

`-T` *time* — (Trunk aggregation only) Specifies the LACP timer value. The supported values are `short` or `long`.

`-u` *address* — Specifies the fixed unicast address for the aggregation.

`-l` *linkn* — Specifies the datalinks that you want to aggregate.

aggr — Specifies the name of the aggregation. For information about the rules to assign names, see "Rules for Valid Link Names" in *Configuring and Managing Network Components in Oracle Solaris 11.4*.

4. **(Optional) Check the status of the aggregation that you created.**

- Display the aggregations and links with the status information.

```
$ dladm show-link
```

- Display the aggregations with the status and per port information.

```
$ dladm show-aggr -x
```

Example 1 Creating a Trunk Aggregation

This example shows how to create an aggregation following the steps in the previous task. The example begins with the removal of existing IP interfaces over the underlying datalinks.

```
$ ipadm show-if
IFNAME       CLASS        STATE     ACTIVE     OVER
lo0          loopback     ok        yes        --
net0         ip           ok        no         --
$ ipadm delete-ip net0
$ dladm create-aggr -L active -l net0 -l net1 trunk0
$ dladm show-aggr -x
LINK       PORT   SPEED   DUPLEX  STATE    ADDRESS            PORTSTATE
trunk0     --     1000Mb  full    up       8:0:27:49:10:b8    --
           net0   1000Mb  full    up       8:0:27:49:10:b8    attached
```

```
net1   1000Mb  full    up       8:0:27:e4:d9:46    attached
```

Example 2 Creating a DLMP Aggregation and Configuring an IP Interface on Top of the Aggregation

This example shows how to create a DLMP aggregation including configuring VNICs over the aggregation.

```
$ dladm create-aggr -m dlmp -l net0 -l net1 -l net2 aggr0
$ dladm show-link
LINK      CLASS     MTU     STATE   OVER
net0      phys      1500    up      --
net1      phys      1500    up      --
net2      phys      1500    up      --
aggr0     aggr      1500    up      net0 net1 net2
$ dladm show-aggr -x
LINK       PORT           SPEED DUPLEX   STATE     ADDRESS            PORTSTATE
aggr0      --             1000Mb full    up        2:8:20:c2:24:45    --
           net0           1000Mb full    up        8:0:27:49:10:b8    attached
           net1           1000Mb full    up        8:0:27:e4:d9:46    attached
           net2           1000Mb full    up        8:0:27:38:7a:97    attached
$ ipadm create-ip aggr0
$ ipadm create-addr -a local=203.0.113.1 aggr0/v4
$ dladm create-vnic -l aggr0 vnic1
```

Next Steps You can perform additional configuration tasks on the aggregation:

- Create IP interfaces and VNICs. See Chapter 3, “Configuring and Administering IP Interfaces and Addresses in Oracle Solaris” in *Configuring and Managing Network Components in Oracle Solaris 11.4* and “How to Configure VNICs and Etherstubs” in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.
- Configure VLANs. See “Configuring VLANs Over a Link Aggregation” on page 61.
- Configure zones. See *Creating and Using Oracle Solaris Zones*.

▼ How to Configure DLMP Aggregation of SR-IOV NICs

To configure a DLMP aggregation using SR-IOV NICs, the aggregated datalinks must have their `iov` property set to `on`. However, the VNICs that you created on the aggregation must have their `iov` property set to either `off` or `inherit`. If the property is set to `on`, the VNICs cannot be created.

You can dynamically add or remove ports without deleting the existing configuration or stopping the network connections.

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 16.

1. **Set the datalinks to be aggregated to be SR-IOV-enabled.**

   ```
   $ dladm set-linkprop -p iov=on link1
   $ dladm set-linkprop -p iov=on link2
   ```

2. **Create an aggregation of SR-IOV-enabled datalinks.**

   ```
   $ dladm create-aggr -l link1 -l link2 -m dlmp aggr
   ```

3. **Configure a VNIC on top of the aggregation.**

   ```
   $ dladm create-vnic -l aggr [-p iov=off] VNIC
   ```

4. **Display the VNIC to verify that the VNIC is associated with a VF.**

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

Example 3 Configuring a DLMP Aggregation of SR-IOV NICs

This example shows how to create DLMP aggregation with two underlying links `net0` and `net1`. The `iov` property of `net0` and `net1` is set to `on`.

```
$ dladm set-linkprop -p iov=on net0
$ dladm set-linkprop -p iov=on net1
$ dladm create-aggr -l net0 -l net1 -m dlmp dlmp0
$ dladm create-vnic -l dlmp0 vnic1
$ dladm create-vnic -l dlmp0 -p iov=off vnic2
$ dladm create-vnic -l dlmp0 -p iov=inherit vnic3
$ dladm show-vnic -V vnic1
LINK        OVER           VF-ASSIGNED
vnic1       dlmp0          ixgbevf0
vnic2       dlmp0          --
vnic3       dlmp0          ixgbevf1
$ dladm show-aggr -C
LINK       PORT           SPEED   DUPLEX   STATE     CLIENTS
dlmp0      --             1000Mb  full     up         --
           net0           1000Mb  full     up        dlmp0,vnic1
           net1           1000Mb full      up        vnic2,vnic3
```

The `dladm show-aggr -C` commands shows clients associated with each aggregated port. For more information, see "Displaying the Clients Associated With Each Port" on page 39.

How to Configure DLMP Aggregation of InfiniBand HCA Ports

You can create a DLMP aggregation of InfiniBand HCA ports. However, you cannot create an aggregation with the combination of Ethernet datalinks and InfiniBand HCA ports together.

In the global zone, you can specify virtual HCA as a member port of DLMP aggregation, even though the virtual HCA port does not share the same physical HCA with the other member ports of DLMP aggregation.

Note - A DLMP aggregation cannot have more than one virtual port (including the physical port) that belongs to a physical HCA port.

Likewise, you cannot create an InfiniBand DLMP aggregation inside a kernel 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 16.

1. **Display the InfiniBand datalinks to determine the ports to be aggregated.**

```
$ dladm show-ib
LINK    HCAGUID        PORTGUID       PORT STATE   GWNAME  GWPORT   PKEYS
net4    21280001EFA276 21280001EFA277 1    up      --      --       FFFF,8001
net5    21280001EFA276 21280001EFA278 2    up      --      --       FFFF,8001
```

2. **Create a DLMP aggregation of InfiniBand datalinks.**

```
$ dladm create-aggr -l net4 -l net5 -m dlmp dlmp_ib0
```

3. **Display the aggregation.**

```
$ dladm show-aggr -x dlmp_ib0
LINK       PORT    SPEED   DUPLEX  STATE     ADDRESS    PORTSTATE
dlmp_ib0   --      32000Mb full    up        unknown    --
           net4    32000Mb full    up        unknown    attached
           net5    32000Mb full    up        unknown    attached
```

4. **Configure an IPoIB VNIC over the aggregation.**

```
$ dladm create-vnic -l dlmp_ib0 -P ffff ibvnic1
```

5. **Display the IPoIB VNIC.**

```
$ dladm show-vnic ibvnic1
```

```
LINK        OVER       SPEED  MACADDRESS        MACADDRTYPE IDS
ibvnic1     dlmp_ib0   32000  80:0:0:4a:fe:..   fixed       PKEY:0xffff
```

6. **Display aggregation information.**

   - **To display the IP interface associated with each port, type the following command:**

     ```
     $ dladm show-aggr -C dlmp_ib0
     LINK       PORT           SPEED DUPLEX   STATE     CLIENTS
     dlmp_ib0   --             32000Mb full    up       --
                net4           32000Mb full    up       ibvnic1
                net5           32000Mb full    up       --
     ```

   - **To display the list of PKEYs associated with DLMP aggregation of InfiniBand HCA ports, type the following command:**

     ```
     $ dladm show-aggr -v
     LINK              MODE  IDS
     dlmp_ib0          dlmp  PKEY:0xFFFF,0x8001
     trunk1            trunk --
     ```

Displaying the Clients Associated With Each Port

Each client (IP interface) is associated with exactly one DLMP aggregated port. Only the port receives or transmits traffic for that client.

Load spreading is implemented across all the VNICs of a DLMP aggregation. If actions such as port addition, port removal, port failure, or port recovery cause a load imbalance, the load distribution is rebalanced across the VNICs.

The following command displays the list of clients associated with each aggregated port.

```
$ dladm show-aggr -C
LINK     PORT    SPEED     DUPLEX   STATE     CLIENTS
dlmp0     --     10000Mb   full     up        --
         net1    10000Mb   full     up        dlmp0,vnic2
         net2    10000Mb   full     up        vnic1
```

In the example, DLMP aggregation `dlmp0` has two ports `net1` and `net2`. The VNICs `vnic1` and `vnic2` are created over `dlmp0`. The clients of the DLMP aggregation `dlmp0` and `vnic2` are associated with the port `net1` and the client `vnic1` is associated with the port `net2`.

Adding a Link to an Aggregation

If you are adding datalinks to a trunk aggregation, you might have to reconfigure the switch to accommodate the additional datalinks even though LACP is configured on the switch.

▼ How to Add a Link to an Aggregation

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 16.

1. **Delete any IP interfaces configured over the link that you want to add to the aggregation.**

   ```
   $ ipadm show-if
   $ ipadm delete-ip interface
   ```

2. **Add the link to the aggregation.**

   ```
   $ dladm add-aggr -l link [-l link] [...] aggr
   ```

3. **(Trunk aggregations only) If necessary, reconfigure the switch.**

   See the switch manufacturer's documentation to perform any reconfiguration tasks on the switch.

Example 4 Adding a Link to an Aggregation

This example shows how to add a link to the aggregation `aggr0`.

```
$ dladm show-link
LINK     CLASS     MTU     STATE     OVER
net0     phys      1500    up        --
net1     phys      1500    up        --
aggr0    aggr      1500    up        net0 net1
net3     phys      1500    up        --
$ ipadm delete-ip net3
$ dladm add-aggr -l net3 aggr0
$ dladm show-link
LINK     CLASS     MTU     STATE     OVER
net0     phys      1500    up        --
net1     phys      1500    up        --
aggr0    aggr      1500    up        net0 net1 net3
```

```
net3     phys      1500    up        --
```

Removing a Link From an Aggregation

When you remove links from an aggregation, you might need to reconfigure the switch to conform to the new setting.

To remove links, use the following syntax:

```
$ dladm remove-aggr -l link aggr
```

EXAMPLE 5 Removing a Link From an Aggregation

In this example, `net3` is removed from `aggr0`.

```
$ dladm show-link
LINK     CLASS     MTU     STATE     OVER
net0     phys      1500    up        --
net1     phys      1500    up        --
aggr0    aggr      1500    up        net0 net1 net3
net3     phys      1500    up        --
$ dladm remove-aggr -l net3 aggr0
$ dladm show-link
LINK     CLASS     MTU     STATE     OVER
net0     phys      1500    up        --
net1     phys      1500    up        --
aggr0    aggr      1500    up        net0 net1
net3     phys      1500    unknown   --
```

Modifying a Trunk Aggregation Attributes

Modifying `policy`, `lacpmode`, and `time` are applicable only to trunk aggregations. These attributes are not supported on DLMP aggregations.

You can modify these attributes even when the aggregation is active, or even with a configured IP interface or VNIC existing over the aggregation.

- To modify the load balancing policy of the aggregation, type:

  ```
  $ dladm modify-aggr -P policy aggr
  ```

policy Represents one or more of the load balancing policies L2, L3, and L4, as explained in "Defining Aggregation Policies for Load Balancing" on page 24.

aggr Specifies the aggregation whose attributes you want to modify.

- To modify the LACP mode of the aggregation, type:

```
$ dladm modify-aggr -L LACP-mode -T time aggr
```

-L *LACP-mode* Indicates the LACP mode in which the aggregation must operate. Possible values are `active`, `passive`, and `off`.

-T *time* Indicates the LACP timer value, either `short` or `long`.

EXAMPLE 6 Modifying a Trunk Aggregation

This example shows how to modify the load balancing policy and LACP mode of a link aggregation.

```
$ dladm modify-aggr -P L2 aggr0
$ dladm modify-aggr -L active -T short aggr0
$ dladm show-aggr
LINK    MODE       POLICY  ADDRPOLICY      LACPACTIVITY   LACPTIMER
aggr0   trunk      L2      auto            active         short
```

Configuring Probe-Based Failure Detection for DLMP Aggregation

In DLMP aggregations, probing is disabled by default and only link-based failure detection is used. To enable probing, you must configure the `probe-ip` property.

In addition to `probe-ip`, `probe-vlan-id` and `probe-fdt` are also configurable properties related to probe-based failure detection.

▼ How to Configure Probe-Based Failure Detection

This procedure applies only to DLMP aggregations and assumes that one already exists.

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 16.

1. **(Optional) Display all the existing aggregations to identify the aggregation for configuring probe-based failure detection.**

   ```
   $ dladm show-aggr
   ```

2. **Configure probing for the aggregation by either using automatically selected source and target IP address or by manual selection.**

   - **To use automatic selection, issue the following command:**

     If you do not specify target addresses, the DLMP aggregation chooses a target IP automatically from one of the next hop routers that are on the same subnet as the source IP address.

     ```
     $ dladm set-linkprop -p probe-ip=+ aggr
     ```

     With this syntax, you do not specify source or target IP addresses. The process automatically selects both source address and target addresses for the probing. The source address is selected from configured addresses in the global zone. A target address is selected from one of the next hop routers that are on the same subnet as the source address.

Note - Probing using automatic selection is a best practice method of setting up probe-based failure detection.

   - **To use manual selection, issue the following command:**

     ```
     $ dladm set-linkprop -p probe-ip=[source[,...]]+[target[,...]] aggr
     ```

     source Specifies the source IP address for probing. Only configured addresses in the global zone can be source addresses. You can format the source IP address in any one of the following ways:

     - `probe-ip=`*IP-address*[/*prefix-length*]+

       IP address and its prefix length. For example, `203.0.113.1/24+`.

     - `probe-ip=`*addr-obj-name*+

       Address object name. For example, `vnic1/addr1+`.

     - `probe-ip=`*interface-name*+

       IP Interface name, which can be the aggregated interface itself or any VNIC over the aggregation. For example, `aggr1+`. The IP interface must have a configured IP address.

- `probe-ip=`*hostname*`+`

  Host name. For example, `sdg1+`

target — Specifies the target IP address for probing. When you specify the target IP address, it must be in any one of the following ways:

- `probe-ip=+`*IP-address*

  For example, `+203.0.113.1`.
- `probe-ip=+`*hostname*

  For example, `+sdg1`

aggr — Name of the aggregation.

Note - Specifying the source and the target IP addresses is optional. If you do not specify the source and the IP target addresses, the source and the probing IP addresses are automatically selected.

3. (Optional) Set the VLAN ID for the probes.

```
$ dladm set-linkprop -p probe-vlan-id=probe-VID  aggr
```

probe-VID is the VLAN ID to be used for both ICMP and transitive probes. The valid values of *probe-VID* ranges from `0` to `4094`. The *probe-VID* value `0` indicates that the probes are untagged. The default value is `0`.

4. (Optional) Set the failure detection time.

```
$ dladm set-linkprop -p probe-fdt=fdt  aggr
```

fdt is the failure detection time specified in seconds. The default value is 10 seconds.

5. (Optional) Display the probe statistics.

```
$ dlstat show-aggr -n -P [[t],[i],[all]]
```

The `-P` specifies the type of the probe to display. You can specify any of the following in a comma-separated list as the probe type:

- `t` – Displays the transitive probes.
- `i` – Displays the ICMP probes.
- `all` – Displays both the transitive and ICMP probes.

Example 7 Configuring Probe-Based Failure Detection

This example shows how you can configure probe-based failure detection by using the default automatic selection of source and probing IP addresses.

```
$ dladm create-aggr -m dlmp -l net0 -l net1 -l net2 aggr1
$ dladm show-aggr -x
LINK        PORT           SPEED DUPLEX   STATE     ADDRESS            PORTSTATE
aggr0       --             1000Mb full    up        2:8:20:c2:24:45    --
            net0           1000Mb full    up        8:0:27:49:10:b8    attached
            net1           1000Mb full    up        8:0:27:e4:d9:46    attached
            net2           1000Mb full    up        8:0:27:38:7a:97    attached

$ ipadm create-ip aggr1
$ ipadm create-addr -a 192.0.2.137/24 aggr1

$ dladm set-linkprop -p probe-ip=+ aggr1
$ dladm set-linkprop -p probe-vlan-id=2100 aggr1
$ dladm set-linkprop -p probe-fdt=15 aggr1

$ dladm show-linkprop -p probe-ip,probe-vlan-id,probe-fdt aggr1
LINK     PROPERTY         PERM   VALUE      EFFECTIVE       DEFAULT      POSSIBLE
aggr1    probe-ip         rw     +           +               -            --
aggr1    probe-vlan-id    rw     2100        2100             0           0-4094
aggr1    probe-fdt        rw     15          15              10           1-600

$ dlstat show-aggr -n -P t,i aggr1
TIME   AGGR   PORT  LOCAL           TARGET          PROBE  NETRTT     RTT
0.45s  aggr1  net0  net0            net1            t16148  --         --
0.45s  aggr1  net0  net0            net1            t16148  0.63ms     0.81ms
1.08s  aggr1  net1  net1            net0            t16148  --         --
1.08s  aggr1  net1  net1            net0            t16148  0.72ms     0.99ms
2.07s  aggr1  net1  192.0.2.137     192.0.2.137     i15535  --         --
2.07s  aggr1  net1  192.0.2.137     192.0.2.137     i15535  0.18ms     0.54ms
```

Monitoring Probe-Based Failure Detection

To display probe-related information, use the `dlstat show-aggr` command.

```
# dlstat show-aggr -n -P [[t],[i],[all]]
```

where `-P` specifies the probe type to display. Specify any of the following in a comma-separated list as the probe type:

- `t` – Displays the transitive probes.
- `i` – Displays ICMP probes.
- `all` – Displays both the transitive and ICMP probes.

The `dladm show-aggr` and `ipadm show-addr` can also used to display aggregation information.

EXAMPLE 8 Displaying Probe-Related Information

The following example displays the statistics of the probes for the DLMP aggregation, `aggr1`.

```
$ dlstat show-aggr -n -P t,i aggr1
TIME     AGGR     PORT   LOCAL        TARGET       PROBE  NETRTT     RTT
0.53s    aggr1    net0   net0         net1         t16148  --         --
0.53s    aggr1    net0   net0         net1         t16148  0.62ms     0.87ms
1.17s    aggr1    net1   net1         net0         t16148  --         --
1.17s    aggr1    net1   net1         net0         t16148  0.72ms     0.99ms
2.24s    aggr1    net1   192.0.2.1    192.0.2.2    i15535  --         --
2.24s    aggr1    net1   192.0.2.1    192.0.2.2    i15535  0.11ms     0.55ms
```

For a description of the output information, see the `dlstat`(8) man page.

EXAMPLE 9 Displaying Detailed Information About the Aggregated Port

The following example displays the detailed aggregation information on each underlying port.

```
$ dladm show-aggr -x
LINK       PORT   SPEED  DUPLEX   STATE   ADDRESS            PORTSTATE
aggr1      --     100Mb  full     up      2:8:20:95:25:d0  --
           net0   100Mb  full     up      1e:34:db:fa:50:a2  attached
           net1   100Mb  full     up      b2:c0:6a:3e:c5:b5  attached
```

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

EXAMPLE 10 Displaying the State of the Aggregated Ports

The following example shows the state of the ports of the aggregation and the target IP addresses of the ports.

```
$ dladm show-aggr -S -n
LINK       PORT       FLAGS   STATE    TARGETS        XTARGETS
aggr1      net1       u--3    active   192.0.2.2      net0
--         net0       u-2-    active   --             net1
```

EXAMPLE 11 Displaying `probe-ip` Property Values

The following example displays the details about the link property `probe-ip` for the specified DLMP aggregation.

```
$ dladm show-linkprop -p probe-ip aggr1
LINK     PROPERTY         PERM   VALUE         EFFECTIVE    DEFAULT      POSSIBLE
aggr1    probe-ip         rw     192.0.2.2     192.0.2.2    --           --
```

EXAMPLE 12 Displaying the IP Address and the State of the Aggregation

The following example displays the IP address and the state of the aggregation.

```
$ ipadm show-addr aggr1
ADDROBJ           TYPE     STATE        ADDR
aggr1/local1      static   ok           192.0.2.1/24
```

Deleting a Link Aggregation

Before deleting the aggregation, remove its configured IP interface and VNICs first.

Note - When you remove a datalink by using the `dladm delete-phys` command, any aggregation (a layer 2 component) that is configured over the datalink is also deleted. This `dladm delete-phys` command also deletes the other layer 2 components such as flows, VNICs, and VLANs configured over the datalink. However, you must manually delete the layer 3 components configured over the datalink such as IP interfaces by using the `ipadm` command.

▼ How to Delete a Link Aggregation

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 16.

1. **Delete the IP interface that is configured over the link aggregation.**

   ```
   $ ipadm delete-ip IP-aggr
   ```

   where *IP-aggr* is the IP interface over the link aggregation.

2. **Delete the link aggregation.**

```
$ dladm delete-aggr aggr
```

Example 13 Deleting a Link Aggregation

This example shows how to delete the aggregation `aggr0`. The deletion is persistent.

```
$ ipadm delete-ip aggr0
$ dladm delete-aggr aggr0
```

Use Case: Configuring a Link Aggregation

The following end-to-end use case shows how to accomplish the following actions:

- Create a DLMP aggregation and add links to it.
- Configure the aggregation's IP interface.
- Configure a VNIC over the aggregation.
- Configure probe-based failure detection with automatic selection of source and target addresses.
- Monitor the ICMP and transitive probes.

This example assumes that the system's routing table is already configured so the target address can be automatically selected.

```
$ 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-aggr -m dlmp -l net0 -l net1 aggr1
$ dladm add-aggr -l net2 aggr1
$ ipadm create-ip aggr1
$ ipadm create-addr -a local=203.0.113.1 aggr1/v4
$ dladm create-vnic -l aggr1 vnic1
$ dladm set-linkprop -p probe-ip=+ aggr1
```

```
$ dladm show-aggr -S
LINK       PORT        FLAGS   STATE     TARGETS         XTARGETS
aggr1      net0        u--3    active    203.0.113.2     net2 net1
--         net1        u-2-    active       --           net2 net0
--         net2        u-2-    active       --           net0 net1

$ dlstat show-aggr -n -P i
   TIME     AGGR      PORT   LOCAL          TARGET        PROBE NETRTT  RTT
   1.16s    aggr1     net0   203.0.113.1     203.0.113.2    i33   --      --
   1.16s    aggr1     net0   203.0.113.1     203.0.113.2    i33   0.08ms  0.33ms
   2.05s    aggr1     net0   203.0.113.1     203.0.113.2    i34   --      --
   2.05s    aggr1     net0   203.0.113.1     203.0.113.2    i34   0.01ms  0.64ms
   4.05s    aggr1     net0   203.0.113.1     203.0.113.2    i35   --      --
   4.05s    aggr1     net0   203.0.113.1     203.0.113.2    i35   0.10ms  0.35ms
   5.54s    aggr1     net0   203.0.113.1     203.0.113.2    i36   --      --
   5.54s    aggr1     net0   203.0.113.1     203.0.113.2    i36   0.08ms  0.34ms

$ dlstat show-aggr -n -P t
   TIME     AGGR   PORT        LOCAL       TARGET  PROBE NETRTT  RTT
   0.30s    aggr1  net2        net2        net0    t38   --      --
   0.30s    aggr1  net2        net2        net0    t38   0.46ms  0.59ms
   0.46s    aggr1  net0        net0        net1    t39   --      --
   0.46s    aggr1  net0        net0        net1    t39   0.46ms  0.50ms
   0.48s    aggr1  net1        net1        net0    t39   --      --
   0.48s    aggr1  net1        net1        net0    t39   0.34ms  0.38ms
   0.72s    aggr1  net2        net2        net1    t38   --      --
   0.72s    aggr1  net2        net2        net1    t38   0.38ms  0.42ms
   0.76s    aggr1  net0        net0        net2    t39   --      --
   0.76s    aggr1  net0        net0        net2    t39   0.33ms  0.38ms
   0.87s    aggr1  net1        net1        net2    t39   --      --
   0.87s    aggr1  net1        net1        net2    t39   0.32ms  0.38ms
   1.95s    aggr1  net2        net2        net0    t39   --      --
   1.95s    aggr1  net2        net2        net0    t39   0.36ms  0.42ms
   1.97s    aggr1  net2        net2        net1    t39   --      --
   1.97s    aggr1  net2        net2        net1    t39   0.32ms  0.38ms
   1.99s    aggr1  net0        net0        net1    t40   --      --
   1.99s    aggr1  net0        net0        net1    t40   0.31ms  0.36ms
   2.12s    aggr1  net1        net1        net0    t40   --      --
   2.12s    aggr1  net1        net1        net0    t40   0.34ms  0.40ms
   2.14s    aggr1  net0        net0        net2    t40   --      --
```

The aggregation `aggr0` with an IP interface configured over it is created. The VNIC `vnic1` is configured on top of the aggregation `aggr0`. Probe-based failure detection is configured without specifying either the source IP address or the target IP address of the probes. To enable probing, the target in the routing table is configured with an IP address, `203.0.113.2`, on the same subnet as the specified IP address, `203.0.113.1`. The ICMP and transitive probe statistics are monitored.

♦ ♦ ♦ CHAPTER 4

Configuring Virtual Networks by Using Virtual Local Area Networks

This chapter discusses about virtual local area networks (VLANs) and also describes the procedures to configure and modify VLANs.

This chapter contains the following topics:

- "Deploying VLANs"
- "Using VLANs With Virtualization"
- "Planning a VLAN Configuration"
- "Configuring a VLAN"
- "Configuring VLANs Over a Link Aggregation"
- "Displaying VLAN Information"
- "Modifying VLANs"
- "Deleting a VLAN"
- "Using VLAN Tagging for SR-IOV-Enabled Ports"
- "Use Cases in of VLAN Deployments"

Note - To administer VLANs and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 16.

Deploying VLANs

A virtual local area network (VLAN) is a subdivision of a local area network at the datalink layer of the protocol stack. VLANs split a single L2 layer network into multiple logical networks such that each logical network is its own broadcast domain. All the devices connected to a VLAN can send broadcast frames to each other regardless of their physical location or their connection to the same physical switch.

In Oracle VM Server for SPARC, the network infrastructure supports 802.1Q VLAN-Tagging. The virtual switch (`vsw`) and virtual network (`vnet`) devices support switching of Ethernet packets based on the VLAN ID and handle the necessary tagging or untagging of Ethernet frames. For more information, see "Using VLAN Tagging" in *Oracle VM Server for SPARC 3.6 Administration Guide*.

When to Use VLANs

Deploy VLANs if you need to do the following:

- Create a logical division of workgroups.

  For example, if all systems on a floor of a building are connected on one switch-based local network, you can create a separate VLAN for each workgroup on the floor.
- Enforce differing security policies for the workgroups.

  For example, a finance department and an information technology department have different security requirements. You can create a separate VLAN for each department and enforce the appropriate security policy on a per-VLAN basis.
- Reduce the size of broadcast domain and improve network efficiency. With VLANs, you split workgroups into manageable broadcast domains.

  For example, in a broadcast domain consisting of 25 users, if the broadcast traffic is intended only for 12 users, then setting up a separate VLAN for those 12 users can reduce traffic and improve network efficiency.

About VLAN IDs

A VLAN is identified through its name and a VLAN ID. You assign the VLAN name and its VLAN ID during configuration. Then, on the switch, you also assign a VLAN ID to each port.

The port VLAN ID must be the same as the VLAN ID assigned to the interface that connects to the port. If port IDs and their corresponding VLANs do not match, packets might go to the wrong destinations. See "How Datalink VLAN ID Mismatch Errors Are Detected" in *Troubleshooting Network Administration Issues in Oracle Solaris 11.4*.

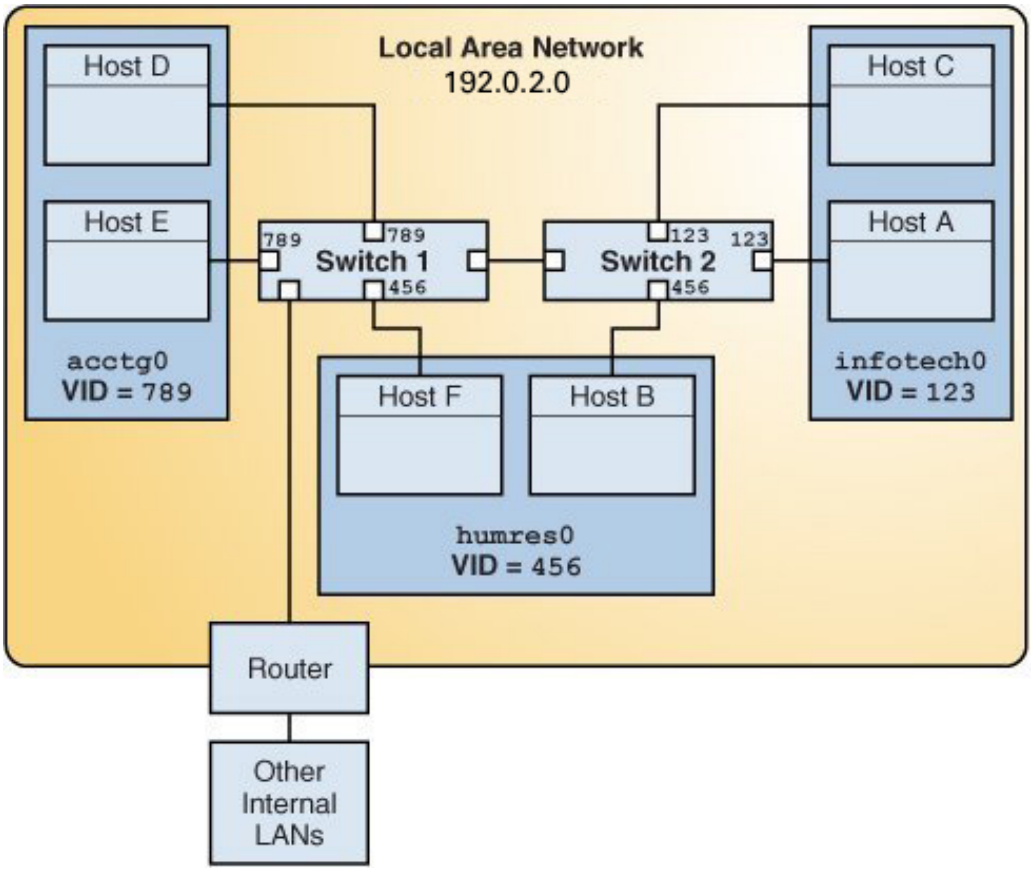
VLAN Topology

VLANs require switches that support the VLAN technology. Switch ports should be configured according to the VLAN topology you adopt. Each switch manufacturer has different procedures

for configuring ports on a switch. For example, to configure the ports of Oracle Switch ES1-24 for VLAN, see "Sun Ethernet Fabric Operating System" in *VLAN Administration Guide*, .

The following figure shows a local area network that has been divided into three VLANs. The setup includes the use of two switches.

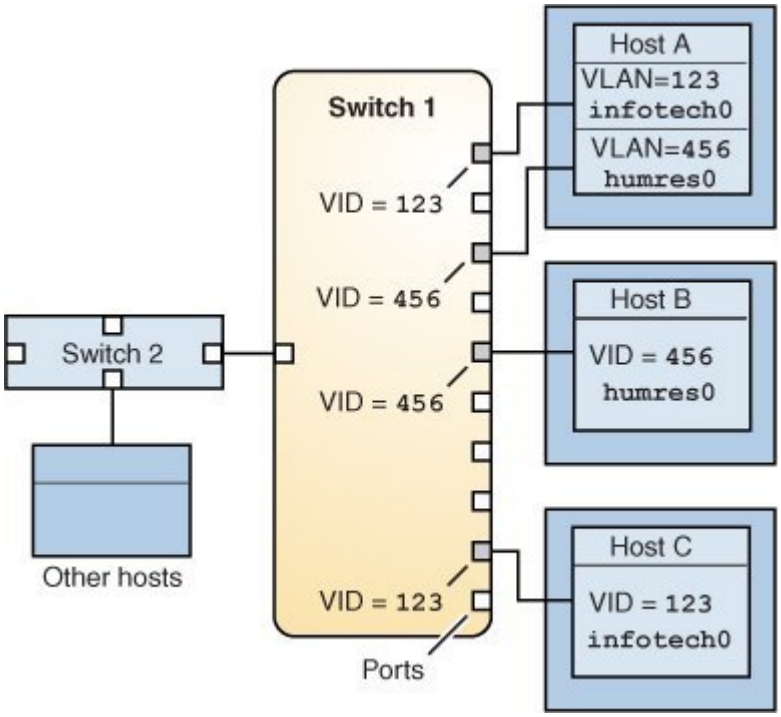
FIGURE 8 Local Area Network With Three VLANs



In the figure, the VLANs correspond to three working groups: accounting (`acctg0`), human resources (`humres0`), and information technology (`infotech0`). Different host systems belong to each VLAN. VLAN member hosts connect to each other through the switch ports that are configured with their specific VLAN IDs.

The next figure is a variation of the previous figure, where only one switch is used to support the VLAN infrastructure.

FIGURE 9 A Switch Connecting Multiple Hosts of Different VLANs



The illustration especially shows that a single host can belong to multiple VLANs. For example, Host A has two configured VLANs, one with ID `123` and the second VLAN with ID `456`. Each VLAN is connected to a switch port with a matching ID. Thus, Host A is a member of both the `infotech0` and the `humres0` VLANs.

Planning a VLAN Configuration

Planning a VLAN configuration involves the following steps:

1. Examine the LAN topology and determine where subdividing into VLANs is appropriate.

2. Create a numbering scheme for the VLAN IDs, and assign a VLAN ID to each VLAN.

Note - If VLAN numbering scheme already exists on the network, you must create VLAN IDs within the existing VLAN numbering scheme.

---

3. On each system, determine which interfaces are the components of a particular VLAN.
   a. Determine which links are configured on the system by using the `dladm show-link` command.
   b. Determine which VLAN ID will be associated with each datalink on the system.
   c. Create the VLAN.
4. Check the connections of the datalinks to the switches of the network.

   Note the VLAN ID of each datalink and the switch port where each interface is connected.
5. Configure each port on the switch with the same VLAN ID as the interface to which it is connected.

   Refer to the switch manufacturer's documentation for configuration instructions.

Configuring a VLAN

The following procedure describes how to create a VLAN over a datalink.

For more technical details about the commands you use to configure VLANs, refer to the `dladm`(8) and `ipadm`(8) man pages.

▼ How to Configure a VLAN

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 16.

1. Determine the types of links that are in use on the system.

```
$ dladm show-link
```

2. Create a VLAN link over a datalink.

```
$ dladm create-vlan -l link -v vid VLAN-link
```

link Specifies the link on which the VLAN interface is being created.

vid Indicates the VLAN ID number.

VLAN-link Specifies the name of the VLAN.

3. **Verify the VLAN configuration.**

```
$ dladm show-vlan
```

4. **Create an IP interface over the VLAN.**

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

interface is the interface name, which typically follows the VLAN name.

5. **Configure the IP interface with an IP address.**

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

Example 14 Creating a VLAN

This example shows how to create the VLAN configuration that is illustrated in Figure 8, "Local Area Network With Three VLANs," on page 53.

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

The following steps are performed on specified hosts:

Host A:

```
$ dladm create-vlan -l net0 -v 123 infotech0
$ ipadm create-ip infotech0
$ ipadm create-addr -a IP-addr infotech0
```

Host C:

```
$ dladm create-vlan -l net0 -v 123 infotech0
$ ipadm create-ip infotech0
$ ipadm create-addr -a IP-addr infotech0
```

Host F:

```
$ dladm create-vlan -l net0 -v 456 humres0
```

```
$ ipadm create-ip humres0
$ ipadm create-addr -a IP-addr humres0
```

Host B:

```
$ dladm create-vlan -l net0 -v 456 humres0
$ ipadm create-ip humres0
$ ipadm create-addr -a IP-addr humres0
```

Host D:

```
$ dladm create-vlan -l net0 -v 789 acctg0
$ ipadm create-ip acctg0
$ ipadm create-addr -a IP-addr acctg0
```

Host E:

```
$ dladm create-vlan -l net0 -v 789 acctg0
$ ipadm create-ip acctg0
$ ipadm create-addr -a IP-addr acctg0
```

Note - The IP addresses must be unique for each VLAN.

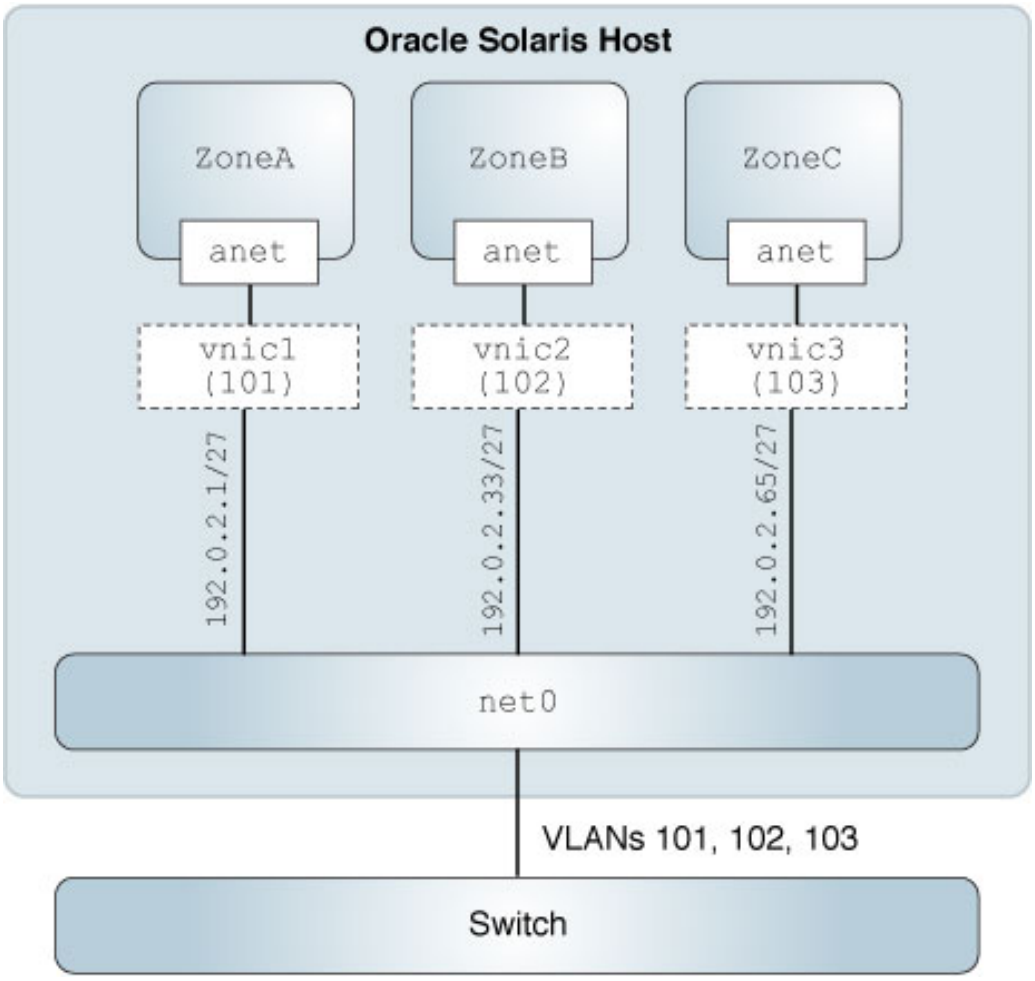
If you issue the command to list configured VLANs, each host would display a screen similar to the following:

```
$ dladm show-vlan
LINK            VID   SVID    PVLAN-TYPE   FLAGS    OVER
infotech0       123    --       --         ----     net0
humres0         456    --       --         ----     net0
acctg0          789    --       --         ----     net0
```

Using VLANs With Virtualization

VLANs and Oracle Solaris Zones, including Oracle Solaris Kernel Zones, can be combined to set up network virtualization. See the following figure that illustrates this type of configuration.

FIGURE 10 VLANs With Zones



For this setup, you create VNICs as VLANs by assigning them VLAN tags or IDs. These VNICs are part of the same L2 broadcast domain as that of the non-global zone. Thus, in the figure, ZoneA, ZoneB, and ZoneC are configured on the host. Their VNICs have corresponding VLAN IDs.

For more detailed information about deploying VLANs in network virtualization, refer to the following resources:

- "How to Configure VNICs as VLANs" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*
- "Configuring Virtual LANs in Kernel Zones" in *Creating and Using Oracle Solaris Kernel Zones*. See also the section *Using Dynamic MAC Addresses and VLAN IDs in Kernel Zones*.
- *Creating and Using Oracle Solaris Zones*

The following two examples show how to configure VLANs and Oracle Solaris zones together. The first uses the method of configuring the `anet` zone property while the second sets up the configuration by directly creating VNICs. Both examples assume that the switch ports are configured for the VLANs and that zones are already created on the system.

EXAMPLE 15 Configuring VLANs With Zone's `anet` Resource

```
solaris$ zonecfg -z ZoneA
      zonecfg:ZoneA> add anet
      zonecfg:ZoneA:anet> set linkname=net0
      zonecfg:ZoneA:anet> set lower-link=net0
      zonecfg:ZoneA:anet> set vlan-id=11
      zonecfg:ZoneA:anet> set allowed-address=192.0.2.10/24
      zonecfg:ZoneA:anet> set defrouter=192.0.2.1
      zonecfg:ZoneA:net> end

solaris$ zonecfg -z ZoneB
      zonecfg:ZoneA> add anet
      zonecfg:ZoneA:anet> set linkname=net1
      zonecfg:ZoneA:anet> set lower-link=net1
      zonecfg:ZoneA:anet> set vlan-id=12
      zonecfg:ZoneA:anet> set allowed-address=192.0.2.20/24
      zonecfg:ZoneA:anet> set defrouter=192.0.2.21
      zonecfg:ZoneA:anet> end

solaris:~$ dladm show-vnic
LINK            OVER           SPEED  MACADDRESS        MACADDRTYPE IDS
ZoneA/anet      net0           1000   2:8:20:47:8c:85   random      VID:11
ZoneB/anet      net1           1000   2:8:20:47:8c:85   random      VID:12
```

EXAMPLE 16 Configuring VNICs as VLANs for Zones

The example proceeds in 3 stages:

1. Creation of VNICs

2. Assigning of a VNIC to each zone
3. Configuring each VNIC's IP interface from inside the specific zone

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

solaris$ dladm create-vnic -v 101 -l net0 vnic1
solaris$ dladm create-vnic -v 102 -l net0 vnic2
solaris$ dladm create-vnic -v 103 -l net0 vnic3
solaris$ dladm show-vnic
LINK      OVER    SPEED   MACADDRESS         MACADDRTYPE   IDS
vnic1     net0    1000    2:8:20:35:b:9a     random        VID:101
vnic2     net0    1000    2:8:20:fa:94:57    random        VID:102
vnic3     net0    1000    2:8:20:51:1c:4a    random        VID:103

solaris$ dladm show-link
LINK      CLASS    MTU      STATE     OVER
net0      phys     1500     up        --
vnic1     vnic     1500     up        net0
vnic2     vnic     1500     up        net0
vnic3     vnic     1500     up        net0

solaris$ zonecfg -z ZoneA
zonecfg:ZoneA> add net
zonecfg:ZoneA:net> set physical=vnic1
zonecfg:ZoneA:net> end
zonecfg:ZoneA> verify
zonecfg:ZoneA> commit
zonecfg:ZoneA> exit
solaris$ zoneadm -z ZoneA reboot


solaris$ zonecfg -z ZoneB
zonecfg:ZoneB> add net
zonecfg:ZoneB:net> set physical=vnic2
zonecfg:ZoneB:net> end
zonecfg:ZoneB> verify
zonecfg:ZoneB> commit
zonecfg:ZoneB> exit
solaris$ zoneadm -z ZoneB reboot


solaris$ zonecfg -z ZoneC
zonecfg:ZoneC> add net
zonecfg:ZoneC:net> set physical=vnic3
```

```
zonecfg:ZoneC:net> end
zonecfg:ZonCe> verify
zonecfg:ZoneC> commit
zonecfg:ZoneC> exit
solaris$ zoneadm -z ZoneC reboot

solaris:~$ dladm show-vnic
LINK            OVER           SPEED  MACADDRESS        MACADDRTYPE IDS
ZoneA/vnic1     net0           1000   2:8:20:47:8c:85   random      VID:101
ZoneB/vnic2     net0           1000   2:8:20:47:8c:85   random      VID:102
ZoneC/vnic3     net0           1000   2:8:20:47:8c:85   random      VID:103


solaris:~$ zlogin ZoneA
ZoneA:~$ ipadm create-ip vnic1
ZoneA:~$ ipadm create-addr -a 192.0.2.1 vnic1
vnic1/v4

solaris:~$ zlogin ZoneB
ZoneB:~$ ipadm create-ip vnic2
ZoneB:~$ ipadm create-addr -a 192.0.2.5 vnic2
vnic2/v4

solaris:~$ zlogin ZoneC
ZoneC:~$ ipadm create-ip vnic3
ZoneC:~$ ipadm create-addr -a 192.0.2.8 vnic3
vnic3/v4
```

Configuring VLANs Over a Link Aggregation

Configuring VLANs on a link aggregation is similar to configuring VLANs over an interface. Link aggregations are described in Chapter 2, “Configuring High Availability by Using Link Aggregations”.

▼ How to Configure VLANs Over a Link Aggregation

Before You Begin Create the link aggregation. See “How to Create a Link Aggregation” on page 34.

Ensure that your role has the appropriate rights profile to perform this procedure. See “Using Rights Profiles to Perform Network Configuration” on page 16.

1. **List the link aggregations that are configured on the system.**

```
$ dladm show-aggr
```

2. **For every VLAN that you want to create over the link aggregation you selected, you can use the following command:**

   ```
   $ dladm create-vlan -l link -v vid VLAN-link
   ```

   link Specifies the aggregation on which the VLAN is being created.

   vid Indicates the VLAN ID number.

   VLAN-link Specifies the name of the VLAN.

3. **For every VLAN that you created in the previous step, create an IP interface over the VLAN.**

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

   interface is the interface name, which typically follows the VLAN name.

4. **For each IP interface on a VLAN, configure a valid IP address.**

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

Example 17 Configuring Multiple VLANs Over a Link Aggregation

In this example, two VLANs are configured on a link aggregation. The VLANs are assigned VLAN IDs 193 and 194, respectively.

```
$ dladm show-link
LINK     CLASS     MTU     STATE     OVER
net0     phys      1500    up        --
net1     phys      1500    up        --
aggr0    aggr      1500    up        net0 net1

$ dladm create-vlan -l aggr0 -v 193 acctg0
$ dladm create-vlan -l aggr0 -v 194 humres0

$ ipadm create-ip acctg0
$ ipadm create-ip humres0

$ ipadm create-addr -a 192.0.2.50/24 acctg0
ipadm: acctg0/v4
$ ipadm create-addr -a 192.0.2.60/24 humres0
ipadm: humres0/v4
```

Displaying VLAN Information

The `dladm show-link` command displays information about all datalinks, whether they are used for VLANs or not. For VLAN-specific information, use `dladm show-vlan` instead.

```
$ dladm show-link
LINK     CLASS    MTU      STATE     OVER
net0     phys     1500     up        --
net1     phys     1500     up        --
net2     phys     1500     up        --
web1     vlan     1500     up        net0
auth1    vlan     1500     up        net0
app1     vlan     1500     up        net0
web2     vlan     1500     up        net1
auth2    vlan     1500     up        net1
app2     vlan     1500     up        net1
web3     vlan     1500     up        net2
auth3    vlan     1500     up        net2

$ dladm show-vlan
LINK      VID   SVID    PVLAN-TYPE   FLAGS    OVER
web1      111    --       --         ----     net0
auth1     112    --       --         ----     net0
app1      113    --       --         ----     net0
web2      111    --       --         ----     net1
auth2     112    --       --         ----     net1
app2      113    --       --         ----     net1
web3      111    --       --         ----     net2
auth3     113    --       --         ----     net2
```

EXAMPLE 18 Displaying the VLAN IDs That Can be Created on the Datalinks

In Oracle Solaris Kernel Zones, the `dladm show-phys -v` syntax shows the VLAN IDs that can be created on each physical datalink.

```
$ dladm show-phys -v
LINK    VID   INUSE  CLIENT
net0    40    yes    vnic0,vnic1
        20    no     --
        15    yes    vnic2
net1    32    no     --
        11    no     --
        10    no     --
```

EXAMPLE 19 Viewing the Allowed MAC Addresses and Allowed VLAN IDs

In Oracle Solaris Kernel Zones, you can use the `dladm show-phys -o` syntax displays the allowed MAC addresses and VLAN IDs you can use.

```
$ dladm show-phys -o link,media,device,allowed-addresses,allowed-vids
LINK   MEDIA       DEVICE   ALLOWED-ADDRESSES   ALLOWED-VIDS
net0   Ethernet    zvnet0   fa:16:3f,           100-199,
                            fa:80:20:21:22      400-498,500
```

Modifying VLANs

You can modify a VLAN by using the `dladm modify-vlan` command in the following ways:

- Change the VLAN ID of a VLAN
- Migrate a VLAN to another underlying link

Modifying the VLAN ID of a VLAN

To change the VLAN ID of a VLAN, use one of the following commands:

- `$ dladm modify-vlan -v` *vid* `-L` *datalink*

  vid specifies the new VLAN ID that you are assigning to the VLAN and *datalink* refers to the underlying link over which the VLAN is configured.

  Note - This syntax is valid if only a single VLAN exists on the datalink. Otherwise, use the second syntax.

  If you modify the VLAN ID on the link, you must also configure the switch port for the new VLAN ID.

- `$ dladm modify-vlan -v` *vid vlan*

  Use this command to change the unique VLAN IDs of multiple VLANs over a single datalink. Each VLAN on the datalink has a unique VLAN ID so you must change the VLAN IDs one at a time. In the setup shown in Figure 10, “VLANs With Zones,” on page 58, you would change the VLAN IDs of `web1`, `auth1`, and as follows:

  ```
  $ dladm modify-vlan -v 123 web1
  ```

```
$ dladm modify-vlan -v 456 app1
$ dladm modify-vlan -v 789 auth1
```

Migrating a VLAN to Another Underlying Link

You can migrate a VLAN between underlying datalinks without needing to delete and reconfigure the VLAN. The underlying link can be a physical link, a link aggregation, or an etherstub. For more information about etherstubs, see "Network Virtualization Components" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

To successfully migrate a VLAN, the target underlying datalink must be able to accommodate the datalink properties of the VLAN. Otherwise, migration fails and the user is notified.

Certain hardware-dependent properties might change after a VLAN migration. For example, a VLAN always shares the same MAC address as its underlying datalink. Therefore, when you migrate a VLAN, the VLAN's MAC address changes to the primary MAC address of the target datalink. Other properties that might be affected are the datalink state, link speed, and MTU size. However, after a successful migration, all applications continue to operate normally without interruption.

Note - A migrated VLAN does not retain any of its hardware lane statistics from the original datalink. Available hardware lanes for the VLAN on the target datalink become the new source of statistics information. However, software statistics that are displayed by default by the `dlstat` command are preserved.

You can perform a VLAN migration either globally or selectively.

Global Migration

Global migration moves all of the VLANs from one underlying datalink to another. You need to specify only the source and target datalinks.

```
$ dladm modify-vlan -l net1 -L ether0
```

`-l` Refers to the target datalink to which the VLANs are migrated.

`-L` Refers to the original datalink over which the VLANs are configured.

Note - You must specify the target datalink before the source datalink.

Selective Migration

Selective migration is used to migrate only selected VLANs. To perform selective VLAN migration, you specify the VLANs that you want to move. In the following example, which is based on Figure 10, "VLANs With Zones," on page 58, VLANs are moved from `net0` to `net3`.

```
$ dladm modify-vlan -l net3 web1,auth1,app1
```

Note - When migrating VLANs selectively, do not include the `-L` option, which applies only to global migration.

You can change the VLAN IDs of VLANs while performing a migration. Using Figure 10, "VLANs With Zones," on page 58 as the basis, the following example shows how you would migrate multiple VLANs and change their VLAN IDs at the same time.

```
$ dladm show-vlan
LINK    VID   SVID    PVLAN-TYPE   FLAGS    OVER
web1    111    --       --         -----    net0
auth1   112    --       --         -----    net0
app1    113    --       --         -----    net0

$ dladm modify-vlan -l net3 -v 123 web1
$ dladm modify-vlan -l net3 -v 456 auth1
$ dladm modify-vlan -l net3 -v 789 app1
$ dladm show-vlan
LINK    VID   SVID    PVLAN-TYPE   FLAGS    OVER
web1    123    --       --         -----    net3
auth1   456    --       --         -----    net3
app1    789    --       --         -----    net3
```

Note - A parallel command, `dladm modify-vnic`, migrates VNICs that are configured as VLANs. You must use the correct subcommand depending on the object you are modifying. See the next example. Refer also to "Modifying the VLAN IDs of VNICs" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

EXAMPLE 20 Modifying the VLAN ID of a VLAN VNIC in a Zone

You can modify the VLAN ID of a VLAN VNIC when you need to move one VLAN to another. For example, if any department that shared a VLAN with another department needs to move to its own newly configured VLAN, the VLAN ID of the VLAN VNICs need to be modified. The following example shows how to modify VLAN IDs of VLAN VNICs assigned to zones. It is based on the configuration in Example 16, "Configuring VNICs as VLANs for Zones," on page 59, where 3 zones are each configured with a VNIC as follows:

```
solaris:~$ dladm show-vnic
LINK            OVER           SPEED  MACADDRESS        MACADDRTYPE IDS
ZoneA/vnic1     net0           1000   2:8:20:47:8c:85   random      VID:101
ZoneB/vnic2     net0           1000   2:8:20:47:8c:85   random      VID:102
ZoneC/vnic3     net0           1000   2:8:20:47:8c:85   random      VID:103
```

To move the zones' VLANs, you would proceed as follows:

```
usr@solaris:~$ dladm modify-vnic -v 221 -L vnic1
usr@solaris:~$ dladm modify-vnic -v 222 -L vnic2
usr@solaris:~$ dladm modify-vnic -v 223 -L vnic3

usr@solaris:~$ dladm show-vnic
LINK            OVER           SPEED  MACADDRESS        MACADDRTYPE IDS
ZoneA/vnic1     net0           1000   2:8:20:47:8c:85   random      VID:221
ZoneB/vnic2     net0           1000   2:8:20:47:8c:85   random      VID:222
ZoneC/vnic3     net0           1000   2:8:20:47:8c:85   random      VID:223
```

Deleting a VLAN

To delete a VLAN configuration,you must first delete any existing IP configurations on the VLAN. Otherwise, the operation fails. This requirement applies whether you are using the normal `dladm delete-vlan` command or the `dladm delete-phys` command, which covers a wider scope of L2 components.

EXAMPLE 21 Deleting a VLAN Configuration

This example shows how to delete the VLAN `web1`.

```
$ dladm show-vlan
LINK      VID   SVID    PVLAN-TYPE   FLAGS    OVER
web1      111    --       --         ----     net0
auth1     112    --       --         ----     net0
app1      113    --       --         ----     net0
web2      111    --       --         ----     net1
auth2     112    --       --         ----     net1
app2      113    --       --         ----     net1
web3      111    --       --         ----     net2
auth3     113    --       --         ----     net2

$ ipadm delete-ip web1
$ dladm delete-vlan web1
```

Using VLAN Tagging for SR-IOV-Enabled Ports

When using some Intel network adapters that support SR-IOV, a virtual function might be the target of malicious behavior. Unexpected software-generated frames can throttle traffic between the host and the virtual switch, which might negatively affect performance. To drop unexpected and potentially malicious frames, configure all SR-IOV-enabled ports to use VLAN tagging. For more information, see *Oracle VM Server for SPARC 3.6 Administration Guide*.

Use Cases in of VLAN Deployments

This section contains uses cases that show different ways that you can use VLAN technology.

Combining Link Aggregation and VLAN Configurations

This sample case shows how you can combine link aggregations and VLANs for your network to provide high availability to the VLANs. At the end of the process, the system has 8 operational VLANs configured on a link aggregation.

```
$ ipadm show-if
IFNAME    CLASS      STATE   ACTIVE   OVER
lo0       loopback   ok      yes      --
net0      ip         ok      yes      --
net1      ip         ok      yes      --
net2      ip         ok      yes      --
net3      ip         ok      yes      --

$ ipadm delete-ip net0
$ ipadm delete-ip net1
$ ipadm delete-ip net2
$ ipadm delete-ip net3

$ dladm create-aggr -P L2,L3 -l net0 -l net1 -l net2 -l net3 default0

$ dladm show-link
LINK        CLASS     MTU   STATE    OVER
net0        phys      1500  up       --
net1        phys      1500  up       --
net2        phys      1500  up       --
net3        phys      1500  up       --
default0    aggr      1500  up       net0 net1 net2 net3
```

```
$ ipadm create-ip default0
$ ipadm create-addr -a 203.0.113.4/24 default0

$ dladm create-vlan -v 2 -l default0 orange0
$ dladm create-vlan -v 3 -l default0 green0
$ dladm create-vlan -v 4 -l default0 blue0
$ dladm create-vlan -v 5 -l default0 white0
$ dladm create-vlan -v 6 -l default0 yellow0
$ dladm create-vlan -v 7 -l default0 red0
$ dladm create-vlan -v 8 -l default0 cyan0

$ dladm show-link
LINK        CLASS      MTU  STATE    OVER
net0        phys      1500  up       --
net1        phys      1500  up       --
net2        phys      1500  up       --
net3        phys      1500  up       --
default0    aggr      1500  up       net0 net1 net2 net3
orange0     vlan      1500  up       default0
green0      vlan      1500  up       default0
blue0       vlan      1500  up       default0
white0      vlan      1500  up       default0
yellow0     vlan      1500  up       default0
red0        vlan      1500  up       default0
cyan0       vlan      1500  up       default0

$ dladm show-vlan

LINK      VID   SVID    PVLAN-TYPE   FLAGS    OVER
orange0    2     --        --        -----    default0
green0     3     --        --        -----    default0
blue0      4     --        --        -----    default0
white0     5     --        --        -----    default0
yellow0    6     --        --        -----    default0
red0       7     --        --        -----    default0
cyan0      8     --        --        -----    default0

$ ipadm create-ip orange0
$ ipadm create-ip green0
$ ipadm create-ip blue0
$ ipadm create-ip white0
$ ipadm create-ip yellow0
$ ipadm create-ip red0
$ ipadm create-ip cyan0

$ ipadm create-addr -a 203.0.113.5/24 orange0
$ ipadm create-addr -a 203.0.113.6/24 green0
```

```
$ ipadm create-addr -a 203.0.113.7/24 blue0
$ ipadm create-addr -a 203.0.113.8/24 white0
$ ipadm create-addr -a 203.0.113.9/24 yellow0
$ ipadm create-addr -a 203.0.113.10/24 red0
$ ipadm create-addr -a 203.0.113.11/24 cyan0
```

Configuring Flows for a VLAN

This case shows how to configure flows for a VLAN that is assigned to a non-global zone. The zone has multiple services such as production, replication, and backup running over the same VLAN link. Flows are configured to provision the link's bandwidth among these services. In this virtualization setup, a VNIC is used and configured as a VLAN.

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

solaris:~$ dladm create-vnic -v 101 -l net0 vnic1

solaris$ zonecfg -z zone1
Use 'create' to begin configuring a new zone.
zonecfg:zone1> create -t SYSsolaris
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
solaris$ zoneadm -z zone1 boot

solaris$ zlogin zone1
zone1$ ipadm create-ip vnic1
zone1:~$ ipadm create-addr -a 192.0.2.1 vnic1

zone1:~$ flowadm add-flow -a local_ip=192.0.2.1,local_port=80  \
-p max-bw=500M,priority=high -l vnic1 flow_prod
zone1:~$ flowadm add-flow -a local_ip=192.0.2.1,local_port=138 \
-p max-bw=250M,priority=medium -l vnic1 flow_repl
zone1:~$ flowadm add-flow -a local_ip=192.0.2.1,local_port=21 \
-p max-bw=250M,priority=low -l vnic1 flow_bkup

zone1:~$ flowadm
FLOW        LINK    PROTO LADDR       LPORT  RADDR   RPORT  DSFLD
flow_prod   vnic1   --    192.0.2.1   80     --      --     --
```

```
flow_repl   vnic1   --    192.0.2.1   138    --      --     --
flow_bkup   vnic1   --    192.0.2.1   21     --      --     --

zone1:~$ flowadm show-flowprop
FLOW        PROPERTY        PERM VALUE          DEFAULT        POSSIBLE
flow_prod   max-bw          rw   500            --             --
flow_prod   priority        rw   high           medium         low,medium,high
flow_prod   hwflow          r-   off            --             on,off
flow_repl   max-bw          rw   250            --             --
flow_repl   priority        rw   medium         medium         low,medium,high
flow_repl   hwflow          r-   off            --             on,off
flow_bkup   max-bw          rw   250            --             --
flow_bkup   priority        rw   low            medium         low,medium,high
flow_bkup   hwflow          r-   off            --             on,off
```

Configuring Bandwidth for VLAN VNICs

This example shows how to configure bandwidth limit on the VLAN VNICs to provision the bandwidth of the physical link among the VLAN VNICs.

Three services run in different zones that share the same physical link. One VLAN VNIC is assigned for each service in the zones. The bandwidth of the VNIC is configured to provision the bandwidth of the physical link. This ensures that the available network resources are efficiently used.

```
solaris$ dladm show-link
LINK     CLASS     MTU     STATE     OVER
net0     phys      1500    up        --
solaris$ dladm create-vnic -v 10 -l net0 vnic_prod
solaris$ dladm create-vnic -v 11 -l net0 vnic_repl
solaris$ dladm create-vnic -v 12 -l net0 vnic_bkup
solaris$ dladm show-vnic
LINK          OVER    SPEED   MACADDRESS         MACADDRTYPE   IDS
vnic_prod     net0    1000    2:8:20:35:b:9a     random        VID:10
vnic_repl     net0    1000    2:8:20:fa:94:57    random        VID:11
vnic_bkup     net0    1000    2:8:20:51:1c:4a    random        VID:12

solaris$ dladm set-linkprop -p max-bw=1G vnic_prod
solaris$ dladm set-linkprop -p max-bw=500M vnic_repl
solaris$ dladm set-linkprop -p max-bw=200M vnic_bkup

solaris$ zonecfg -z zone1
Use 'create' to begin configuring a new zone.
zonecfg:zone1> create -t SYSsolaris
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic_prod
```

```
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
solaris$ zoneadm -z zone1 boot

solaris$ zlogin zone1
zone1$ ipadm create-ip vnic_prod
zone1:~$ ipadm create-addr -a 192.0.2.1 vnic_prod
vnic_prod/v4
solaris$ zoneadm -z zone1 reboot

solaris$ zonecfg -z zone2
Use 'create' to begin configuring a new zone.
zonecfg:zone2> create -t SYSsolaris
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vnic_repl
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
solaris$ zoneadm -z zone2 boot

solaris$ zlogin zone2
zone2$ ipadm create-ip vnic_repl
zone2:~$ ipadm create-addr -a 192.0.2.1 vnic_repl/v4

solaris$ zonecfg -z zone3
Use 'create' to begin configuring a new zone.
zonecfg:zone3> create -t SYSsolaris
zonecfg:zone3> add net
zonecfg:zone3:net> set physical=vnic_bkup
zonecfg:zone3:net> end
zonecfg:zone3> verify
zonecfg:zone3> commit
zonecfg:zone3> exit
solaris$ zoneadm -z zone3 boot

solaris$ zlogin zone3
zone3$ ipadm create-ip vnic_bkup
zone3:~$ ipadm create-addr -a 192.0.2.1 vnic_bkup/v4
```

♦ ♦ ♦ CHAPTER 5

Configuring Private Virtual Local Area Networks

This chapter discusses private VLANs (PVLANs) and describes the procedures to configure and modify PVLANs.

This chapter contains the following topics:

- "Overview of Private VLANs"
- "Configuring a Private VLAN"
- "Modifying Private VLANs"
- "Deleting a VLAN"
- "Assigning a PVLAN to a Zone"

Note - To administer private VLANs and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 16.

Overview of Private VLANs

Private VLAN (PVLAN) technology defined in RFC 5517 enables you to divide a VLAN into sub-VLANs to isolate network traffic. A regular VLAN is a single broadcast domain. When you configure a PVLAN, the single broadcast domain is partitioned into smaller subdomains. The standard (IEEE 802.1Q) VLAN is called the *primary VLAN* and the sub-VLANs are called the *secondary VLANs*. The secondary VLANs can be either isolated VLANs or community VLANs.

- **Isolated VLAN**

  The ports that are associated with an isolated VLAN can communicate only with the primary VLAN and not with any other secondary VLAN. You can create only one isolated VLAN within a primary VLAN domain.

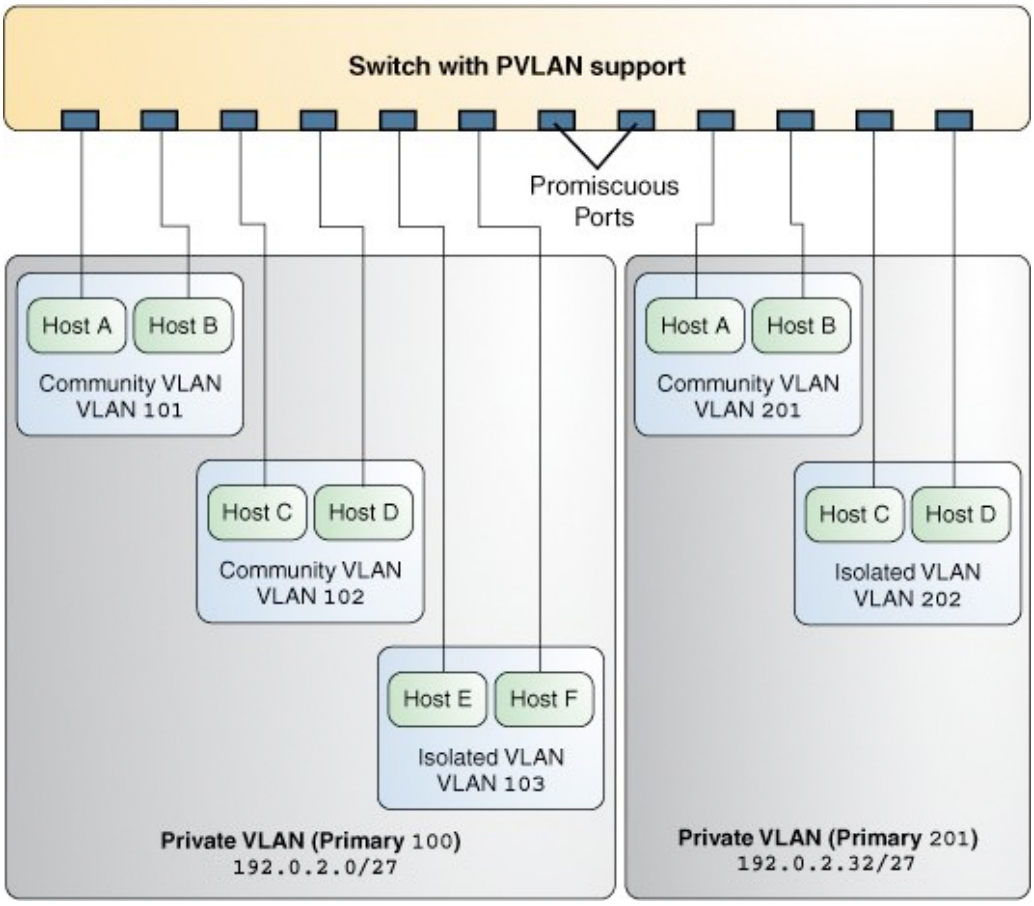
- **Community VLAN**

  The ports associated with community VLAN can communicate with the primary VLAN and the other ports that are in the same community VLAN. You can create multiple community VLANs within a primary VLAN domain.

In Oracle VM Server for SPARC that has a configured PVLAN, the inter-vnet-links feature supports the communication restrictions of isolated and community PVLANs. Inter-vnet-links are disabled for isolated PVLANs and are enabled only for virtual networks that are in the same community for community PVLANs. Direct traffic from other virtual networks outside of the community is not permitted. For more information, see "Using Private VLANs" in *Oracle VM Server for SPARC 3.6 Administration Guide*.

The following figure shows a simple PVLAN configuration with two PVLANs configured on a switch that supports PVLAN.

FIGURE 11 Private VLAN



In the figure, the private VLAN with the primary VLAN ID `100` has three secondary VLANs with secondary VLAN IDs `101`, `102`, and `103`. Two of the secondary VLANs are community VLANs with the hosts Host A and Host B and Host C and Host D, and the other is an isolated VLAN with the hosts Host E and Host F.

The ports associated with the community VLANs `101` and `102` can communicate with the primary VLAN `100` and also can communicate with each other, that is, Host A can communicate with Host B and Host C can communicate with Host D. However, Host A and Host B in community VLAN `101` cannot communicate with Host C and Host D in community

VLAN `102`. The ports associated with the isolated VLAN can communicate only with the primary VLAN and not with each other.

For more information, see “Configuring a Private VLAN” on page 79.

PVLAN Ports

PVLANs can span multiple switches similar to regular VLANs. A trunk port carries frames either from a primary VLAN or a secondary VLAN. The two different types of ports associated with the PVLAN are promiscuous trunk ports and secondary trunk ports.

- **Promiscuous trunk port** – All the frames going out from the promiscuous trunk port are tagged with the primary VLAN ID. This port is configured on the top-level switch uplink port. The switch maps the primary VLAN ID and the secondary VLAN ID.
- **PVLAN secondary trunk port** – All the frames going out from PVLAN secondary trunk port are tagged with the secondary VLAN ID.

Note - Even though the PVLAN ports are isolated at layer 2, the ports can still communicate with each other at layer 3 as long as the external policy gateway allows the communication.

Tagging the Outgoing Traffic

In Oracle Solaris, you must set the tag mode property for a PVLAN depending on whether the promiscuous trunk port is on the system or the switch. Thus, you need to identify whether the PVLAN is configured on the switch or only on the system.

To send out the packets with the primary VLAN ID, set the `tag-mode` property to `primary`. To send out packets with the secondary VLAN ID, set the property to `secondary`. The switch converts the secondary VLAN ID to the primary VLAN ID. By default, the property is set to `primary`. Thus, the packets are sent out with the primary VLAN ID.

The `secondary` setting is also used if you want a PVLAN to span existing multiple PVLAN switches.

EXAMPLE 22 Setting the Tag Mode for a PVLAN

The following example shows how you can change the tag mode on a datalink.

```
$ dladm set-linkprop -p pvlan-tag-mode=secondary net0
$ dladm show-linkprop -p pvlan-tag-mode net0
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
net0     pvlan-tag-mode  rw   secondary    secondary    primary   secondary,
                                                                  primary
```

Note - The value of `tag-mode` depends on whether the switch supports PVLANs.

PVLAN Configuration Requirements

When configuring a PVLAN, note the following restrictions:

- The primary VLAN ID and secondary VLAN ID for community VLANs must be unique.

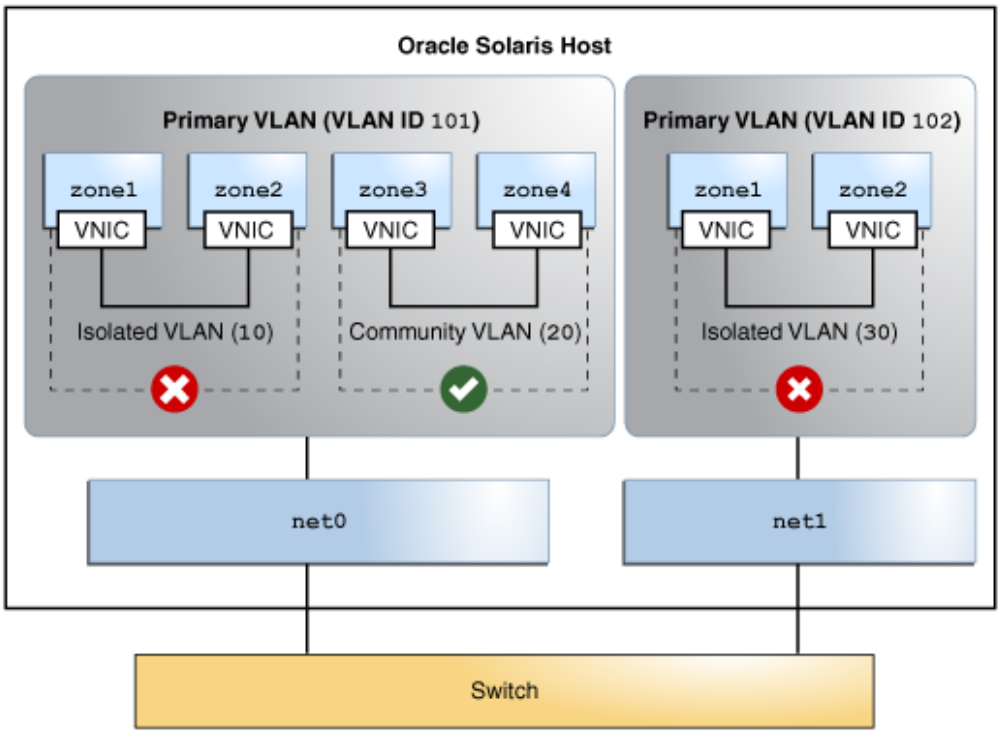
  For example, if you have a community VLAN with a primary VLAN ID of `3` and a secondary VLAN ID of `100`, you cannot create another community VLAN that uses either `3` or `100` as the secondary VLAN ID. That is, any combination containing the ID pair such as `4`, `100` or `4`, `3` is invalid.
- You can reuse the secondary VLAN ID of an isolated VLAN.

  For example, if you have an isolated VLAN with a primary VLAN ID of `3` and a secondary VLAN ID of `100`, you can reuse the VLAN ID `100` as the primary VLAN ID or the community secondary VLAN ID.

PVLANs With Zones

You can configure multiple private virtual networks within a single network unit such as a switch by combining VLANs and Oracle Solaris zones. With PVLAN you can provide network isolation between zones which are in the same VLAN without introducing any additional VLANs. The following figure shows a system with two physical NICs and two configured PVLANs, `101` and `102`.

FIGURE 12 PVLAN With Zones



An isolated and a community VLAN are configured on Primary VLAN `101`. Only one isolated VLAN is configured on primary VLAN `102`. On the Primary VLAN `101`, the zones in the isolated VLAN and the community VLAN cannot communicate with each other. However, the zones within the community VLAN can communicate with each other.

For information about how to assign PVLAN to a zone, see “Assigning a PVLAN to a Zone” on page 80.

Configuring a Private VLAN

To create a PVLAN, use the `dladm create-vlan` command. To configure a PVLAN VNIC to host the PVLAN traffic, use the `dladm create-vnic` command. For information about configuring a PVLAN VNIC, see "How to Configure VNICs as PVLANs" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

To create a PVLAN, use the following syntax:

```
dladm create-vlan [ -l link
   -v VLAN-ID[,PVLAN-SVID[,PVLAN-type]]
   [VLAN-link]
```

link — Specifies the Ethernet link over which the VLAN is created.

VLAN-ID — Primary ID associated with a VLAN.

PVLAN-SVID — PVLAN Secondary VLAN ID associated with the VLAN.

PVLAN-type — PVLAN type associated with the VLAN, which can be either `isolated` or `community`. The default value is `isolated`.

VLAN-link — Name of the VLAN.

EXAMPLE 23 Creating an Isolated PVLAN by Using the `dladm create-vlan` Command

The following example shows how you can create a PVLAN with a primary VLAN ID of `3`, a secondary VLAN ID of `100`, and the PVLAN type set to `isolated`.

```
$ dladm create-vlan -v 3,100,isolated -l net0 vlan1
$ dladm show-vlan
LINK     VID   SVID    PVLAN-TYPE   FLAGS    OVER
vlan1    3     100     isolated     -----    net0
```

Modifying Private VLANs

To modify the primary VLAN ID, secondary VLAN ID, and the PVLAN type of PVLANs, use the `dladm modify-vlan` command. You can use this command only if you created the PVLAN with the `dladm create-vlan` command and not the `dladm create-vnic` command. For information about modifying PVLAN VNICs, see "Modifying PVLAN VNICs" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

You modify the primary VLAN ID, the secondary VLAN ID, and the PVLAN type tuple together. Use the following syntax:

```
dladm modify-vlan [-v VID[,PVLAN-SVID[,PVLAN-type]]
      VLAN-link
```

For example:

```
$ dladm show-vlan
LINK      VID   SVID    PVLAN-TYPE   FLAGS    OVER
vlan1     10    102     isolated     ----     net0
$ dladm modify-vlan -v 15,103,community vlan1
$ dladm show-vlan
LINK      VID   SVID    PVLAN-TYPE   FLAGS    OVER
vlan1     15    103     community    ----     net0
```

Deleting a Private VLAN

To delete PVLAN configurations, first delete any existing IP configurations on the PVLAN before you can delete the PVLAN. Otherwise, the operation fails. Then, use the `dladm delete-vlan` command.

EXAMPLE 24 Deleting a VLAN Configuration

This example shows how to delete a VLAN configuration.

```
$ dladm show-vlan
LINK      VID   SVID    PVLAN-TYPE   FLAGS    OVER
vlan1     15    103     community    ----     net0
vlan2     20    105     isolated     ----     net1
$ ipadm delete-ip vlan1
$ dladm delete-vlan vlan1
$ dladm show-vlan
LINK      VID   SVID    PVLAN-TYPE   FLAGS    OVER
vlan2     20    105     isolated     ----     net1
```

Assigning a PVLAN to a Zone

For better isolation, you can assign a PVLAN to a zone, which you can achieve in two ways:

- Create a PVLAN in the global zone and then assign it to a non-global zone.

- Configure a PVLAN during zone configuration.

The two examples after the following procedure show you how to do both methods.

▼ How to Create a PVLAN and Assign 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 16.

1. **Create a PVLAN.**

```
$ dladm create-vlan -l link -v VLAN-ID[,PVLAN-SVID[,PVLAN-type]] [VLAN-link]
```

2. **Verify the VLAN that you created.**

```
$ dladm show-vlan
```

3. **Configure the zone and assign the PVLAN that you created.**

```
global$ zonecfg -z zone-name
zonecfg:zone-name> add net
zonecfg:zone-name:net> set physical=VLAN-ID
zonecfg:zone-name:net> end
zonecfg:zone-name> verify
zonecfg:zone-name> commit
zonecfg:zone-name> exit
global$ zoneadm -z zone-name reboot
```

Example 25 Creating and Assigning a PVLAN to a Zone

In this example, the zone already exists and the creation and assigning of the PVLAN occurs later.

```
$ dladm create-vlan -l net0 -v 110,120,community vlan110
$ dladm show-vlan
LINK         VID     SVID    PVLAN-TYPE  FLAGS  OVER
vlan110      110     120     community   -----  net0

global$ zonecfg -z zone2
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vlan110
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
```

```
global$ zoneadm -z zone2 reboot

$ dladm show-vlan
LINK            VID     SVID    PVLAN-TYPE  FLAGS  OVER
vlan110         110     120     community   -----  net0
zone2/vlan110   110     120     community   -----  net0
```

The PVLAN that is created is assigned to the zone.

Example 26 Creating a PVLAN While You Configure a Zone

In this example, the PVLAN is assigned as part of the configuration of a zone.

```
global$ zonecfg -z zone2
zonecfg:zone2> add anet
zonecfg:zone2:anet> set vlan-id=100,200,community
zonecfg:zone2:anet> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global$ zoneadm -z zone2 reboot
```

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

♦ ♦ ♦ CHAPTER 6

Administering Bridging Features

You can use bridging to connect separate network segments so that they communicate as if they were a single network segment. This chapter describes how to configure and administer bridged networks.

This chapter contains the following topics:

- "Overview of Bridged Networks"
- "Creating a Bridge"
- "Modifying the Protection Type for a Bridge"
- "Adding Links to an Existing Bridge"
- "Removing Links From a Bridge"
- "Displaying Bridge Configuration Information"
- "Deleting a Bridge From the System"
- "Administering VLANs on Bridged Networks"
- "Debugging Bridges"

Note - To administer bridges and issue commands described in this chapter, you must have the appropriate rights profile. See "Using Rights Profiles to Perform Network Configuration" on page 16.

Overview of Bridged Networks

Bridges connect various nodes in the network into a single network. When connected by a bridge, the attached network segments communicate as if they were a single network segment. The network segments share a single broadcast network and communicate as if they were a single network segment when connected. Thus, nodes communicate by using network protocols such as IP rather than routers to forward traffic across network segments. Bridges use a packet-forwarding mechanism to connect subnetworks together and enable a system to transmit packets

to their destinations by using the shortest connection routes. If you do not use a bridge, you must configure IP routing to permit the forwarding of IP traffic between nodes.

Although you can use both bridging and routing to distribute information about the locations of resources on the network, they differ in several ways. Routing is implemented at the IP layer (L3) and uses routing protocols. No routing protocols are used on the datalink layer.

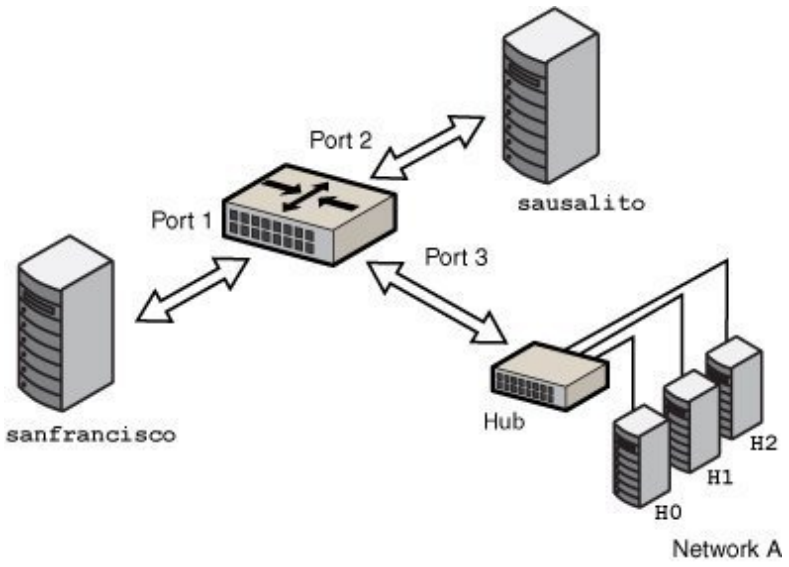
Bridging is used to distribute information about the locations of resources on the network. In a bridged network, the destinations of forwarded packets are determined by examining the network traffic that is received on the links that are attached to the bridge. A bridged network uses protocols, such as Spanning Tree Protocol (STP) and Transparent Interconnection of Lots of Links (TRILL). For more information, see “Bridging Protocols” on page 88.

Caution - On SPARC based systems that use bridging, do not set the `local-mac-address?` property to `false`. Otherwise, these systems would incorrectly use the same MAC address on multiple ports and on the same network.

Simple Bridged Network

The following figure shows a simple bridged network configuration.

FIGURE 13 Simple Bridged Network

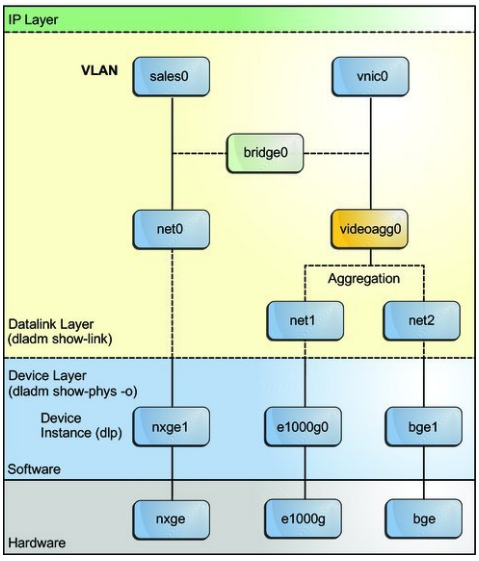


The bridge, `goldengate`, is an Oracle Solaris system that has bridging configured. The systems `sanfrancisco` and `sausalito` are physically connected to the bridge. Network A uses a hub that is physically connected to the bridge on one side and to three computer systems on the other side. The bridge ports are the links `net0`, `net1`, and `net2`.

How Oracle Solaris Bridges Are Implemented in the Network Stack

In Oracle Solaris, you can configure bridges on the datalink layer of the same network stack implementation, as shown in the following figure.

FIGURE 14 Bridges in the Network Stack for Oracle Solaris

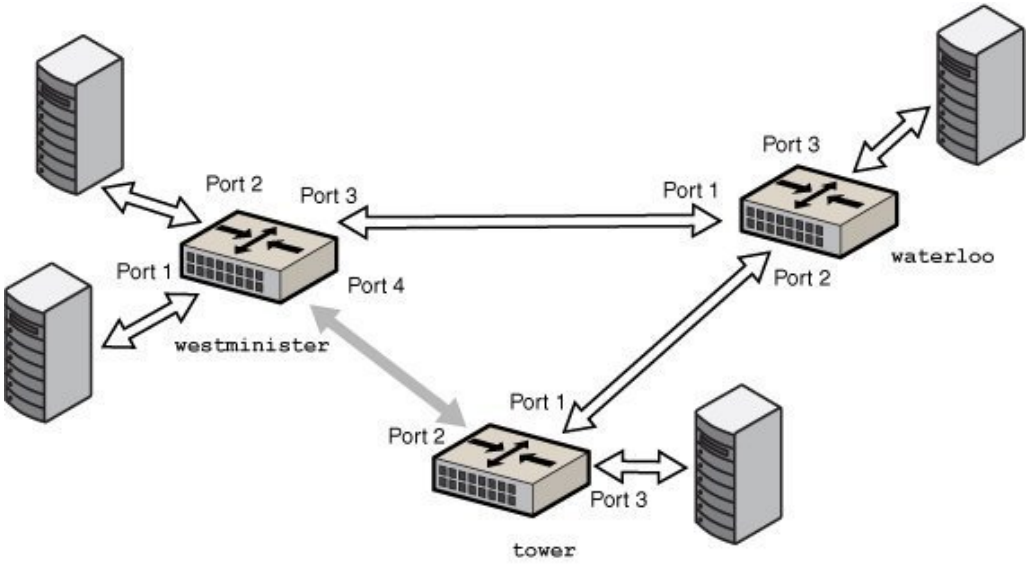


Two interfaces, `net0` and `videoagg0`, are configured as a bridge, `bridge0`. Packets that are received on one interface are forwarded to the other interface. After the bridge configuration, both interfaces can still be used to configure VLANs and IP interfaces.

Bridged Network Ring

Bridged networks can be formed into rings that physically connect several bridges together. The following figure shows a bridged network ring configuration.

FIGURE 15 Bridged Network Ring



The figure shows a bridged network that is configured in a ring. The configuration shows three bridges. Two systems are physically connected to the `westminister` bridge. One system is physically connected to the `waterloo` bridge and one system is physically connected to the `tower` bridge. The bridges are physically connected to each other through the bridge ports.

This type of configuration can cause problems with old packets saturating the network links by endlessly looping around the ring. To protect against such looping conditions, Oracle Solaris bridges implement both the STP and TRILL protocols. Note that most hardware bridges also implement STP loop protection.

How a Bridged Network Works

When a packet is received by the bridge, its source address is examined. The source address of the packet associates the node from which the packet was sent with the link on which it is received. Thereafter, when a received packet uses that same address as the destination address, the bridge forwards the packet over the link to that address.

The link that is associated with a source address might be an intermediate link that is connected to another bridge in the bridged network. Over time, all of the bridges within the bridged network "learn" which of the links sends a packet toward a given node. Therefore, destination address of the packet is used to direct the packet to its final destination by means of hop-by-hop bridging.

A local "link-down" notification indicates that all nodes on a given link are no longer reachable. In this situation, packet forwarding to the link is halted and all forwarding entries over the link are flushed. Older forwarding entries are also flushed over time. When a link is restored, packets that are received over the link are treated as new. The learning process begins again, based on the source address of a packet. This process enables the bridge to properly forward packets over that link when the address is used as the destination address.

Bridging Protocols

Bridged networks use the following protocols:

- Spanning Tree Protocol (STP)

  STP is the default protocol that is used by the bridged networks. Bridging uses the STP mechanism to prevent network loops that potentially render the subnetworks unusable. To forward packets to their destinations, bridges must listen in promiscuous mode on every link that is attached to the bridge. Listening in promiscuous mode causes bridges to become vulnerable to the occurrences of forwarding loops, in which packets infinitely circle at full-line rate.

Caution - Do not configure a link into a bridge when the highest possible levels of performance are required. Bridging requires the underlying interfaces to be in promiscuous mode, which disables a number of important optimizations that are in the hardware (NIC), driver, and other layers of the system. The disabling of these performance enhancements is an unavoidable consequence of the bridging mechanism.

These performance issues only affect links that are configured to be part of a bridge. You can use a bridge on a system where some of the links of the system are not bridged and are hence not subject to those constraints.

- Transparent Interconnection of Lots of Links (TRILL)

  Oracle Solaris supports the TRILL protection enhancement, which avoids loops without disabling links. TRILL helps to load-balance the traffic between several paths to the destination.

When STP is used for loop protection, the physical loop is mitigated by preventing one of the connections in the loop from forwarding packets. Figure 15, “Bridged Network Ring,” on page 87 shows that the physical link between the `westminster` and `tower` bridges is not used to forward packets.

Unlike STP, TRILL does not shut down physical links to prevent loops. Instead, TRILL computes the shortest-path information for each TRILL node in the network and uses that information to forward packets to individual destinations.

You can use TRILL by specifying the `-P trill` option in the `dladm create-bridge` or `dladm modify-bridge` commands. For more information, see “Creating a Bridge” on page 90 and “Modifying the Protection Type for a Bridge” on page 92.

For information about STP, see IEEE 802.1D-1998. For information about TRILL, see the Internet Engineering Task Force (IETF) TRILL draft documents (`http://tools.ietf.org/wg/trill`).

STP Daemon

Each bridge that you create by using the `dladm create-bridge` command is represented as an identically named Service Management Facility (SMF) instance of `svc:/network/bridge`. Each instance runs a copy of the `/usr/lib/bridged` daemon, which implements the STP.

For example, the following command creates a bridge called `pontevecchio`:

```
$ dladm create-bridge pontevecchio
```

The system creates an SMF service instance called `svc:/network/bridge:pontevecchio` and an observability node called `/dev/net/pontevecchio0`. The observability node is intended for use with the `snoop` command and the `wireshark` packet analyzer. You can use the `dlstat` command to obtain the run time statistics of the bridge.

For safety purposes, all ports run standard STP by default. A bridge that does not run some form of bridging protocol, such as STP, can form long-lasting forwarding loops in the network. Because Ethernet has no hop-count or time-to-live (TTL) on packets, any such loops are fatal to the network.

When a particular port is not connected to another bridge (for example, because the port has a direct point-to-point connection to a host system), you can administratively disable STP for that port. Even if all ports on a bridge have STP disabled, the STP daemon still runs for the following reasons:

- To handle any new ports that are added
- To implement BPDU guarding

- To enable or disable forwarding on the ports, as necessary

When a port has STP disabled, the `bridged` daemon continues to listen for BPDUs (BPDU guarding). The daemon uses `syslog` to flag any errors and disables forwarding on the port to indicate a serious network misconfiguration. The link is re-enabled when the link goes down and comes up again, or when you manually remove the link and add it again.

If you disable the SMF service instance for a bridge, the bridge stops on those ports as the STP daemon is stopped. If the instance is restarted, STP starts from its initial state.

TRILL Daemon

Each bridge that you create by using the `dladm create-bridge -P trill` command is represented by an identically named SMF instance of `svc:/network/bridge` and `svc:/network/routing/trill`. Each instance of `svc:/network/routing/trill` runs a copy of the `/usr/lib/trilld` daemon, which implements the TRILL protocol.

For example, the following command creates a bridge called `bridgeofsighs`:

```
$ dladm create-bridge -P trill bridgeofsighs
```

The system creates two SMF services called `svc:/network/bridge:bridgeofsighs` and `svc:/network/routing/trill:bridgeofsighs`. In addition, the system creates an observability node called `/dev/net/bridgeofsighs0`.

Creating a Bridge

In Oracle Solaris, use the `dladm` command and the SMF feature to administer bridges. The SMF commands enable, disable, and monitor bridge instances by using the fault-managed resource identifier (FMRI) of the instance, `svc:/network/bridge`. The `dladm` command creates or destroys bridges, and assigns links to bridges or removes those links. The links that are assigned to the bridge must be an Ethernet type, which includes 802.3 and 802.11 media.

To create a bridge between links, you must create at least one bridge instance. Each bridge instance is separate. Bridges do not include a forwarding connection between them, and a link is a member of a maximum of one bridge.

To create a bridge, use the following command:

```
$ dladm create-bridge [-P protect] [-p priority] \
    [-d forward-delay] [-l link...] \
```

bridge-name

-P *protect*
: Specifies the protection method. It can be set to one of the following values.
  - `stp` – STP protection method (the default)
  - `trill` – TRILL protection method

-p *priority*
: Specifies an IEEE STP priority value for a bridge to determine the root bridge node in the network. The default value is `32768`. Valid values are from `0` (highest priority) to `61440` (lowest priority), in increments of 4096.

-d *forward-delay*
: Specifies the STP forward delay parameter for the bridge. When the bridge that is created is the root node, all the bridges in the network use this timer to sequence the link states when a port is enabled. The default value is `15` seconds. Valid values are from `4` to `30` seconds.

-l *link*
: Adds a link to the bridge. If any of the specified links cannot be added, the command fails and the bridge is not created.

bridge-name is an arbitrary string that must be a legal SMF service instance name. This name is an FMRI component that has no escape sequences, which means that white space, ASCII control characters, and the following characters cannot be present:

```
; / ? : @ & = + $ , % < > # "
```

The name `default` and all names beginning with the `SUNW` string are reserved. Names that have trailing digits are reserved for the creation of observability devices, which are used for debugging. Because of the use of observability devices, the names of legal bridge instances are further constrained to be a legal `dlpi` name. The name must begin and end with an alphabetic character or an underscore character. The rest of the name can contain alphanumeric and underscore characters.

For more information about bridge creation options, see the description of the `dladm create-bridge` command in the `dladm`(8) man page.

EXAMPLE 27 Creating a Bridge

The following example shows how to create the `brooklyn` bridge by connecting the `net0` and `net1` links.

```
$ dladm create-bridge -P stp -d 12 -l net0 -l net1 brooklyn
$ dladm show-bridge
BRIDGE       PROTECT ADDRESS           PRIORITY DESROOT
```

```
goldengate    stp     32768/8:0:20:bf:f 32768    8192/0:d0:0:76:14:38
brooklyn      stp     32768/8:0:20:e5:8 32768    8192/0:d0:0:76:14:38
```

Modifying the Protection Type for a Bridge

STP is a mechanism to prevent network loops that could render the subnetworks unusable. In addition to using STP for bridges, Oracle Solaris supports the TRILL protection enhancement. STP is used by default, but you can use TRILL by specifying the `-P trill` option for the bridging commands.

To modify the protection type from STP to TRILL or from TRILL to STP, use the following command:

`$` **`dladm modify-bridge -P`** *protection-type bridge-name*

The `-P` *protection-type* option specifies which protection type to use, either `stp` (the default) or `trill`.

EXAMPLE 28 Modifying the Protection Type for a Bridge

The following example shows how to modify the protection type for the `brooklyn` bridge from the default STP to TRILL.

```
$ dladm modify-bridge -P trill brooklyn
```

Adding Links to an Existing Bridge

A link can be a member of at most one bridge. So, if you want to move a link from one bridge instance to another bridge instance, you must first remove the link from the current bridge before adding it to another bridge.

Links that are assigned to the same bridge must have the same MTU value. Although you can change the MTU value on an existing link, the bridge instance goes into maintenance state until you remove or change the assigned links so that the MTU values match before you restart the bridge.

With the support for FMA datalink alert, an FMA alert is raised whenever, an MTU mismatch is detected between the two datalinks. For more information, see "How Datalink MTU Mismatch Errors Are Detected" in *Troubleshooting Network Administration Issues in Oracle Solaris 11.4*.

Note - The links that are assigned to a bridge cannot be VLANs, VNICs, or tunnels. Only links that are acceptable as part of an aggregation or links that are aggregations can be assigned to a bridge.

To add a new link to an existing bridge, use the following command:

```
$ dladm add-bridge -l new-link  bridge-name
```

For example:

```
$ dladm add-bridge -l net2 rialto
```

Removing Links From a Bridge

Before you delete any bridge, all of its links must first be removed. To remove links, use the following command:

```
$ dladm remove-bridge [-l link]... bridge-name
```

For example:

```
$ dladm remove-bridge -l net0 -l net1 -l net2 charles
```

Setting Link Properties for a Bridge

You can set the following link properties for bridges:

`default-tag`	The default VLAN ID for untagged packets that are sent to a link and received from a link. The valid values are `0` to `4094`. The default value is `1`.
`forward`	Enables and disables traffic forwarding through the bridge. This property exists on all links except VNIC links. The valid values are `1` (true) and `0` (false). The default value is `1`.
`stp`	Enables and disables STP and RSTP. Valid values are `1` (true) and `0` (false). The default value is `1`, which enables STP and RSTP.
`stp-cost`	Represents the STP and RSTP cost values for using the link. Valid values are from `1` to `65535`. The default value is `0`, which is used to signal that cost is automatically computed by link type.

`stp-edge`	Specifies whether the port is connected to other bridges. Valid values are `1` (true) and `0` (false). The default value is `1`.
`stp-p2p`	Specifies the connection mode type. Valid values are `true`, `false`, and `auto`. The default value is `auto`.
`stp-priority`	Sets the STP and RSTP port priority value. Valid values are from `0` to `255`. The default value is `128`.

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

To modify the link properties of a bridge, use the following command:

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

EXAMPLE 29 Setting Link Properties for a Bridge

The following example shows how to disable traffic forwarding and set the connection mode type. To set the properties for the bridge, you must set the properties on the links that connect the bridge.

```
$ dladm create-bridge -P stp -d 12 -l net0 -l net1 brooklyn
$ dladm set-linkprop -p forward=0 net0
$ dladm set-linkprop -p stp-p2p=true net1
```

The following example shows how to reset multiple properties of a bridge.

```
$ dladm reset-linkprop -p default-tag,stp-priority brooklyn
```

Displaying Bridge Configuration Information

You can display bridge configuration information by using the `dladm show-bridge` command.

Displaying Information About Configured Bridges

You can use the `dladm show-bridge` and `dlstat show-bridge` commands to show different kinds of information about configured bridges.

Use the following command options:

- To view the list of bridges:

  ```
  $ dladm show-bridge
  ```

```
$ dladm show-bridge
BRIDGE       PROTECT ADDRESS           PRIORITY DESROOT
goldengate   stp     32768/8:0:20:bf:f 32768    8192/0:d0:0:76:14:38
baybridge    stp     32768/8:0:20:e5:8 32768    8192/0:d0:0:76:14:38
```

- To show the link-related status of a bridge:

  ```
  $ dladm show-bridge -l bridge-name
  ```

- To show link-related statistics for the bridge:

  ```
  $ dlstat show-bridge bridge-name
  ```

- To show kernel forwarding entries for the bridge:

  ```
  $ dladm show-bridge -f bridge-name
  ```

- To show TRILL information about the bridge:

  ```
  $ dladm show-bridge -t bridge-name
  ```

- To show statistics of all the bridges and their connected links:

  ```
  $ 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        --
  ```

For more information about the `dladm show-bridge` command options, see the dladm(8) man page and for information about the `dlstat show-bridge` command options, see the dlstat(8) man page.

EXAMPLE 30 Displaying Bridge Information

The following examples show how to use the `dladm show-bridge` command with various options.

- The following command shows link-related status information for a single bridge instance, `tower`. To view configured properties, use the `dladm show-linkprop` command.

  ```
  $ dladm show-bridge -l tower
  LINK         STATE        UPTIME   DESROOT
  net0         forwarding   117      8192/0:d0:0:76:14:38
  ```

```
net1         forwarding   117      8192/0:d0:0:76:14:38
```

- The following command shows the kernel forwarding entries for the specified bridge, `avignon`:

```
$ dladm show-bridge -f avignon
DEST              AGE     FLAGS  OUTPUT
8:0:20:bc:a7:dc   10.860  --     net0
8:0:20:bf:f9:69   --      L      net0
8:0:20:c0:20:26   17.420  --     net0
8:0:20:e5:86:11   --      L      net1
```

- The following command shows the TRILL information about the specified bridge, `key`:

```
$ dladm show-bridge -t key
NICK   FLAGS LINK        NEXTHOP
38628  --    london      56:db:46:be:b9:62
58753  L     --          --
```

Displaying Configuration Information About Bridge Links

You use the `dladm show-link` command with the `-o all` option to display the `BRIDGE` field in the output. If a link is a member of a bridge, this field identifies the name of the bridge of which it is a member. For links that are not part of a bridge, the field is blank if the `-p` option is used. Otherwise, the field shows `--`.

The observability node of the bridge also appears in the `dladm show-link` output as a separate link. For this node, the existing `OVER` field lists the links that are members of the bridge.

Use the following command to view configuration information about any link that is a member of a bridge.

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

The `-p` option produces output in a parsable format.

Deleting a Bridge From the System

Before you delete a bridge, you must first remove any links attached to the bridge.

▼ How to Delete a Bridge From the System

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 16.

1. **Remove any links attached to the bridge.**

   `$ dladm remove-bridge [-l` *link*`]...` *bridge-name*

2. **Delete the bridge from the system.**

   `$ dladm delete-bridge` *bridge-name*

Example 31 Deleting a Bridge From the System

The following example shows how to remove the `net0`, `net1`, and `net2` links from the `coronado` bridge, and then delete the bridge itself from the system.

```
$ dladm remove-bridge -l net0 -l net1 -l net2 coronado
$ dladm delete-bridge coronado
```

Administering VLANs on Bridged Networks

By default, VLANs that are configured on the system forward packets among all the ports on a bridge instance. When you invoke the `dladm create-vlan` or `dladm create-vnic -v` command and the underlying link is a part of a bridge, the command also enables packet forwarding of the specified VLAN on that bridge link. For more information about VLANs, see Chapter 4, "Configuring Virtual Networks by Using Virtual Local Area Networks".

To configure a VLAN on a link and disable packet forwarding to or from other links on the bridge, you must disable forwarding by setting the `forward` property for the VLAN with the `dladm set-linkprop` command. For more information, see "Setting Link Properties for a Bridge" on page 93.

▼ How to Configure VLANs Over a Datalink That Is Part of a Bridge

Before You Begin This procedure assumes that the bridge already exists. For information about how to create a bridge, see "Creating a Bridge" on page 90.

Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 16.

1. **List the link-related information of the bridge to determine the links that are a part of the bridge.**

   `$ dladm show-bridge -l` *bridge-name*

2. **Create the VLAN over the link that is a part of the bridge.**

   `$ dladm create-vlan -l` *link* `-v` *vid VLAN-link*

   link — Specifies the link on which the VLAN interface is being created.

   Note - In this procedure, the link should part of the bridge that you have created.

   vid — Indicates the VLAN ID number.

   VLAN-link — Specifies the name of the VLAN.

3. **Repeat this command for every VLAN you want to create. For every VLAN that you created, create an IP interface over the VLAN.**

   `$ ipadm create-ip` *interface*

   where *interface* is the VLAN name.

4. **For each IP interface on a VLAN, configure a valid IP address.**

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

VLANs and the STP and TRILL Protocols

VLANs are ignored by the standards-compliant STP. The bridging protocol computes just one loop-free topology by using tag-free BPDU messages, and uses this tree topology to enable and disable links. You must configure any duplicate links that are provisioned in your networks such that when those links are automatically disabled by STP, the configured VLANs are not disconnected. You must either run all VLANs everywhere on your bridged backbone or carefully examine all redundant links.

The TRILL protocol does not follow the complex STP rules. Instead, TRILL automatically encapsulates packets that have the VLAN tag intact and passes them through the network.

Debugging Bridges

Each bridge instance is assigned an *observability node*, which appears in the `/dev/net/` directory and is named with the bridge name plus a trailing `0`, for example, `/dev/net/bridgeofsighs0`.

The observability node is intended for use with the `snoop` command and the `wireshark` packet analyzer. This node operates like a standard Ethernet interface except for the transmission of packets, which are silently dropped. You cannot plumb IP on top of an observability node, and you cannot perform bind requests (`DL_BIND_REQ`) unless you use the `passive` option, which enables you only to receive packets and not to send them.

The observability node makes a single unmodified copy of every packet handled by the bridge. It is available to the user for monitoring and debugging. This behavior is similar to monitoring a port on a traditional bridge and is subject to the usual datalink provider interface (DLPI) promiscuous mode rules. You can also use the `pfmod` command or features in the `snoop` command and the `wireshark` packet analyzer to filter packets based on the VLAN ID.

The delivered packets, which are the packets that are sent to the observability node, represent the data received by the bridge.

Note - When the bridging process adds, removes, or modifies a VLAN tag, the data shown by the `snoop` command and the `wireshark` packet analyzer describes the state prior to the processes taking place. This rare situation might be confusing if distinct `default-tag` values are used on different links.

To see the packets that are transmitted and received on a particular link after the bridging process is complete, run the `snoop` command on the individual links rather than on the bridge observability node.

You can also obtain statistics about how network packets use network resources on links by using the `dlstat` command. For information, see Chapter 8, “Monitoring Network Traffic and Resource Usage” in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

♦ ♦ ♦ CHAPTER 7

Exchanging Network Connectivity Information With Link Layer Discovery Protocol

This chapter describes how to enable systems to exchange system and network connectivity information throughout the local network by using the Link Layer Discovery Protocol (LLDP).

This chapter contains the following topics:

- "Overview of LLDP"
- "Information the LLDP Agent Advertises"
- "Enabling LLDP on the System"
- "Specifying TLV Units and Values for the LLDP Packet of an Agent"
- "Disabling LLDP"
- "Monitoring LLDP Agents"

Note - To configure LLDP and issue commands described in this chapter, you must have the appropriate rights profiles. See "Using Rights Profiles to Perform Network Configuration" on page 16.

Overview of LLDP

Link Layer Discovery Protocol (LLDP) enables systems in the network to exchange connectivity and management information for the purpose of topology discovery. The information can include system capabilities, management addresses, and other information relevant to network operations. The network diagnostics service uses LLDP to detect problems that might lead to limited or degraded network connectivity.

On any LAN, individual components such as systems and switches are not configured in isolation. To host network traffic efficiently, the configuration of systems on the network must be coordinated with each other.

When you manually configure each system, switch, and other components, ensuring compatibility among the components is a challenge. The manual configuration of systems is risky and can easily cause misconfigurations, particularly if different administrators work independently on different systems. A better alternative is to use LLDP, which enables systems to transmit their individual configuration information to peer systems and helps to detect any misconfigurations.

Oracle Solaris supports the use of LLDP to promote the exchange of system and network connectivity information between systems on the network, which reduces the risk of misconfigured network resources.

In this release, LLDP is used by the network diagnostics service to automatically detect problems that could lead to limited or degraded network connectivity, or both. Enabling the LLDP service enhances the ability to perform network diagnostics on your Oracle Solaris system. For more information about network diagnostics, see Chapter 4, “Performing Network Diagnostics With the network-monitor Transport Module Utility” in *Troubleshooting Network Administration Issues in Oracle Solaris 11.4*.

In Oracle Solaris, LLDP is also used to exchange data center bridging exchange protocol (DCBX) Type-Length-Value (TLV) units. DCBX provides configuration information about DCB features such as priority-based flow control (PFC) and enhanced transmission selection (ETS). For more information about DCB, see Chapter 8, “Managing Converged Networks by Using Data Center Bridging”.

With LLDP, the system administrator can easily detect faulty system configurations, particularly in complex networks such as virtual local area networks (VLANs) and link aggregations. Information about the network topology can be obtained readily without having to trace physical connections between servers, switches, and other devices that comprise the network.

Components of an LLDP Implementation

LLDP is implemented with the following components:

- **LLDP package** – You install this package to enable the LLDP. This package includes the LLDP daemon, command-line utilities, the service manifest and scripts, and other components that are required for LLDP to operate.
- **LLDP service** – You can enable the LLDP service by using the `svcadm` command. This service uses the fault management resource identifier (FMRI) of the service management facility (SMF) service instance, `svc:/network/lldp:default`, to manage the LLDP daemon, `lldpd`. This LLDP service is responsible for starting, stopping, restarting, or refreshing the `lldpd` daemon. This service is automatically enabled after you install the LLDP package.

- `lldpadm` **command** – You can use this command to administer LLDP on individual links and to configure the operating mode of LLDP, to specify TLV units that are transmitted, and to configure DCBX TLV units. For information about TLV units, see "Information the LLDP Agent Advertises" on page 104.

  You must use this command to set the per-agent LLDP properties and global LLDP properties and to get LLDP information for a specific agent or its peer.

  The `lldpadm` subcommands are described in the following sections. For more information about the `lldpadm` command, see the `lldpadm`(8) man page.
- **LLDP daemon** – The LLDP services manage LLDP agents on the system. They also interact with `snmpd`, the daemon for the Simple Network Management Protocol (SNMP), to retrieve LLDP information that is received on the system through SNMP.
- **LLDP agents** – LLDP agents are the LLDP instances that are associated with a physical datalink on which LLDP is enabled. LLDP agents transmit information about the datalink to its peer and also receive information from the peer. You can configure an LLDP agent to advertise specific information about the associated physical datalink. You can enable LLDP only on physical datalinks.

Information Sources of the LLDP Agent

The LLDP agent transmits and receives LLDP data units (LLDPDUs). The agent manages and stores information that is contained in these LLDPDUs in the following types of data stores:

- **Local management information base (MIB)** – This data store contains network information that pertains to a system's specific link on which the LLDP agent is enabled. A local MIB contains both common and unique information. For example, the chassis ID is common information that is shared among all the LLDP agents on the system. However, port IDs for the system's datalinks are different. Therefore, each agent manages its own local MIB.
- **Remote MIB** – Information in this data store is received from LLDP agents of peer hosts.

LLDP Agent Modes

The LLDP agent operates in the following modes:

- Transmit only (`txonly`) – The LLDP agent does not process incoming LLDPDUs. Therefore, the remote MIB is empty.
- Receive only (`rxonly`) – The agent processes only incoming LLDPDUs and stores the information in remote MIBs. However, no information from the local MIB is transmitted.

- Transmit and receive (`both`) – The agent transmits local information and processes incoming LLDPDUs and therefore maintains both local and remote MIBs.
- Disabled (`disable`) – The agent does not exist.

For information about setting agent modes, see "How to Enable LLDP for Specific Ports" on page 109.

Information the LLDP Agent Advertises

The LLDP agent transmits system and connectivity information in the LLDP packets or LLDPDUs. These packets contain information units that are individually formatted in TLV format. The information units are also called *TLV units*.

Mandatory TLV Units

Certain TLV units are mandatory and are included in the LLDP packets by default when LLDP is enabled. You cannot use the `lldpadm` command to exclude any of these units.

The following TLV units are mandatory:

- Chassis ID – Information that is generated by the `hostid` command
- Port ID – MAC address of the physical NIC
- TTL (time to live)
- End of protocol data unit (PDU)

Multiple LLDP agents can be enabled on a single system depending on the number of links. The chassis ID and port ID combination uniquely identifies an agent and distinguishes it from other agents on the system.

EXAMPLE 32 Displaying the Chassis ID and Port ID

The following example displays the chassis ID and port ID for an LLDP agent.

```
$ hostid
004e434e

$ dladm show-phys -m net4
LINK                SLOT     ADDRESS            INUSE CLIENT
net4                primary  0:1b:21:87:8b:b4   yes   net4

$ lldpadm show-agent -l net4
```

```
AGENT              CHASSISID          PORTID
net4               004e434e           00:1b:21:87:8b:b4
```

Oracle Solaris LLDP agents use `hostid` as the chassis ID and the port's MAC address as the port ID.

Optional TLV Units

Optional TLV units can be added to an LLDP packet. These optional TLV units enable vendors to insert vendor-specific TLV units to be advertised. LLDP enables defining additional TLV units by using organization unique identifiers (OUIs). An OUI identifies the category for a TLV unit depending on whether the OUI follows the IEEE 802.1 or IEEE 802.3 standard. The LLDP agent properties can be configured to enable or disable the transmission of these optional TLV units.

The following table lists each TLV group, its corresponding name, and the TLV units for each property, and their descriptions. You configure any one of these properties to specify the TLV units to be included in the packets when LLDP is enabled.

TABLE 1 Optional TLV Units for an LLDP Agent

TLV Group	TLV Name	TLV units	Description
Basic management	`basic-tlv`	`sysname`, `portdesc`, `syscapab`, `sysdesc`, `mgmtaddr`	Specifies the system name, port description, system capability, system description, and management address to be advertised.
802.1 OUI	`dot1-tlv`	`vlanname`, `pvid`, `linkaggr`, `pfc`, `appln`, `evb`, `etscfg`, `etsreco`	Specifies the following to be advertised: VLAN name, port VLAN ID, link aggregation, TLV units for priority-based flow control, application, enhanced transmission selection, and edge virtual bridging.
802.3 OUI	`dot3-tlv`	`max-framesize`	Specifies the maximum frame size to be advertised.
Oracle specific OUI (which is defined as `0x0003BA`)	`virt-tlv`	`vnic`	Specifies the VNIC to be advertised if a virtual network is configured.

TLV Unit Properties

Each TLV unit has properties that you can further configure with specific values. If the TLV unit is enabled as an LLDP agent's property, then that TLV unit is advertised in the network

only with the specified values. For example, consider the TLV unit `syscapab`, which advertises a system's capabilities. These capabilities can potentially include support for routers, bridges, repeaters, telephones, and other devices. However, you can set `syscapab` so that only those capabilities that are actually supported on your specific system, such as routers and bridges, are advertised.

The procedure for configuring TLV units depends on whether you are configuring *global TLV units* or *per-agent TLV units*. For information about how to configure TLV units, see "Specifying TLV Units and Values for the LLDP Packet of an Agent" on page 111.

Global TLV units apply to all LLDP agents on the system. The following table lists the global TLV units and their corresponding possible configurations.

TABLE 2 Global TLV Units and Their Properties

TLV Unit	Property Name	Possible Property Values	Value Description
`syscapab`	`supported`	`other`, `repeater`, `bridge`, `wlan-ap`, `router`, `telephone`, `docsis-cd`, `station`, `cvlan`, `sylvan`, `tpmr`	Represents the primary supported functions of the system. Default values are `router`, `station`, and `bridge`.
	`enabled`	Subset of the values listed for `supported`	Represents the enabled functions of the system.
`mgmtaddr`	`ipaddr`	`ipv4` or `ipv6`	Specifies the type of IP addresses that are associated with the local LLDP agent. The addresses are used to reach higher layer entities and assist in discovery by network management. Only one type can be specified.

TLV units that are specific to LLDP agent are managed on a per-agent basis. With per-agent TLV units, the values that you provide are used when the TLV unit is enabled for transmission by a specific LLDP agent.

The following table lists the TLV values and their corresponding possible configurations for an LLDP agent.

TABLE 3 Per-Agent TLV Units and Their Properties

TLV Unit	Property Name	Possible Property Values	Value Description
`pfc`	`willing`	`on`, `off`	Sets an LLDP agent to accept or reject configuration information from a remote system that pertains to priority-based flow control.
`appln`	`apt`	Values are taken from the information that is defined	Configures the Application Priority Table. This table contains the list of application TLV units and their corresponding priorities. The application is

TLV Unit	Property Name	Possible Property Values	Value Description
		in the Application Priority Table.	identified by the `id`/`selector` pair. The contents of the table use the following format: `id/selector/priority` For more information, see "Application Priority Configurations" on page 133.
`etscfg`	`willing`	`on`, `off`	Sets an LLDP agent to accept or reject configuration information from a remote system that pertains to enhanced transmission selection.

For information about per-agent TLV units, see Chapter 8, "Managing Converged Networks by Using Data Center Bridging".

Enabling LLDP on the System

You can configure LLDP to exchange system information with other hosts or peers on the network.

The SMF property `auto-enable-agents` controls the way in which you can enable LLDP agents on the system. With this property, you can choose to enable LLDP globally across all the physical links or only one physical link at a time.

The SMF property `auto-enable-agents` can have one of the following three possible values:

- `yes` enables LLDP on all ports in the transmit and receive (`both`) mode provided that no previous LLDP configuration exists on a port. If a configuration exists on a port, then that port's configuration is retained. For example, if a port has been previously configured with LLDP in the `rxonly` mode, then the LLDP service will not switch the agent to run in the transmit and receive (`both`) mode. LLDP on that port continues to be in the `rxonly` mode. This is the default value of the SMF property `auto-enable-agents`.
- `force` enables LLDP in the transmit and receive (`both`) mode on all ports and overrides any existing LLDP configurations on any port. For example, if a previous LLDP configuration on a port runs in the `rxonly` mode, the LLDP agent is switched to run in the transmit and receive (`both`) mode, which is the default LLDP mode.
- `no` disables automatic enabling of LLDP on all ports except those with existing LLDP configurations. On these ports, the existing LLDP configuration is retained.

Note - Every time you customize the `auto-enable-agents` property, you must restart the LLDP service for the new value to become effective.

How to Install the LLDP Package

By default, LLDP is enabled and ready to be used after you have completed installing the LLDP package.

Before You Begin Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Your Assigned Administrative Rights" in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **Install the package.**

   ```
   $ pkg install lldp
   ```

2. **Determine whether the LLDP service has started.**

   ```
   $ svcs lldp
   STATE          STIME    FMRI
   online         Jul_10   svc:/network/lldp:default
   ```

   If the LLDP service is disabled, start the service with the following command:

   ```
   $ svcadm enable svc:/network/lldp:default
   ```

How to Enable LLDP Globally

Before You Begin You have installed the LLDP package. For the procedure, see "How to Install the LLDP Package" on page 108.

Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 16.

1. **Change the SMF `auto-enable-agents` property to `yes` if it is set to `no`.**

   ```
   $ svccfg -s svc:/network/lldp:default setprop lldp/auto-enable-agents = "yes"
   ```

   By default, this property is set to `yes`.

2. **Restart the LLDP service.**

   ```
   $ svcadm restart svc:/network/lldp:default
   ```

3. **(Optional) Customize global TLV units.**

```
$ lldpadm set-tlvprop -p property=value global-TLV
```

where *property* refers to the property of the global TLV unit.

Next Steps For an explanation of global TLV units, see "TLV Unit Properties" on page 105.

To display a list of global TLVs, type `lldpadm show-tlvprop` or refer to Table 2, "Global TLV Units and Their Properties," on page 106.

For instructions about how to define TLV values, see "How to Define TLV Units" on page 113.

For information about the `lldpadm` command, see the `lldpadm`(8) man page.

▼ How to Enable LLDP for Specific Ports

Before You Begin You have installed the LLDP package. For the procedure, see "How to Install the LLDP Package" on page 108.

Ensure that your role has the appropriate rights profile to perform this procedure. See "Using Rights Profiles to Perform Network Configuration" on page 16.

1. **Change the SMF `auto-enable-agents` property to `no` if it is set to `yes`.**

   ```
   $ svccfg -s svc:/network/lldp:default setprop lldp/auto-enable-agents = "no"
   ```

   By default, this property is set to `yes`.

2. **Restart the LLDP service if you have changed the SMF property `auto-enable-agents` in step 2.**

   ```
   $ svcadm restart svc:/network/lldp:default
   ```

3. **Enable LLDP agents on selected ports or links.**

   ```
   $ lldpadm set-agentprop -p mode=value agent
   ```

   where *agent* is the LLDP agent and is identified by the physical link on which the agent is enabled. For example, if you enable LLDP on `net0`, the agent is `net0`.

   The property `mode` can be set to one of four possible values that represent the LLDP agent's modes of operation: `txonly`, `rxonly`, `both`, and `disable`. For an explanation of these values, see "LLDP Agent Modes" on page 103.

4. **Specify the TLV units that the LLDP agent can advertise.**

   ```
   $ lldpadm set-agentprop -p property=value agent
   ```

   For an explanation of the properties of the LLDP agent, see "Information the LLDP Agent Advertises" on page 104.

   To display a list of the other properties of the LLDP agent, type `lldpadm show-agentprop` or refer to Table 1, "Optional TLV Units for an LLDP Agent," on page 105.

   For instructions about how to specify TLV units for LLDP packet of an agent, see "How to Specify TLV Units for the LLDP Packet of an Agent" on page 112.

5. **(Optional) Customize the per-agent TLV units.**

   ```
   $ lldpadm set-agenttlvprop -p property=value -a agent per-agent-TLV
   ```

   where *property* refers to the property of the per-agent TLV unit.

   For an explanation of per-agent TLV units, see "TLV Unit Properties" on page 105.

   To display a list of per-agent TLVs, type `lldpadm show-agenttlvprop` or refer to Table 3, "Per-Agent TLV Units and Their Properties," on page 106.

   For instructions about how to define TLV values, see "How to Define TLV Units" on page 113.

   For information about the `lldpadm` command, see the `lldpadm`(8) man page.

Example 33 Customizing the `auto-enable-agents` SMF Property

The following example shows the different way in which LLDP is enabled if you change the value of the SMF property `auto-enable-agents`. For example, given a system with four ports, LLDP is configured on two ports as follows:

- `net0`: `both` mode
- `net1`: `rxonly` mode
- `net2` and `net3`: none

If the SMF property `auto-enable-agents` has the default value `yes`, LLDP is automatically enabled on `net2` and `net3`. You can display the LLDP configuration as follows:

```
$ lldpadm show-agentprop -p mode
AGENT   PROPERTY   PERM   VALUE   EFFECTIVE   DEFAULT   POSSIBLE
net0    mode       rw     both      --        disable   txonly,rxonly,both,disable
net1    mode       rw     rxonly    --        disable   txonly,rxonly,both,disable
net2    mode       rw     both      --        disable   txonly,rxonly,both,disable
```

```
net3    mode       rw     both      --        disable   txonly,rxonly,both,disable
```

If you switch the SMF property to `no`, the configuration changes when you restart the service.

```
$ svccfg -s svc:/network/lldp:default setprop lldp/auto-enable-agents = "no"
$ svcadm restart svc:/network/lldp:default
$ lldpadm show-agentprop -p mode
AGENT   PROPERTY   PERM   VALUE    EFFECTIVE    DEFAULT   POSSIBLE
net0    mode       rw     both       --         disable   txonly,rxonly,both,disable
net1    mode       rw     rxonly     --         disable   txonly,rxonly,both,disable
net2    mode       rw     disable    --         disable   txonly,rxonly,both,disable
net3    mode       rw     disable    --         disable   txonly,rxonly,both,disable
```

In the sample output, `net2` and `net3`, whose LLDP modes were previously automatically enabled, are now flagged as disabled. However, no change occurs on `net0` and `net1`, whose LLDP agents were previously configured.

Example 34 Enabling LLDP on Multiple Datalinks

This example shows how to enable LLDP selectively. A system has two datalinks, `net0` and `net1`. On `net0`, to set the agent to transmit and receive LLDP packets, and on `net1`, to set the agent to only transmit LLDP packets, type the following commands:

```
$ svccfg -s svc:/network/lldp:default setprop lldp/auto-enable-agents = "no"
$ svcadm restart svc:/network/lldp:default
$ lldpadm set-agentprop -p mode=both net0
$ lldpadm set-agentprop -p mode=txonly net1
$ lldpadm show-agentprop -p mode
AGENT   PROPERTY   PERM   VALUE    EFFECTIVE   DEFAULT   POSSIBLE
net0    mode       rw     both       --        disable   txonly,rxonly,both,disable
net1    mode       rw     txonly     --        disable   txonly,rxonly,both,disable
```

Specifying TLV Units and Values for the LLDP Packet of an Agent

You can specify TLV units such as `dot1-tlv` and `basic-tlv` as property values for an LLDP agent. You can further configure these property values. To specify TLV units, use the `lldpadm set-agentprop` command. For more information, see `lldpadm`(8) man page. When the TLV unit is specified as a property of an LLDP agent, then that TLV unit is advertised in the network only with the values that you have specified for the TLV units. For information about TLV units and properties see, “TLV Unit Properties” on page 105.

How to Specify TLV Units for the LLDP Packet of an Agent

This procedure explains how to specify TLV units to be advertised in an LLDP packet that an agent transmits.

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 16.

1. **(Optional) Identify the LLDP agent property that can contain the TLV unit that you want to add by displaying the TLV units.**

   ```
   $ lldpadm show-agentprop agent
   ```

   This command helps you to see the TLV units that are already set for each property. If you do not specify a property, this command displays all the LLDP agent properties and their TLV values. For a list of agent properties, see Table 1, "Optional TLV Units for an LLDP Agent," on page 105.

2. **Add or remove the TLV unit from the property.**

   ```
   $ lldpadm set-agentprop -p property[+|-]=value[,...] agent
   ```

   You can use qualifiers to add (+) or remove (-) values from the list of values for properties that accept multiple values.

   If you do not use the add (+) or remove (-) qualifiers, then the value that you set replaces all the values that were previously defined for the property.

3. **(Optional) Display the new values for the property.**

   ```
   $ lldpadm show-agentprop -p property agent
   ```

Example 35 Adding Optional TLV Units to an LLDP Packet

In the following example, the LLDP agent has `net0` configured to advertise VLAN information in its LLDP packet. The LLDP packet is further configured to include system capabilities, link aggregation, and virtual NIC information as items that the LLDP can advertise. Later, the VLAN description is removed from the packet.

```
$ lldpadm show-agentprop net0
AGENT   PROPERTY   PERM   VALUE          EFFECTIVE   DEFAULT   POSSIBLE
net0    mode       rw     both             --        disable   txonly,rxonly,both,
disable
net0    basic-tlv  rw     sysname,         --        none      none,portdesc,
```

```
                        sysdesc                         sysname,sysdesc,
                                                        syscapab,mgmtaddr,
                                                        all
net0    dot1-tlv    rw      vlanname,         --      none      none,vlanname,pvid,
                            pvid,pfc                    linkaggr,pfc,appln,
                                                        evb,etscfg,etsreco,all
net0    dot3-tlv    rw      max-framesize     --      none      none, max-framesize,
                                                        all
net0    virt-tlv    rw      none              --      none      none,vnic,all

$ lldpadm set-agentprop -p basic-tlv+=syscapab,dot1-tlv+=linkaggr,virt-tlv=vnic net0
$ lldpadm set-agentprop -p dot1-tlv-=vlanname net0

$ lldpadm show-agentprop -p net0
AGENT   PROPERTY   PERM   VALUE          EFFECTIVE   DEFAULT   POSSIBLE
net0    mode       rw     both             --        disable   txonly,rxonly,both,
                                                               disable
net0    basic-tlv  rw     sysname,         --        none      none,portdesc,
                          sysdesc,                             sysname,sysdesc,
                          syscapab                             syscapab,mgmtaddr,
                                                               all
net0    dot1-tlv   rw     pvid,pfc         --        none      none,vlanname,pvid,
                          linkaggr                             linkaggr,pfc,appln,
                                                               evb,etscfg,etsreco,all
net0    dot3-tlv   rw     max-framesize    --        none      none, max-framesize,
                                                               all
net0    virt-tlv   rw     vnic             --        none      none,vnic,all
```

How to Define TLV Units

This procedure explains how to provide values for specific TLV units.

Tip - You can reset the TLV properties to their default values by using the `lldpadm reset-tlvprop` command for global TLV units and `lldpadm reset-agenttlvprop` command for per-agent TLV units.

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 16.

1. **Configure a global or a per-agent TLV unit.**

   - **To configure a global TLV unit, set the appropriate TLV property to contain the values that you want to advertise.**

```
$ lldpadm set-tlvprop -p TLV-property=value[,value,value,...] TLV-name
```

where *TLV-name* is the name of the global TLV unit and *TLV-property* is a property of that TLV unit. You can assign multiple values to the property. For a list of global TLV units and their properties, see Table 2, “Global TLV Units and Their Properties,” on page 106.

- **To configure a per-agent TLV unit, configure the appropriate TLV property of the LLDP agent to contain the values that you want the agent to advertise.**

```
$ lldpadm set-agenttlvprop -p TLV-property[+|-]=value[,value,value,...] -a agent TLV-name
```

where *TLV-name* is the name of the agent TLV unit and *TLV-property* is a property of that TLV unit. You can assign multiple values to the property. For a list of per-agent TLV units and their properties, see Table 3, “Per-Agent TLV Units and Their Properties,” on page 106.

You can use qualifiers to add (+) or remove (-) values from the list of values for properties that accept multiple values.

2. **(Optional) Display the values of the TLV property that you have configured.**

- **To display the global TLV property values:**

```
$ lldpadm show-tlvprop
```

- **To display the values of the TLV property of an agent:**

```
$ lldpadm show-agenttlvprop
```

Example 36 Defining TLV Values for the `syscapab` and `mgmtaddr` TLV Units

In the following example, specific information about the capabilities of the system to be advertised in the LLDP packet and management IP address is configured.

```
$ lldpadm set-tlvprop -p supported=bridge,router,repeater syscapab
$ lldpadm set-tlvprop -p enabled=router syscapab
$ lldpadm set-tlvprop -p ipaddr=192.0.2.2 mgmtaddr

$ lldpadm show-tlvprop
TLVNAME    PROPERTY   PERM  VALUE        DEFAULT         POSSIBLE
syscapab   supported  rw    bridge,      bridge,router,  other,router,
                            router,      station         repeater,bridge,
                            repeater                     wlan-ap,telephone,
                                                         docis-cd,station,
                                                         cvlan,svlan,tpmr
```

```
syscapab    enabled     rw     router       none           bridge,router,
                                                           repeater
mgmtaddr    ipaddr      rw     192.0.2.2    none           --
```

For information about configuring per-agent TLV properties, see Chapter 8, “Managing Converged Networks by Using Data Center Bridging”.

Disabling LLDP

This section describes how to disable LLDP selectively on individual ports.

How to Disable LLDP

To disable LLDP across all of the system's interfaces, perform the following steps.

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 16.

1. **Change the SMF LLDP property to `no`, which disables automatic enabling of LLDP on all ports except those with existing LLDP configurations.**

   ```
   $ svccfg -s svc:/network/lldp:default setprop lldp/auto-enable-agents = "no"
   ```

2. **Restart the LLDP service.**

   ```
   $ svcadm restart svc:/network/lldp:default
   ```

3. **Disable LLDP on each port whose previous LLDP configuration is retained.**

   - **To disable LLDP by changing the mode of the agent:**

     ```
     $ lldpadm set-agentprop -p mode=disable agent
     ```

     where *agent* is the LLDP agent and is identified by the physical link on which the agent is enabled. For example, if you enable LLDP on `net0`, the agent is `net0`.

   - **To disable LLDP by removing the LLDP configuration from the port:**

     ```
     $ lldpadm reset-agentprop -p mode agent
     ```

     In this command, you do not set a value for the `mode` property.

Caution - If `auto-enable-agents` that is set to `no` is switched back to `yes`, LLDP behaves differently than if the agent's mode on that port were simply disabled.

Monitoring LLDP Agents

The `lldpadm show-agent` command displays complete information that is advertised by an LLDP agent. Relative to a given system, the advertisement can be information about the local system that is transmitted to the rest of the network or information that is received by the system from other systems on the same network.

Displaying the Advertised Information

The information can be either local or remote. *Local* information comes from the local LLDP agent. *Remote* information comes from other LLDP agents on the network that is received by the local LLDP agent.

Use the `lldpadm show-agent` command to display the advertised information.

```
$ lldpadm show-agent -[l|r][v] agent
```

- `-l` displays local information advertised by the local LLDP agent.
- `-r` displays remote information received by the LLDP agent.
- `-v` displays detailed local or remote information.

EXAMPLE 37 Displaying Advertised LLDP Agent Information

The following example shows how to display the information that is being advertised locally or remotely by an LLDP agent. By default, the information is displayed in short form. By using the `-v` option, you can obtain verbose or detailed information.

To display local information that is advertised by the LLDP agent:

```
$ lldpadm show-agent -l net0
AGENT   CHASSISID   PORTID
net0    004bb87f    00:14:4f:01:77:5d
```

To display remote information that is advertised by the LLDP agent:

```
$ lldpadm show-agent -r net0
AGENT   SYSNAME   CHASSISID   PORTID
net0    hostb     0083b390    00:14:4f:01:59:ab
```

To display local information in the verbose mode, use the -v option:

```
$ lldpadm show-agent -l -v net4
                                    Agent: net4
                          Chassis ID Subtype: Local(7)
                                  Chassis ID: 00843300
                             Port ID Subtype: MacAddress(3)
                                     Port ID: 00:1b:21:89:03:d0
                           Port Description: --
                               Time to Live: 21 (seconds)
                                System Name: --
                          System Description: --
                       Supported Capabilities: --
                         Enabled Capabilities: --
                           Management Address: --
                           Maximum Frame Size: --
                                 Port VLAN ID: --
                                 VLAN Name/ID: vlan1/22
                             VNIC PortID/VLAN ID: 02:08:20:63:2d:9d,02:08:20:e5:6c:af/21
                     Aggregation Information: --
                                  PFC Willing: On
                                      PFC Cap: 8
                                      PFC MBC: False
                                   PFC Enable: 4
                                  PFC Pending: True
                 Application(s)(ID/Sel/Pri): --
                                  ETS Willing: On
                         ETS Configured CBS: 0
                         ETS Configured TCS: 8
                         ETS Configured PAT: 0,1,2,3,4,5,6,7
                         ETS Configured BAT: 40,20,0,40,0,0,0,0
                         ETS Configured TSA: 2,2,2,2,2,2,2,2
                        ETS Recommended PAT: 0,1,2,3,4,5,6,7
                        ETS Recommended BAT: 40,20,0,40,0,0,0,0
                        ETS Recommended TSA: 2,2,2,2,2,2,2,2
                                  EVB Mode: Station
                            EVB GID (Station): Not Supported
                     EVB ReflectiveRelay REQ: Not Requested
                    EVB ReflectiveRelay Status: RR Not Enabled
                              EVB GID (Bridge): Not Supported
          EVB ReflectiveRelay Capable (RRCAP): Not Supported
          EVB ReflectiveRelay Control (RRCTR): Not Enabled
                         EVB max Retries (R): 0
```

```
            EVB Retransmission Exponent (RTE): 0
                EVB Remote or Local(ROL) and
                    Resource Wait Delay (RWD): Local
                EVB Resource Wait Delay (RWD): 0
               EVB Remote or Local (ROL) and
                      Reinit Keep Alive (RKA): Local
                  EVB Reinit Keep Alive (RKA): 0
                     Next Packet Transmission: 4 (seconds)
```

To display remote information in the verbose mode, use the -v option:

```
$ lldpadm show-agent -r -v net4
                                    Agent: net4
                          Chassis ID Subtype: Local(7)
                                  Chassis ID: 00843300
                             Port ID Subtype: MacAddress(3)
                                     Port ID: 00:1b:21:89:03:d0
                           Port Description: --
                               Time to Live: 21 (seconds)
                                System Name: --
                          System Description: --
                       Supported Capabilities: --
                         Enabled Capabilities: --
                           Management Address: --
                           Maximum Frame Size: --
                                 Port VLAN ID: --
                                 VLAN Name/ID: vlan1/22
                              VNIC PortID/VLAN ID: 02:08:20:63:2d:9d,02:08:20:e5:6c:af/21
                      Aggregation Information: --
                                  PFC Willing: On
                                      PFC Cap: 8
                                      PFC MBC: False
                                   PFC Enable: 4
                  Application(s)(ID/Sel/Pri): --
                                  ETS Willing: On
                         ETS Configured CBS: 0
                         ETS Configured TCS: 8
                         ETS Configured PAT: 0,1,2,3,4,5,6,7
                         ETS Configured BAT: 40,20,0,40,0,0,0,0
                         ETS Configured TSA: 2,2,2,2,2,2,2,2
                        ETS Recommended PAT: 0,1,2,3,4,5,6,7
                        ETS Recommended BAT: 40,20,0,40,0,0,0,0
                        ETS Recommended TSA: 2,2,2,2,2,2,2,2
                                  EVB Mode: Station
                           EVB GID (Station): Not Supported
                      EVB ReflectiveRelay REQ: Not Requested
                    EVB ReflectiveRelay Status: RR Not Enabled
                             EVB GID (Bridge): Not Supported
          EVB ReflectiveRelay Capable (RRCAP): Not Supported
```

```
EVB ReflectiveRelay Control (RRCTR): Not Enabled
                EVB max Retries (R): 0
  EVB Retransmission Exponent (RTE): 0
      EVB Remote or Local(ROL) and
          Resource Wait Delay (RWD): Local
      EVB Resource Wait Delay (RWD): 0
     EVB Remote or Local (ROL) and
            Reinit Keep Alive (RKA): Local
        EVB Reinit Keep Alive (RKA): 0
            Information Valid Until: 19 (seconds)
```

Displaying LLDP Statistics

You can display LLDP statistics to obtain information about LLDP packets that are being advertised by the local system or by remote systems. The statistics refer to significant events that involve LLDP packet transmission and reception.

- To display all statistics about LLDP packet transmission and reception:

  `$ lldpadm show-agent -s` *agent*

- To display selected statistics information, use the `-o` option:

  `$ lldpadm show-agent -s -o` *field*`[,`*field*`,...]`*agent*

  field refers to any field name in the output of the `show-agent -s` command.

EXAMPLE 38 Displaying LLDP Packet Statistics

This example shows how to display information about LLDP packet advertisement.

```
$ lldpadm show-agent -s net0
AGENT IFRAMES IERR IDISCARD OFRAMES OLENERR TLVDISCARD TLVUNRECOG AGEOUT
net0        9    0        0      14       0          4          5      0
```

This output provides the following information:

- `AGENT` specifies the name of the LLDP agent, which is identical to the datalink on which the LLDP agent is enabled.
- `IFRAMES`, `IERR`, and `IDISCARD` display information about packets being received, incoming packets with errors, and incoming packets that are dropped.
- `OFRAMES` and `OLENERR` refer to outgoing packets and packets that have length errors.

- `TLVDISCARD` and `TLVUNRECOG` display information about TLV units that are discarded and TLV units that are not recognized.
- `AGEOUT` refers to packets that have timed out.

The example indicates that out of 9 frames received into the system, 5 TLV units are unrecognized, possibly because of noncompliance with standards. The example also shows that 14 frames were transmitted by the local system to the network.

EXAMPLE 39 Displaying Selected LLDP Packet Statistics

This example shows how to display selected statistics information.

```
$ lldpadm show-agent -s -o iframes,oframes net4
IFRAMES  OFRAMES
0        10
```

♦ ♦ ♦ CHAPTER 8

Managing Converged Networks by Using Data Center Bridging

Traditionally, different networks are used for traffic management based on the application requirements, and load distribution of network traffic that is based on the available bandwidth. For example, local area network (LAN) uses Ethernet and storage area network (SAN) uses fibre channel. However, data center bridging enhances the Ethernet making it more suitable for running different types of traffic, which is converged traffic, and also to support features such as losslessness. DCB enables efficient network infrastructure by consolidating SAN and LAN and thereby reducing operational and management costs in large deployments.

This chapter contains the following topics:

- "Overview of Data Center Bridging" on page 121
- "Priority-Based Flow Control" on page 123
- "Enhanced Transmission Selection" on page 124
- "Enabling DCBX" on page 125
- "Setting the Mode of Operation for DCB" on page 126
- "Customizing Priority-Based Flow Control for DCB" on page 128
- "Displaying PFC Configuration Information" on page 130
- "Application Priority Configurations" on page 133
- "Customizing Enhanced Transmission Selection for DCB" on page 134
- "Recommending ETS Configuration to the Peer" on page 136
- "Displaying ETS Configuration Information" on page 138

Overview of Data Center Bridging

Data center bridging is used to manage the bandwidth, relative priority, and flow control of multiple traffic types when sharing the same network link, for example, when sharing a datalink between networking and storage protocols. Fibre channel can be dedicated to host this type

of traffic. However, using dedicated links to service only fibre channel traffic can be costly. Therefore, fibre channel over Ethernet (FCoE) is more commonly used. DCB addresses the sensitivity of fibre channels to packet loss while traversing an Ethernet network.

DCB enables information exchange with the peers about the features that support converged network by using LLDP. The information is related to the configurations affecting the integrity of network packets especially in heavy traffic environments, such as data centers. DCB enables efficient network infrastructure by consolidating storage area network (SAN) and local area network (LAN) and thereby reducing operational and management costs in large deployments.

DCB distinguishes traffic based on priorities, which are also called class of service (CoS) priorities. The host and the next hop use the DCB exchange protocol (DCBX) to negotiate a network configuration, such as no traffic loss and minimum bandwidth share, based on priorities. This process enables packets from different applications on the host and in the network to be treated according to their priorities and the corresponding configuration to be negotiated by using DCBX.

Oracle Solaris supports the IEEE 802.1qaz DCBX specification and also the pre-standard Converged Enhanced Ethernet (CEE) DCBX specification v1.01 to enable interoperation with a diverse set of switches when using DCB. For more information about how to choose different versions for negotiations, see “Setting the Mode of Operation for DCB” on page 126.

Each packet in a DCB network has a VLAN header that contains a DCB 3-bit priority value, which is a DCB priority. This IEEE 802.1p priority value differentiates each Ethernet packet in the network from the other packets. Depending on the priority values of the packets, you can configure DCB to allocate specific bandwidth to the packets. For example, all packets with a priority `1` must have PFC enabled and all packets with a priority `2` must have PFC disabled and a bandwidth share of 10%.

You can configure DCB features such as priority-based flow control (PFC) and enhanced transmission selection (ETS) based on priorities. For more information about PFC and ETS, see “Priority-Based Flow Control” on page 123 and “Enhanced Transmission Selection” on page 124.

The DCB `cos` datalink property enables you to specify the CoS or priority of the datalink. The `cos` value that is set on a primary datalink does not apply to the VNICs that are created over this physical link. For information about customizing PFC based on the `cos` property, see “Customizing Priority-Based Flow Control for DCB” on page 128. For information about customizing ETS based on the `cos` property, see “Customizing Enhanced Transmission Selection for DCB” on page 134.

In Oracle Solaris, LLDP is used to exchange DCBX type-length-value (TLV) units. For more information about LLDP, see Chapter 7, “Exchanging Network Connectivity Information With Link Layer Discovery Protocol”. Provided that the underlying network interface card (NIC) supports DCB features such as priority-based flow control and enhanced transmission selection,

configuration information for these features can be shared with peer hosts on the network, as follows:

- PFC prevents packet loss by implementing a mechanism that pauses traffic flow for packets with a defined class of service (CoS). For more information about CoS, see the description of the `cos` link property in the `dladm`(8) man page.
- ETS enables bandwidth sharing among packets based on the defined CoS. See "Enhanced Transmission Selection" on page 124.

Considerations When Using DCB

Note the following considerations for using DCB:

- DCB is *only* supported on Intel Niantic physical NICs.

  To verify whether the NIC supports DCB, issue the following command:

  ```
  $ dladm show-linkprop -p ntcs agent
  ```

  A property value that is greater than zero (`0`) indicates that the NIC supports DCB.
- DCB ports that are configured in DCB mode cannot be aggregated (trunk or DLMP mode).
- Because DCB supports only the IEEE and CEE versions of DCBX (not the CIN version), external bridges must support the IEEE or CEE version to interoperate with Oracle Solaris DCB.
- DCB supports ETS configuration and recommendation TLVs.
- DCB is only supported in the eight traffic class configuration.
- DCB does not support congestion notification (CN).

Priority-Based Flow Control

Priority-based flow control (PFC) extends the standard IEEE 802.3x PAUSE frame to include IEEE 802.1p CoS values. With PFC, instead of halting all traffic on the link when a PAUSE frame is sent, traffic is paused only for those CoS values that are enabled in the PFC frame. A PFC frame is sent for the enabled `cos` property value for which traffic needs to be paused. The sending host stops traffic for that `cos` property value while traffic for other disabled `cos` property values are unaffected. After a time interval that is specified in the PFC frame, transmission resumes for the paused packets.

Pausing based on CoS values ensures that packets are not dropped for that `cos` property value. For packets without any defined CoS value or with CoS values that do not have PFC enabled,

no PAUSE frames are sent. Therefore, traffic continues to flow, and packets might be dropped during traffic congestion. Handling loss of packets depends on the protocol stack, for example, TCP.

Two types of DCB information exist on the host: local DCB information and remote DCB information. For the PFC features to be effective, the local and remote types of DCB information for PFC on the host must be symmetric. The local host must be able to match the DCB information that it receives from the peer. If you enable DCB on your system, DCB synchronizes the DCB information with the peer.

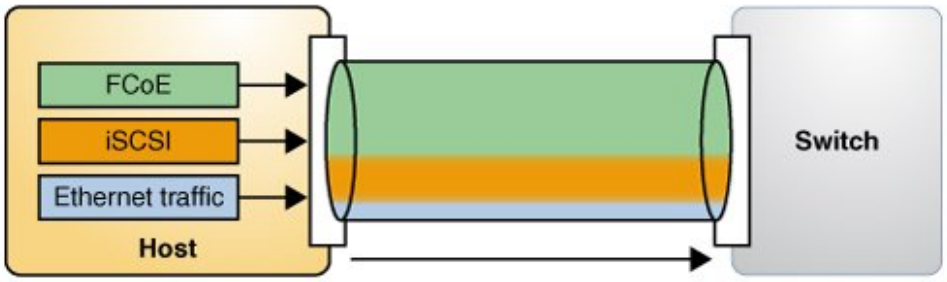
In most cases, the default configuration for PFC is sufficient. This configuration is automatically set up when you enable LLDP. However, you can adjust different options when configuring PFC. For more information, see “Customizing Priority-Based Flow Control for DCB” on page 128 and “Displaying PFC Configuration Information” on page 130.

Enhanced Transmission Selection

ETS is a DCB feature that enables you to allocate bandwidth on a NIC to applications based on their DCB priority.

The following figure shows the ETS feature of DCB in a network.

FIGURE 16 Enhanced Transmission Selection in DCB



The host in the figure has different types of traffic, such as FCoE and iSCSI, that share the link bandwidth. In the figure, the priority and bandwidth as shown in the following table are assigned for different types of traffic.

Traffic	Priority	Bandwidth
FCoE	3	60%
iSCSI	4	30%
Ethernet (non iSCSI) traffic	0	10%

The DCBX ETS TLV with the corresponding ETS bandwidth allocation is as follows:

Priority	0	1	2	3	4	5	6	7
Bandwidth Allocation Percentage	10	0	0	60	30	0	0	0

To use the ETS feature, the NIC must support the feature and run in the DCB mode. When you enable LLDP, the default configuration for the ETS feature is automatically set up if the underlying link supports DCB. However, you can modify the default configuration. For more information, see “Customizing Enhanced Transmission Selection for DCB” on page 134, “Recommending ETS Configuration to the Peer” on page 136, and “Displaying ETS Configuration Information” on page 138.

Enabling DCBX

Support for DCBX is automatically enabled when you enable LLDP. This procedure provides alternative manual steps in case certain automatic processes fail.

▼ How to Enable the Data Center Bridging Exchange Feature Manually

Before You Begin Ensure that you install LLDP. For more information about enabling LLDP, see “Enabling LLDP on the System” on page 107.

1. **Become an administrator.**
   For more information, see “Using Your Assigned Administrative Rights” in *Securing Users and Processes in Oracle Solaris 11.4*.

2. **Verify that the LLDP service is running.**

```
$ svcs lldp
```

If the LLDP service is disabled, start the service with the following command:

```
$ svcadm enable svc:/network/lldp:default
```

3. **Ensure that the LLDP agent is running on Rx and Tx modes.**

```
$ lldpadm show-agentprop -p mode agent
```

If the LLDP agent is not enabled on both modes, type the following command:

```
$ lldpadm set-agentprop -p mode=both agent
```

For the other possible configurations of the LLDP agents, see "Enabling LLDP on the System" on page 107.

4. **Verify that the underlying NIC supports DCB.**

```
$ dladm show-linkprop -p ntcs agent
```

A property value that is greater than zero (`0`) indicates that the NIC supports DCB.

Setting the Mode of Operation for DCB

Oracle Solaris hosts can use either the IEEE DCBX or CEE DCBX protocol to exchange information about DCB features with their peers that are directly connected, for example, the first hop switch. The exchange enables negotiating and configuring DCB features both on the host and the peer.

The following table shows the DCBX applications that are supported for the IEEE DCBX and CEE DCBX versions.

TABLE 4 Applications Supported for IEEE DCBX and CEE DCBX Versions

Applications	IEEE DCBX	CEE DCBX
PFC	Yes	Yes
Application TLVs	Yes	Yes
ETS configuration	Yes	No
ETS recommendation	Yes	No

You can select `ieee`, `cee`, or `auto` as the mode of operation for DCB depending on the standard that the switch supports. The default mode is `auto`, which by default operates in IEEE mode and

switches to CEE mode when the host receives a CEE packet from the peer. To select the mode, you need to set the property `dcbx-version` by using the `lldpadm` command.

If you explicitly set the mode, the mode does not transition to CEE or IEEE when the CEE or IEEE packets are received.

The transition from IEEE to CEE happens only once. Subsequently, when the peer changes its DCBX mode, Oracle Solaris will not switch mode automatically.

Some switches initiate exchange of information regardless of the peer. However, some switches might listen to the peer and respond only if it supports the DCBX version. For example, if you configure the DCBX mode of an Oracle Solaris host as `auto`, then by default it sends IEEE DCBX packets. If this host is connected to a switch that does not support IEEE DCBX, then the switch might not respond even if it supports CEE DCBX version. In such cases, you must explicitly configure the DCBX mode as `cee`.

How to Set the Mode of Operation for DCB

1. **Become an administrator.**

2. **(Optional) Display the current DCBX mode.**

```
$ lldpadm show-agentprop -p dcbx-version net0
AGENT        PROPERTY        PERM VALUE   EFFECTIVE     DEFAULT        POSSIBLE
net0         dcbx-version    rw   auto    ieee          auto           auto,ieee,cee
```

3. **Set the mode of the operation.**

```
$ lldpadm set-agentprop -p dcbx-version=DCBX-mode net0
```

where *DCBX-mode* can be set to one of the following values:

- `auto` – The default DCBX mode. When the mode is `auto`, DCBX operates in IEEE mode and switches to CEE mode when the host receives a CEE packet from the peer.
- `ieee` – Uses only Ithe EEE protocol to exchange information.
- `cee` – Uses only the CEE protocol to exchange information.

4. **(Optional) Display the current mode.**

```
$ lldpadm show-agentprop -p dcbx-version net0
AGENT        PROPERTY        PERM VALUE   EFFECTIVE     DEFAULT        POSSIBLE
net0         dcbx-version    rw   auto    cee           auto           auto,ieee,cee
```

Tip - To reset the DCBX mode to the default setting, use the following command:

```
$ lldpadm reset-agentprop -p dcbx-version net0
```

Customizing Priority-Based Flow Control for DCB

PFC and ETS are enabled by default only if the underlying NIC is in DCB mode. If you prefer to use only PFC, then you must remove `etscfg` from the `dot1 -tlv` property of the LLDP agent by using the following command:

```
$ lldpadm set-agentprop -p dot1-tlv-=etscfg net0
```

For a list of possible values for `dot1 -tlv`, refer to Table 1, “Optional TLV Units for an LLDP Agent,” on page 105.

Setting the PFC-Related Datalink Properties

The PFC feature of DCB provides the following datalink properties:

- `pfcmap` – Provides information about priority definitions and mappings. The `pfcmap` property refers to an 8-bit mask (`0`–`7`) that represent priorities. The lowest bit represents priority `0` while the highest bit represents priority `7`. Each bit in this mask signifies whether PFC is enabled for a corresponding priority. By default, `pfcmap` is set to `1111111`, which means that PFC is enabled on all the priorities.
- `pfcmap-remote` – Specifies the operative PFC mapping on the remote peer. This property is read-only.

In a DCB network, when a receiver is unable to keep up with the incoming rate of traffic, it sends a PFC frame to the sender requesting the sender to pause traffic for priorities that have PFC enabled. For any packet transmitted over a link, DCB sends a PFC frame to the sending host if traffic congestion accumulates on the receiving host. To send PFC frames properly, the communicating hosts must have symmetric DCB configuration information. A system can automatically adjust its PFC configurations to match the PFC configurations on the remote peer. You can determine the operative PFC mapping on the local host by using the `dladm show-linkprop` command that displays the `EFFECTIVE` value for the `pfcmap` property. For more information, see “Displaying Datalink Properties” on page 130.

▼ How to Customize Priority-Based Flow Control for DCB

1. **Become an administrator.**

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

2. **Ensure that the `flow-control` property of the datalink is set to `pfc`.**

   ```
   $ dladm show-linkprop -p flow-control datalink
   ```

   If the property is not set to `pfc` or `auto`, use the following command:

   ```
   $ dladm set-linkprop -p flow-control=pfc datalink
   ```

3. **Set the `pfcmap` property to a value other than the default value `11111111`.**

   ```
   $ dladm set-linkprop -p pfcmap=value datalink
   ```

   For example, to enable priority only on CoS priority 6, type the following command:

   ```
   $ dladm set-linkprop -p pfcmap=01000000 net0
   ```

Setting the PFC TLV Units

DCB uses PFC TLV units to exchange PFC information between hosts. Either hosts must have the same value for the `pfcmap` property or at least one host must be willing to accept the configuration of its peer. You can set the `pfcmap` property by using the `dladm set-linkprop` command.

The TLV property `willing` indicates whether the host is ready to accept the peer's configuration if the host configuration differs from the peer's configuration. By default, the property value of `willing` is set to `on`, which indicates that the host will accept the peer's configuration.

To verify that the host can synchronize its PFC information with the PFC information on the remote peer, you must determine whether the `willing` is set to `on` by using the following command:

```
$ lldpadm show-agenttlvprop -p willing -a agent pfc
```

If the PFC TLV property `willing` is set to `off`, type the following command to set the property `willing` to `on` and enable synchronization.

```
$ lldpadm set-agenttlvprop -p willing=on -a agent pfc
```

where *agent* is the datalink on which the agent is enabled.

EXAMPLE 40 Enabling Synchronization Between the Host and the Peer

To enable synchronization for a `net0` datalink, type the following command:

```
$ lldpadm set-agenttlvprop -p willing=on -a net0 pfc
```

```
$ dladm show-linkprop -p pfcmap,pfcmap-remote net0
LINK   PROPERTY      PERM  VALUE      EFFECTIVE  DEFAULT    POSSIBLE
net0   pfcmap        rw    11111111   00010000   11111111   00000000-11111111
net0   pfcmap-remote r-    --         00010000   11111111   --
```

In the example, the `pfcmap` and `pfcmap-remote` properties have the value `00010000`. This indicates that the local host has synchronized with the peer. Hence, PFC is enabled with a priority of `4` on both the host and the peer.

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

Displaying PFC Configuration Information

This section describes commands to display information related to PFC after LLDP and DCB are configured and provides examples to show the use of these commands.

Displaying Datalink Properties

The following command displays the priority definitions and the effective PFC mappings on the datalink:

```
$ dladm show-linkprop -p pfcmap,pfcmap-remote datalink
```

On a datalink with matching PFC information between the local and remote peers, the values of the `EFFECTIVE` column for `pfcmap` and `pfcmap-remote` properties are identical regardless of the value set for the `pfcmap` property. If the ability to synchronize is disabled on the local host, then the `EFFECTIVE` field for the `pfcmap` property reflects the value of the `pfcmap` property for the local host.

EXAMPLE 41 Displaying PFC-Related Datalink Properties

This example shows how to display the status of physical datalink properties that are related to priority-based flow control.

```
$ dladm show-linkprop -p pfcmap,pfcmap-remote net0
LINK   PROPERTY       PERM   VALUE      EFFECTIVE  DEFAULT    POSSIBLE
net0   pfcmap         rw     11111111   11111111   11111111   00000000-11111111
net0   pfcmap-remote  r-     --         --         --         --
```

In the example, the `value` field for the `pfcmap` property has the value of `11111111`. This value indicates that the PFC mapping on the local host has the default value where all eight priorities are enabled. The `EFFECTIVE` values for the `pfcmap` and `pfcmap-remote` properties are `11111111` and `--`. These mismatched values for the `EFFECTIVE` field indicate that the local host has not synchronized its PFC information with the remote peer.

You can use the `lldpadm show-agenttlvprop` command to verify the value of the `willing` property and the `lldpadm show-agent -r` command to check the PFC TLV information from the peer.

Displaying the Capability of the Local Host to Synchronize PFC Information

The following command displays the PFC TLV property that controls a host's capability to synchronize its PFC mapping with a peer.

```
$ lldpadm show-agenttlvprop -a agent pfc
```

where *agent* is identified by the datalink on which LLDP is enabled.

EXAMPLE 42 Displaying the Capability of the Local Host to Synchronize PFC Information

This example shows how to display the current status of the host's ability to adjust to PFC configurations of the peer.

```
$ lldpadm show-agenttlvprop -a net0 pfc -p willing
AGENT   TLVNAME   PROPERTY   PERM   VALUE   EFFECTIVE   DEFAULT   POSSIBLE
net0    pfc       willing    rw     off       --        on        on,off
```

For more information, see “Setting the PFC TLV Units” on page 129.

Displaying PFC Mapping Information Between Host and Peer

The `PFC Pending` value returns a `True` status if PFC information between the host and the peer does not converge. After the mismatch is resolved, the status of `PFC Pending` returns to `False`.

The following command alerts you to a mismatch of PFC mapping information between the local host and the peer.

```
$ lldpadm show-agent -lv -o "PFC Pending" agent
```

```
$ lldpadm show-agent -lv -o "PFC Pending" agent
```

EXAMPLE 43 Verifying Symmetry of PFC Information Between Host and Peer

The following example shows how to verify in actual running time whether PFC information is synchronized between the host and peer, or whether a mismatch occurs.

```
$ lldpadm show-agent -lv -o "PFC Pending" net0
PFC Pending: True
```

To display all the information that the agent advertises, use the `-v` (verbose) option of the `lldpadm show-agent` command:

```
$ lldpadm show-agent -v net0
```

Displaying Priority Definitions

The following command displays PFC information about the physical link with regards to enabled priorities on the NIC:

```
$ dladm show-phys -D pfc datalink
```

EXAMPLE 44 Displaying CoS Priority Definitions

This example shows how to display the current priority definitions on a specific physical link based on the value of the `pfcmap` property. For example, assume that `pfcmap` is configured as

01000000. To display the corresponding priority mappings on the physical link, you would proceed as follows:

```
$ dladm show-phys -D pfc net0
LINK  COS    PFC   PFC_EFFECT   CLIENTS
net0   0     YES   NO           net0,vnic1
       1     YES   YES          vnic2
       2     YES   NO           vnic3
       3     YES   NO           vnic4
       4     YES   NO           vnic5
       5     YES   NO           vnic6
       6     YES   NO           vnic7
       7     YES   NO           vnic8
```

For the physical link net0, priority is enabled for all VNIC clients configured over the datalink. However, the local host adjusts its PFC mapping to the PFC mapping on the peer, as shown by the values of the PFC_EFFECT field, where priority is disabled on COS 0 and 2-7. As a result, no PFC frames would be exchanged for traffic on any VNIC except vnic2 regardless of the availability of resources. With this configuration, packet drops are allowed on traffic that flows on all VNICs except vnic2. For traffic on vnic2, PFC PAUSE frames are sent when traffic congestion occurs to prevent packet loss on this client.

Application Priority Configurations

DCBX exchanges the priority information that is associated with the application. Application TLV units that are exchanged through DCBX contain information about the priority to be used for an application on the host. The priority is defined in the Application Priority Table. Each entry in the table contains the name of the application and the priority assigned to the application. When you set a priority for an application, all the DCB settings of the PFC and ETS of that priority are applicable to the application. The application TLV uses the table data for exchanging application priority information with the other hosts.

By default, the application feature accepts priority mappings from the peer. The willing property of the application TLV appln is similar to PFC and enables information exchange between the peers.

Entries on the table uses the following format:

protocol-ID/*selector*/*priority*

The pair *protocol-ID*/*selector* identifies the application. The priority for a corresponding application is identified by a priority that contains a value from 0 to 7.

To exchange this information about the priority of an application with other hosts, you set an application TLV as follows:

```
$ lldpadm set-agenttlvprop -p property=value -a agent appln
```

For example, for FCoE traffic, the protocol ID is `0x8906` and the selector ID is `1`. Suppose that the priority `4` is assigned to this application. Based on Table 3, “Per-Agent TLV Units and Their Properties,” on page 106 that lists the parameters for setting an application TLV you would type the following command:

```
$ lldpadm set-agenttlvprop -p apt=8906/1/4 -a net0 appln
$ lldpadm show-agenttlvprop -a net0 appln -p apt
AGENT   TLVNAME   PROPERTY   PERM   VALUE      EFFECTIVE    DEFAULT   POSSIBLE
net0    appln     apt        rw     8906/1/4   8906/1/4     --        --
```

Customizing Enhanced Transmission Selection for DCB

The default configuration is automatically set up when LLDP is enabled and DCB is supported by the underlying link. In this default configuration, the `cos` value `0` is assigned all of the bandwidth. However, you can use the `dladm set-linkprop` command to configure the `cos` values on a datalink to assign part of the bandwidth to that datalink.

ETS configuration and recommendation TLVs are enabled by default for a NIC. For a list of possible values for `dot1-tlv`, refer to Table 1, “Optional TLV Units for an LLDP Agent,” on page 105.

If you want to remove the `pfc` TLV, type the following command:

```
$ lldpadm set-agenttlvprop -p dot1-tlv-=pfc agent
```

Setting the ETS-Related Datalink Properties

The properties of datalinks that refer to PFC information apply to the prevention of packet loss based on the priorities defined for the packets. The ETS properties relate to assigning shares of the underlying link's bandwidth based on priorities.

DCB provides the following ETS-related properties:

- `cos` – Specifies the class of service or priority of the datalink. The value of this property ranges from `0` to `7`. The default value is `0`. The `cos` value is set in the VLAN tag of the packets that are transmitted over this link.

- `ets-bw-local` – Indicates the ETS bandwidth that is allocated on the transmit (Tx) side for the datalink. This property is configurable only if the underlying physical NIC has DCB capabilities and supports ETS and the link's `cos` property is not set to `0`. You set a value for this property on a datalink by specifying the percentage of total bandwidth of the underlying physical link. The sum of the values for the `ets-bw-local` property for all the datalinks over the same physical NIC must not exceed 100%.

  The bandwidth percentage that is defined on `ets-bw-local` is not reserved only for that datalink. If the allocated bandwidth is not used, then it can be used by other datalinks on that physical NIC. Further, the bandwidth allocation is enforced only on the transmission side of the host's traffic.
- `ets-bw-remote-advice` – Specifies the recommended ETS bandwidth value sent to the peer. By default, the locally configured value of the `ets-bw-local` property is recommended to the peer. However, you can recommend a value that is different from the `ets-bw-local` property by explicitly configuring the `ets-bw-remote-advice` datalink property.

  Configuring the `ets-bw-remote-advice` property is useful if the bandwidth assignment for a datalink is asymmetrical, which means that the receive (Rx) and transmit (Tx) bandwidth are different. When you explicitly set the `ets-bw-remote-advice` property, transmission of the ETS recommendation DCBX TLV starts automatically.
- `ets-bw-local-advice` – Specifies the recommended bandwidth share for the datalink, which is sent by the peer to the local host. This is a read-only property.
- `ets-bw-remote` – Specifies the bandwidth share that is configured on the peer for the datalink. This is a read-only property.

To set the priority and to allocate a bandwidth to the VNIC, use the following commands:

- To set the priority to the VNIC:

  ```
  $ dladm set-linkprop -p cos=value  VNIC
  ```
- To allocate a percentage of the bandwidth of the underlying physical link to a VNIC:

  ```
  $ dladm set-linkprop -p ets-bw-local=value  VNIC
  ```

  The value that you assign to the `ets-bw-local` property represents a percentage of the total bandwidth capacity of the underlying link. The sum of all the allocated bandwidth values that you assign to the clients must not exceed 100 percent.
- To explicitly recommend a bandwidth that is sent to the peer:

  ```
  $ dladm set-linkprop -p ets-bw-remote-advice=value  VNIC
  ```

You can determine the actual bandwidth share that is implemented on the local host's datalink and the bandwidth share that is configured on the peer's datalink by using the `dladm show-linkprop` command. The value in the `EFFECTIVE` field of the output for the `ets-bw-local`

and `ets-bw-remote` properties shows the actual bandwidth share implemented. For more information, see "Displaying ETS Configuration Information" on page 138.

For the appropriate bandwidth to be used for packets with specific priorities, symmetric or synchronized ETS information between the communicating hosts is preferable. Specifically, should be the local system that is able to adjust its bandwidth share to the value of `ets-bw-local-advice`. An Oracle Solaris system can automatically adjust its ETS configurations to match the ETS recommendation from the peer.

Setting ETS TLV Units

The ETS TLV (`etscfg`) configuration determines how the host responds to ETS recommendations from the peer. This TLV unit has only one configurable property, `willing`. By default, this property is set to `on` and enables the local host to synchronize its ETS configuration with the ETS recommendation of the remote peer.

To verify that the host can synchronize its ETS information with the ETS information of the remote peer, use the following command:

```
$ lldpadm show-agenttlvprop -p willing -a agent etscfg
```

If the `willing` property is set to `off`, type the following command to establish synchronization:

```
$ lldpadm set-agenttlvprop -p willing=on -a agent etscfg
```

To prevent synchronization of information for a specific agent, set the `willing` property to `off` as follows:

```
$ lldpadm set-agenttlvprop -p willing=off -a agent etscfg
```

where *agent* is the datalink on which the agent is enabled.

Recommending ETS Configuration to the Peer

The configured ETS bandwidth values (`ets-bw-local`) for each priority can be recommended to the peer so that the peer can configure the same values. You must enable the `etsreco` property in the `dot1-tlv` type of the LLDP agent on the NIC to recommend the ETS bandwidth values. The recommended values can be the same as the locally configured ETS values or you can also explicitly configure the recommended values by setting the new datalink property

`ets-bw-remote-advice` by using the `dladm set-linkprop` command. Configuring the `ets-bw-remote-advice` property is useful if the assigned bandwidth for a datalink is asymmetrical, which means that the receive (Rx) and transmit (Tx) bandwidth are different.

By default, the configured values of the `ets-bw-local` property are used to recommend values to the peer. However, you can recommend a different ETS value by setting a different value for the `ets-bw-remote-advice` property. For example, if the network traffic is more on the Tx, then you can configure a higher ETS value for the `ets-bw-local` property (Tx on the host) and a lower value configured for the `ets-bw-remote-advice` property (Rx to the host).

EXAMPLE 45 Recommending an ETS Configuration to the Peer

1. Ensure that the `etsreco` property is enabled by displaying the `dot1-tlv` type property of the LLDP agent for `net5`.

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

2. Allocate a share of 20% of the underlying link's bandwidth for `vnic1`.

```
$ dladm set-linkprop -p ets-bw-local=20 vnic1
$ dladm show-linkprop -p ets-bw-local vnic1
LINK     PROPERTY        PERM VALUE        EFFECTIVE    DEFAULT   POSSIBLE
vnic1    ets-bw-local    rw   20           20           0         --
```

   By default, the same value is recommended for the peer.

```
$ dladm show-linkprop -p ets-bw-remote-advice vnic1
LINK     PROPERTY             PERM VALUE  EFFECTIVE    DEFAULT   POSSIBLE
vnic1    ets-bw-remote-advice rw   --      20           0         --
```

3. Display the information exchanged by LLDP.

```
$ lldpadm show-agent -l -v net5
```

4. Display the bandwidth recommended to the peer.

```
$ dladm show-phys -D ets -r net5
LINK      COS ETSBW_RMT_EFFECT ETSBW_RMT_ADVICE CLIENTS
--         0   0               80               net5
```

```
1   0               0               --
2   0               0               --
3   0               20              vnic1
4   0               0               --
5   0               0               --
6   0               0               --
7   0               0               --
```

By default, the values configured for the `ets-bw-local` property for each priority on `net5` are sent as the recommended values to the peer. The `ETSBW_RMT_ADVICE` shows the values recommended to the peer. The output also shows that the peer has not configured any ETS bandwidth on its end. You can also display bandwidth recommended to the peer by using the `lldpadm show-agent` command.

5. Recommend a different value to the peer.

```
$ dladm set-linkprop -p ets-bw-remote-advice=10 vnic1
$ dladm show-linkprop -p ets-bw-remote-advice vnic1
LINK     PROPERTY              PERM VALUE     EFFECTIVE    DEFAULT   POSSIBLE
vnic1    ets-bw-remote-advice  rw   10         10           0         --
```

6. Display the bandwidth recommended to the peer.

```
$ dladm show-phys -D ets -r net5
LINK   COS ETSBW_RMT_EFFECT ETSBW_RMT_ADVICE CLIENTS
--     0   0                90               net5
       1   0                0                --
       2   0                0                --
       3   0                10               vnic2
       4   0                0                --
       5   0                0                --
       6   0                0                --
       7   0                0                --
```

The `ETSBW_RMT_EFFECT` field shows value `0` for `vnic2`, which indicates that the peer has not set any bandwidth on its end, even though you have recommended bandwidth values. This situation means that peer might not have enabled LLDP or does not support ETS.

Displaying ETS Configuration Information

You can use the following commands to display information about the ETS configuration:

- `$ dladm show-linkprop -p ets-bw-local,ets-bw-remote,ets-bw-local-advice,ets-bw-remote-advice` *datalink*

This command displays the information related to ETS on a physical link.

- $ **dladm show-phys -D ets** *phys-link*

  This command displays the local and remote ETS configuration on the physical link with regard to bandwidth allocation and distribution across the link.

- $ **lldpadm show-agenttlvprop -a** *agent* **etscfg**

  agent is the datalink on which LLDP is enabled. This command displays the ETS TLV property that controls the capability of a host to synchronize ETS information with a peer.

EXAMPLE 46 Displaying ETS-Related Datalink Properties

This example shows how to display the status of datalink properties that are related to ETS before synchronization is enabled.

```
$ dladm show-linkprop -p cos,ets-bw-local,ets-bw-remote,ets-bw-local-advice, \
ets-bw-remote-advice vnic1
LINK    PROPERTY             PERM  VALUE  EFFECTIVE  DEFAULT    POSSIBLE
vnic1   cos                  rw    2      2          0          0-7
vnic1   ets-bw-local         rw    10     10         0          --
vnic1   ets-bw-remote        r-    20     20         --         --
vnic1   ets-bw-local-advice  r-    20     20         --         --
vnic1   ets-bw-remote-advice rw    10     10         0          --
```

The output shows that the host has set and recommended an ETS value of 10% for `vnic1` with a `cos` value of `2`. However, the peer has set and recommended an ETS value of 20% with a `cos` value of `2` for `vnic1`. Because the synchronization is not enabled (`willing` is not enabled) the host has not accepted the peer's recommendation, which is reflected in the `EFFECTIVE` value of the `ets-bw-local` property (locally configured value).

EXAMPLE 47 Displaying the Capability of the Local Host to Synchronize ETS Information

This example shows how to display the current status of the local host's ability to adjust to the ETS configurations of the peer.

```
$ lldpadm show-agenttlvprop -a net0 etscfg
AGENT   TLVNAME   PROPERTY   PERM   VALUE   EFFECTIVE   DEFAULT   POSSIBLE
net0    etscfg    willing    rw     off       --        on        on,off
```

To enable synchronization, type the following commands:

```
$ lldpadm set-agenttlvprop -p willing=on -a net0 etscfg
```

```
$ dladm show-linkprop -p  cos,ets-bw-local,ets-bw-remote, \
ets-bw-local-advice,ets-bw-remote-advice vnic1
LINK    PROPERTY             PERM  VALUE   EFFECTIVE  DEFAULT    POSSIBLE
vnic1   cos                  rw    2       2          0          0-7
vnic1   ets-bw-local         rw    10      20         0          --
vnic1   ets-bw-remote        r-    20      20         --         --
vnic1   ets-bw-local-advice  r-    20      20         --         --
vnic1   ets-bw-remote-advice rw    10      10         0          --
```

Because the synchronization is enabled (the property `willing` is enabled), the host has accepted the peer's recommendation, which is reflected in the `EFFECTIVE` value of `ets-bw-local`.

The following example shows effective ETS values on the host and the peer for each priority value on the physical link.

```
$ dladm show-phys -D ets net4
LINK     COS ETSBW_LCL_EFFECT ETSBW_RMT_EFFECT ETSBW_LCL_SOURCE CLIENTS
net4      0   0                30               local            net4
          1   0                0                local            --
          2   0                0                local            --
          3   0                0                local            --
          4   0                70               local            --
          5   0                0                local            --
          6   0                0                local            --
          7   0                0                local            --
```

`ETSBW_LCL_EFFECT` Displays the effective ETS bandwidth as a percentage for the priority.

`ETSBW_RMT_EFFECT` Displays the effective ETS bandwidth as a percentage for the priority value on the peer.

`ETSBW_LCL_SOURCE` Indicates the source for the `ETSBW_LCL_EFFECT` value. This value could be either `local`, which is the configured value, or `remote`, which is the recommended value.

The following example shows the local ETS information including the locally configured values, local effective values, and the values recommended by the peer.

```
$ dladm show-phys -D ets -l net5
LINK                COS ETSBW_LCL ETSBW_LCL_EFFECT ETSBW_LCL_ADVICE CLIENTS
--                  0   80        80               0                net5
                    1   0         0                0                --
                    2   0         0                0                --
                    3   20        20               0                vnic2
                    4   0         0                0                --
                    5   0         0                0                --
                    6   0         0                0                --
```

```
                  7   0         0               0               --
```

Because the peer has not recommended any values, the local effective value (`ETSBW_LCL_EFFECT`) is set by using the local configured value (`ETSBW_LCL`).

The following example displays information about the peer.

```
$ dladm show-phys -D ets -r net5
LINK                COS ETSBW_RMT_EFFECT ETSBW_RMT_ADVICE CLIENTS
--                  0   0                20               net5
                    1   0                0                --
                    2   0                0                --
                    3   0                80               vnic2
                    4   0                0                --
                    5   0                0                --
                    6   0                0                --
                    7   0                0                --
```

The output shows that the remote peer does not have any value set in the `ETSBW_RMT_EFFECT` field even though the host had recommended the peer to set 80% for priority `3`.

♦ ♦ ♦ APPENDIX A

Comparison: Trunk Aggregations and DLMP

This table presents a general comparison of the two types of link aggregation.

Feature	Trunk Aggregations	DLMP Aggregations
Link-based failure detection	Supported	Supported
LACP	Supported	Not supported
Use of standby interfaces	Not supported	Not supported[1]
Span multiple switches	Not supported unless using vendor proprietary solution	Supported
Switch configuration	Required	Not required
Policies for load balancing	Supported	Not applicable
Load spreading across all of the aggregation's ports	Supported	Limited[2]
User-defined flows for resource management	Supported	Supported
Link protection	Supported	Supported
Back-to-back configuration	Supported	Not supported[3]

[1] Each DLMP client is associated with exactly one DLMP port. The remaining ports act as available ports for the DLMP clients, but you cannot configure these available ports.

[2] The aggregation spreads its VNICs across all ports. However, individual VNICs cannot spread the load on multiple ports.

[3] DLMP aggregations must always use an intermediary switch to send packets to other destination systems. However, no switch configuration is required for DLMP.

♦ ♦ ♦ APPENDIX B

Link Aggregations and IPMP: Feature Comparison

Link aggregation and IPMP are different technologies that achieve improved network performance and maintain network availability.

The following table presents a general comparison between link aggregation and IPMP.

Feature	Link Aggregation	IPMP
Network technology type	Layer 2 (link layer).	Layer 3 (IP layer).
Configuration tool	`dladm`	`ipadm`
Link-based failure detection	Supported.	Supported.
Probe-based failure detection	Trunk: Based on LACP, targeting the immediate peer host or switch. DLMP: Supported. ICMP-based, targeting any defined systems in the same subnet as DLMP addresses, across multiple levels of intervening Layer 2 switches.	ICMP-based, targeting any defined system in the same IP subnet as test addresses across multiple levels of intervening Layer 2 switches.
Use of standby interfaces	Trunk: Not supported. DLMP: Not supported.	Supported. Standby interfaces can be configured.
Span multiple switches	Trunk: Supported. However, requires switch vendor extensions. DLMP: Supported.	Supported.
Switch configuration	Trunk: Required. DLMP: Not required.	Not required.
Back-to-back configuration	Trunk: Supported. DLMP: Not supported.	Not supported.
Media types supported	Ethernet-specific.	Broadcast-capable.

Feature	Link Aggregation	IPMP
Load-spreading support	Trunk: Supported and controlled by the administrator by using the `dladm` command. Inbound load spreading is supported. DLMP: Supported across clients and VNICs of the aggregation. However, load spreading by individual clients and VNICs over the aggregation is not supported.	Supported. Controlled by the kernel. Inbound load spreading is indirectly affected by the source address selection.
Level of support when integrating with VNICs	Excellent support. Aggregation is configured in the control domain or the global zone only and is transparent to the zones.	Supported. However, VNIC properties such as bandwidth limit, dedicated Rx or Tx rings, and link protection cannot be enforced on an IPMP group. Requires multiple VNICs to be assigned to the zones and needs to be configured in every zone.
User defined flows for resource management	Supported.	Not supported.
Link protection	Supported.	Not supported.
Protocol requirements	None.	None.

In link aggregations, incoming traffic is spread over the multiple links that comprise the aggregation in trunk mode. Therefore, networking performance is enhanced as more NICs are installed to add links to the aggregation.

IPMP's traffic uses the IPMP interface's data addresses as they are bound to the available active interfaces. If, for example, all the data traffic is flowing between only two IP addresses but not necessarily over the same connection, then adding more NICs will not improve performance with IPMP because only two IP addresses remain usable.

♦ ♦ ♦ APPENDIX C

Packet Format of Transitive Probes

The transitive probe packet is a proprietary protocol packet with an Ethernet type, `ETHERTYPE_ORCL` (`0x881b`). For more information about the transitive probes, see "Probe-Based Failure Detection" on page 28. For an example to display the statistics of the probes, see Example 8, "Displaying Probe-Related Information," on page 46. The following figure shows the transitive probe packet format.

FIGURE 17 Transitive Probe Packet

Field	Description
`Dest MAC`	Destination MAC address.
`Src MAC`	Source MAC address.
`Proto#`	Protocol number. The Layer 2 payload of `ETHERTYPE_ORCL` packets must start with a 16-bit protocol number, which is 1 for the transitive probe packet.

Field	Description
`Typ`	Probe packet type. The probe packet type is `0` for request and `1` for response.
`Pad`	Padding (all-zero).
`Seq#`	Probe sequence number.
`Timestamp`	Probe timestamp.
`TL`	Target information length. For Ethernet, target length is 6 bits (Ethernet MAC address length).
`Target`	Target port MAC address.
`SL`	Source information length. For Ethernet, source length is 6 bits.
`Source`	Source port MAC address.

Index

A

adding
  links to an external bridge, 92
administering VLANs on bridged networks, 97
application priority configurations, 133
application TLV units, 133, 133
  See also PFC
assigning a PVLAN to a zone, 80
`auto-enable-agents`, 107

B

`basic-tlv`, 105
bridged network ring, 86
bridged networks, 83, 83
  adding links to an existing bridge, 92
  administering VLANs on bridges, 97
  bridged network ring, 86
  creating a bridge, 90
  debugging bridges, 99
  deleting a bridge, 96
  displaying configuration information, 94
  displaying configuration information about bridge links, 96
  example of creating a bridge, 91
  example of deleting a bridge from the system, 97
  example of displaying bridge information, 95
  example of modifying the protection type for a bridge, 92
  how a bridge network works, 87
  modifying the protection type, 92
  network stack, 85
  overview, 83
  protocols, 88
  removing links, 93
  setting link properties, 93
  simple bridged network, 84
  STP daemon, 89
  TRILL daemon, 90
  VLANs and STP and TRILL protocols, 98
bridges
  adding links to an existing bridge, 92
  configuring a VLAN over a link that is part of a bridge, 97
  creating, 90
  deleting, 96
  naming bridges, 91
  removing links from, 93
bridging protocols, 88

C

class of service *See* CoS
community VLAN, 74
configuring
  PVLAN, 79
CoS
  priority definitions, 123
creating
  bridges, 90
  link aggregations, 33
  PVLAN, 79
  VLAN over a link that is part of a bridge, 97
  VLANs, 55
  VLANs over a link aggregation, 61
customizing
  PFC for DCB, 129

D

data center bridging *See* DCB
datalink multipathing aggregations *See* DLMP aggregations
DCB, 121
  configuring ETS, 134
  considerations, 123
  `cos` property, 122
  customizing PFC, 128
  enabling DCBX, 125
  enhanced transmission selection (ETS), 122
  overview, 121
  priority-based flow control (PFC), 122, 123
DCBX protocol, 101, 121
defining
  LLDP TLV, 113
deleting
  bridge, 96
  PVLAN, 80
  VLANs, 67
disabling
  LLDP, 115
displaying
  aggregated port information, 46
  bridge configuration information, 94
  datalink properties, 130, 139
  ETS configuration information, 138
  IP address state of aggregation, 47
  LLDP advertised information, 116
  LLDP statistics, 119
  PFC configuration information, 130
  PFC mapping information, 132
  PFC synchronization status, 131
  priority definitions, 132
  `probe-ip` property values, 47
  probe-related information, 46
  state of the aggregated port, 46
  synchronization status, 139
  VLAN information, 63
  `willing` property value, 131, 139
`dladm` command
  `add-aggr`, 40
  `add-bridge`, 93
  `create-aggr`, 34
  `create-bridge`, 90
  `create-vlan`, 55, 79
  `delete-aggr`, 47
  `delete-bridge`, 96
  `delete-vlan`, 67, 80
  `modify-aggr`, 41
  `modify-bridge`, 92
  `modify-vlan`, 64, 79
  `remove-bridge`, 93
  `show-aggr`, 34
  `show-bridge`, 94
  `show-linkprop`, 130
  `show-vlan`, 63
DLMP aggregations, 24
  configuring probe-based failure detection, 42
  example of configuring probe-based failure detection, 45
  example of creating DLMP aggregation, 36
  failure detection, 26
  how DLMP aggregation works, 25
  link-based failure detection, 27
  monitoring probe-based failure detection, 45
  port failure, 27
  probe-based failure detection, 28
  topology, 25
`dot1-tlv`, 105
`dot3-tlv`, 105

E

enabling
  DCBX, 125
  LLDP for specific ports, 109
  LLDP globally, 108
  synchronization of PFC information, 130
enhanced transmission selection *See* ETS
ETS, 122, 124
  bandwidth share, 134
  configuring, 134
  displaying information, 138
  ETS TLV units, 136

example of displaying ETS-related datalink properties, 139
example of displaying the capability to synchronize ETS information, 139
example of recommending ETS configuration to the peer, 137
local and remote information, 134
properties, 134
recommending ETS configuration to the peer, 136
setting ETS-related datalink properties, 135
examples
creating a VLAN, 56
creating multiple VLANs over a link aggregation, 62
deleting a VLAN configuration, 67, 80
migrating multiple VLANs, 66

F

failure detection in DLMP aggregation, 26
link-based failure detection, 27
probe-based failure detection, 28

G

global TLV units, 106

H

high-availability
DLMP aggregations, 24

I

ICMP probing, 28
installing
LLDP package, 108
IPMP
link aggregations, comparison with, 145
isolated VLAN, 73

L

LACP
definition of, 23
LACPDU, 24
modes, 24
using a switch with, 23
link aggregation
example of adding a link to an aggregation, 40
example of deleting a link aggregation, 48
Link Aggregation Control Protocol (LACP) *See* LACP
link aggregations, 19
adding datalinks, 40
combined use with VLANs, 68
creating, 34
deleting, 47
deployment possibilities, 19
DLMP aggregations, 24
example of removing a link from an aggregation, 41
IPMP, comparison with, 145
overview, 19
removing, 41
requirements, 31
trunk aggregations, 21
Link Layer Discovery Protocol *See* LLDP
link layer discovery protocol *See* LLDP
link properties
setting for a bridge, 93
link state notification, 32
link-based failure detection, 27
LLDP, 101, 101
agent modes, 103
agents, 103
`auto-enable-agents`, 107
components in Oracle Solaris, 102
defining TLV values, 113
disabling, 115
displaying advertised information, 116
displaying statistics, 119
enabling, 107
enabling for specific ports, 109
enabling globally, 108
example of adding optional TLV units, 112

example of customizing the auto-enable-agents SMF property, 110
example of defining TLV values, 114
example of displaying advertised information, 116
example of displaying selected statistics, 120
example of displaying statistics, 119
example of displaying the chassis ID and port ID, 104
example of enabling LLDP on multiple datalinks, 111
global TLV units, 103, 105
installing, 108
lldpd daemon, 103
management information base (MIB), 103
monitoring agents, 116
optional TLV units, 105
package, 102
per-agent TLV units, 103, 105
SMF property for, 107
specifying agent TLV units, 112
TLV units, 104, 105
LLDP SMF service, 102
lldpadm command, 102
reset-agentprop, 115
set-agentprop, 109, 115
set-agenttlvprop, 110, 113, 128, 129, 133
set-tlvprop, 109, 113
show-agent, 116, 119
show-agenttlvprop, 113, 130
show-tlvprop, 113
LLDPDUs, 103
load balancing
trunk aggregations, 24
local MIB, 103

M

management information base (MIB), 103
managing network datalinks
bridged networks, 83
DCB, 121
features and components, 15
introduction, 15
link aggregations, 19
LLDP, 101
VLANs, 51
mandatory TLV units, 104
migrating
VLAN, 65
mode of operation for DCB
setting, 126
modifying
protection type of a bridge, 92
PVLAN, 79
trunk aggregation, 41
VLAN ID of a VLAN, 64
monitoring
LLDP agents, 116

N

Network Management profile, 16
network stack
bridge implementation, 85

O

optional TLV units, 105

P

PAUSE frames, 123
per-agent TLV units, 106
pfbash shell, 16
PFC, 122, 123
CoS priority mappings, 128
customizing, 128
customizing PFC for DCB, 129
datalink properties, related, 128
displaying datalink properties, 130
displaying information, 130
example of displaying capability to synchronize PFC information, 131
example of displaying CoS priority definitions, 132

example of displaying PFC-related datalink properties, 131
example of enabling synchronization between host and the peer, 130
example of verifying symmetry of PFC information, 132
local and remote information, 128
PAUSE frames, 123
`pfcmap`, 123, 128
`pfcmap-remote`, 128
synchronized information, 128
VNIC clients, 133
PFC mapping, 123
PFC TLV units, 129
priority-based flow control *See* PFC
private virtual local area network, 73
private VLAN
configuring, 79
deleting, 80
modifying, 79
private VLANs, 73
privileges, network configuration, 16
probe-based failure detection, 28
configuring, 42
displaying aggregated port information, 46
displaying `probe-ip` property values, 47
displaying probe-related information, 46
displaying the IP address state of aggregation, 47
displaying the state of the aggregated port, 46
example of configuring probe-based failure detection, 45
ICMP probing, 28
monitoring, 45
transitive probing, 28
promiscuous trunk port, 76
protocol data units (PDUs), 103
protocols
DCBX, 121
LLDP, 101
STP, 86, 88
TRILL, 86, 88
PVLAN
assigning to a zone, 80
community VLAN, 74
configuration requirements, 77
configuring, 79
deleting, 80
isolated VLAN, 73
modifying, 79
overview, 73
ports, 76
promiscuous trunk port, 76
secondary trunk port, 76
with zone, 77
PVLAN ports, 76
PVLAN secondary trunk port, 76
PVLANs, 73

R

RBAC, 16
recommending
ETS configuration to the peer, 136
remote MIB, 103
removing
link from an aggregation, 41
links from a bridge, 93

S

setting
ETS TLV units, 136
ETS-related datalink properties, 134
mode of operation for DCB, 126
PFC TLV units, 129
PFC-related datalink properties, 128
tag-mode, 76
simple bridged network, 84
specifying
TLV units for LLDP packet of an agent, 112
STP
setting as bridge protection type, 92
STP daemon, 89
STP protocol, 88
contrasted with TRILL, 89

T

tag mode, 76
TLV property
  `willing`, 129
topology discovery
  with LLDP, 102
transitive probing, 28
TRILL, 88
  setting as bridge protection type, 92
TRILL daemon, 90
TRILL protocol
  contrasted with STP, 89
trunk aggregations, 21
  back-to-back, 23
  example of creating trunk aggregation, 35
  example of modifying a trunk aggregation, 42
  Link Aggregation Control Protocol (LACP), 23
  load balancing policy, 24
  modifying, 41
  prerequisites, 34
  unique features, 21
  using a switch, 22
  when to use, 21
type -length-value (TLV) units, 104

V

`virt-tlv`, 105
virtual local area networks *See* VLANs
VLANs, 51, 51
  combined use with link aggregations, 68
  configuration, 55
  creating over link aggregations, 61
  deleting, 67
  displaying information, 63
  example to create a VLAN, 56
  example to create a VLAN with zones, 59, 59
  example to create multiple VLANs over a link aggregation, 62
  example to delete a VLAN configuration, 67, 80
  MAC addresses on, 65
  migrating, 64
  modifying VLAN IDs, 64
  overview, 51
  planning a VLAN configuration, 54
  STP and TRILL protocols, 98
  topologies, 52
  used with zones, 57
  VLAN names, 52
  when to use VLANs, 52
  workgroups, 52