Troubleshooting Network Administration Issues in Oracle® Solaris 11.4

Part No: E60994
November 2020

Troubleshooting Network Administration Issues in Oracle Solaris 11.4

Part No: E60994

License Restrictions Warranty/Consequential Damages Disclaimer

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

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.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Trademark Notice

Third-Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Pre-General Availability Draft Label and Publication Date

Pre-General Availability: 2020-01-15

Pre-General Availability Draft Documentation Notice

If this document is in public or private pre-General Availability status:

This documentation is in pre-General Availability status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

Oracle Confidential Label

ORACLE CONFIDENTIAL. For authorized use only. Do not distribute to third parties.

Revenue Recognition Notice

If this document is in private pre-General Availability status:

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your pre-General Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

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: E60994

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

- **Overview** – Describes tasks for troubleshooting network configuration issues in the Oracle Solaris operating system (OS).
- **Audience** – System administrators.
- **Required knowledge** – Understanding of basic and advanced network administration concepts and practices.

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

Troubleshooting General Network Administration Issues

This chapter describes how to troubleshoot various problems that can occur on a network, including issues with network configuration, network connectivity, and various error conditions.

This chapter contains the following topics:

For information about configuring and managing the network, see *Configuring and Managing Network Components in Oracle Solaris 11.4*.

For information about administering an existing TCP/IP network, see *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

For a quick reference of the most commonly used networking commands, see *Oracle Solaris 11.4 Network Administration Cheatsheet*.

What's New in Troubleshooting Network Administration Issues

You can use the Network Conditions Simulator (NCS) to simulate network operating conditions, for example, propagation delay, bandwidth, drop rate, packet reordering, and corruption. See "Simulating Network Operating Conditions Within a Test Environment" on page 23 for more information.

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.

Answers to Common Network Administration Questions

Refer to the following information when troubleshooting general network administration issues.

Question: I configured my system during an installation but my system's network is still not configured correctly after the installation. What should I do?

Answer: The answer depends on which network component is not configured correctly. You use the `dladm` and `ipadm` commands to configure the network. Given the type of configuration parameters that can be set at installation time, most likely an IP interface or address is not configured correctly.

To determine which network components need to be reconfigured, start by displaying the current network configuration as follows:

```
# ipadm
```

If the IP address is incorrect, you will need to delete that address and then create the correct IP address, for example, a static IP address or a DHCP address.

The following example shows how to delete the IPv6 `addrconf` portion of an IP configuration. In this example the IPv6 `addrconf` address is determined by running the `ipadm` command as follows:

```
# ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
lo0               loopback   ok           --         --
lo0/v4            static     ok           --         127.0.0.1/8
lo0/v6            static     ok           --         ::1/128
net0              ip         ok           --         --
net0/v4           dhcp       ok           --         203.0.113.10/24
net0/v6           addrconf   ok           --         fe80::8:20ff:fe90:10df/10
# ipadm delete-addr net2/v6
# ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
lo0               loopback   ok           --         --
lo0/v4            static     ok           --         127.0.0.1/8
lo0/v6            static     ok           --         ::1/128
net0              ip         ok           --         --
net0/v4           dhcp       ok           --         203.0.113.10/24
```

See Chapter 3, "Configuring and Administering IP Interfaces and Addresses in Oracle Solaris" in *Configuring and Managing Network Components in Oracle Solaris 11.4* for complete instructions.

Question: How do I configure a persistent default route on my system?

Answer: Because the `/etc/defaultrouter` file has been removed in this Oracle Solaris release, you can no longer manage default routes by editing this file. Also, after a fresh installation, you can no longer check this file to determine the system's default route.

- Display routes that are created persistently as follows:

  ```
  # route -p show
  ```

- Add a persistent default route as follows:

  ```
  # route -p add default gateway
  ```

- Display the currently active routes on a system as follows:

  ```
  # netstat -rn
  ```

See "Creating Persistent (Static) Routes" in *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer* for instructions.

Question: How do I display the MAC address of a system?

Answer: Display the MAC addresses of the physical links that are in a system as follows:

```
# dladm show-phys -m
```

The output of this command is similar to the output of the `ifconfig` command that was used in previous releases. See the `ifconfig(7)` man page.

Display the MAC addresses of all of the links that are in a system (physical and non-physical) as follows:

```
# dladm show-linkprop -p mac-address
```

Question: I can no longer use the `dladm show-dev` command to display the physical links that are in my system? What command do I use now?

Answer: Use the `dladm show-phys` command as follows:

```
# dladm show-phys
LINK              MEDIA                STATE      SPEED  DUPLEX    DEVICE
net0              Ethernet             up         0      unknown   vnet0
```

Question: How do I display the mapping between link names, devices, and locations on a system?

Answer: Use the `dladm show-phys` command with the `-L` option as follows:

```
# dladm show-phys -L
LINK         DEVICE        LOCATION
net0         e1000g0       MB
net1         e1000g1       MB
net2         e1000g2       MB
net3         e1000g3       MB
net4         ibp0          MB/RISER0/PCIE0/PORT1
net5         ibp1          MB/RISER0/PCIE0/PORT2
net6         eoib2         MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
net7         eoib4         MB/RISER0/PCIE0/PORT2/cloud-nm2gw-2/1A-ETH-2
```

Question: What command do I use to determine the maximum transmission unit (MTU) range that my system supports?

Answer: Use the `ipadm show-ifprop` command to determine this information. The last column of the output displays the supported MTU ranges.

```
# ipadm show-ifprop -p mtu interface
```

Question: What if the naming services settings on my system are lost or not configured correctly after an installation?

Answer: The naming services configuration should be what you specified during the installation. In this release, naming services are configured through the Service Management Facility (SMF). See Chapter 4, “Administering Naming and Directory

Services on an Oracle Solaris System" in *Configuring and Managing Network Components in Oracle Solaris 11.4* for instructions on how to configure naming services and how to import naming services configuration on a client system, if necessary, after an installation.

Question: How can I start over and reconfigure all of the network settings for my system?

Answer: You can unconfigure and reconfigure an Oracle Solaris instance, including its network and naming service settings, as follows:

```
# sysconfig unconfigure -g network,naming_services
```

Question: What is the difference between creating a virtual local area network (VLAN) with the `dladm create-vlan` command and a virtual NIC (VNIC) with the `dladm create-vnic -v VID ...` command? Also, what are the unique features of both commands that would dictate the use of one rather than the other?

Answer: Depending on your networking needs and what you are trying to accomplish, you would use each of these features for different purposes.

A VLAN is a subdivision of a LAN at the datalink layer (L2) of the network stack. VLANs enable you to divide your network into subnetworks without having to add to the physical network environment. So, the subnetworks are virtual and they share the same physical network resources. VLANs facilitate network administration by using smaller groups that are easier to maintain.

VNICs are virtual network devices that use the same datalink interface as a physical network interface card (NIC). You configure VNICs over an underlying datalink. When VNICs are configured, they behave like physical NICs. Depending on the network interface that is in use, you can explicitly assign a MAC address to a VNIC other than the default address.

For more information using network administration features to obtain a desired result, see Chapter 1, "Summary of Oracle Solaris Network Administration" in *Strategies for Network Administration in Oracle Solaris 11.4*.

Troubleshooting Custom Network Configuration Applications During an Installation

In this Oracle Solaris release, you can arbitrarily supply customized network configuration for datalinks, flows, IP interfaces, protocol tunables, and static routes during an installation with the Automated Installer (AI) by specifying various parameters in a system configuration profile. To help guide this process, examples are provided in `/usr/share/auto_install/sc_profiles`. SMF template data for the relevant services is also provided.

If any portion of the configuration within the file is misspecified, the parameter (for example, a datalink) might not be configured correctly after the installation. Errors that are encountered during the installation process are logged in the `/var/adm/messages` directory.

An erroneous specification could be the result of one or both of the following:

- Incorrect property type
- Invalid configuration

  An example of an invalid configuration would be a VNIC that is configured over a non-existent datalink, or a datalink other than an IP tunnel that is specified within a non-global zone.

Use the following commands to determine whether there are any datalink or IP network configuration errors:

```
# svccfg -v validate network/datalink-management:default

# svccfg -v validate network/ip-interface-mangement:default
```

For more information about this feature, see Chapter 3, “Working With System Configuration Profiles” in *Customizing Automated Installations With Manifests and Profiles*.

Troubleshooting Network Connectivity and Configuration Issues

The following are general guidelines for troubleshooting network connectivity and configuration issues.

One of the first signs of trouble on a network is loss of communications by one or more of the systems. If a system does not to come up the first time that it is added to the network, the problem might be a faulty NIC or a problem with a network daemon that is managed by SMF.

If a single system that previously was connected to the network suddenly develops a network problem, the problem could be its network interface configuration. If the systems on a network can communicate with each other, but not with other networks, the problem could be the router. Or, the problem could be with the other network.

Performing Basic Network Configuration Checks

You can troubleshoot network configuration problems with a single system by using the `dladm` and `ipadm` commands. These two commands, when used without any options, provide useful information about your current network configuration. The output of both commands displays information about the current state of the link, IP interface, and IP address, respectively.

The following are some of the ways in which you can use these commands to troubleshoot configuration issues:

- Use the `dladm` command to display general information about all of the datalinks that are on a system:

  ```
  # dladm
  LINK        CLASS     MTU     STATE     OVER
  net0        phys      1500    up        --
  ```

- Display information about the mapping between the datalinks, their generic names, and the corresponding network device instances as follows:

  ```
  # dladm show-phys
  LINK     MEDIA        STATE     SPEED     DUPLEX     DEVICE
  net0     Ethernet     up        1000      full       e1000g0
  ```

- Use the `ipadm` command to display general information about all of the IP interfaces that are on a system:

  ```
  # ipadm
  NAME         CLASS/TYPE STATE     UNDER  ADDR
  lo0          loopback   ok        --     --
   lo0/v4       static    ok        --     127.0.0.1/8
   lo0/v6       static    ok        --     ::1/128
   net0         ip        ok        --     --
   net0/v4      static    ok        --     203.0.113.10/24
  ```

- Use the `ipadm show-if` *interface* command to display information about a specific IP interface:

  ```
  # ipadm show-if net0
  IFNAME     CLASS    STATE    ACTIVE OVER
  net0       ip       ok       yes    --
  ```

- Display information about all of the interfaces on a system as follows:

  ```
  # ipadm show-if
  IFNAME       CLASS        STATE     ACTIVE     OVER
  lo0          loopback     ok        yes        --
  net0         ip           ok        yes        --
  ```

- Display information about all of the IP addresses on the system as follows:

```
# ipadm show-addr
ADDROBJ          TYPE       STATE     ADDR
lo0/v4           static     ok        127.0.0.1/8
net0/v4          static     ok        192.0.2.3/24
```

- Use the `ipadm show-addr` *interface* command to display information about a specific interface's IP address:

```
# ipadm show-addr net0
ADDROBJ           TYPE     STATE        ADDR
net0/v4           dhcp     ok           203.0.113.10/24
```

- Display the properties of a specific IP address as follows:

```
# ipadm show-addrprop net0/v4
ADDROBJ    PROPERTY   PERM CURRENT         PERSISTENT     DEFAULT         POSSIBLE
net0/v4    broadcast  r-   203.0.113.10    --             203.255.255.255  --
net0/v4    deprecated rw   off             --             off             on,off
net0/v4    prefixlen  rw   24              --             8               1-30,32
net0/v4    private    rw   off             --             off             on,off
net0/v4    reqhost    r-   --              --             --              --
net0/v4    transmit   rw   on              --             on              on,off
net0/v4    zone       rw   global          --             global          --
```

See the `ipadm`(8) man page.

Verifying That Network Services and Daemons Are Running

A critical step in troubleshooting issues with network connectivity is to determine the current status of all of the SMF network services that are running on the system.

You can verify the current status of the SMF network services that are running on the system as follows:

```
$ svcs -x service-name
```

The output of this command shows all of the services that are in a maintenance state.

To investigate further, check the log file as follows:

```
$ svcs -Lv service-name-from-svcs-x-output
```

Or, use the following command to check the log file:

```
$ view `svcs -L service-name-from-svcs-x-output
```

You can use either of the following commands to check whether the `network` SMF service is online or in a degraded state:

```
$ svcs '*network*' | grep -v online
```

```
$ svcs '*network*' | egrep 'maint|degrade|off|disable
```

Use the following command to obtain more information about the state of the `svc:/network/loopback:default` SMF service:

```
$ svcs -xv svc:/network/loopback:default
svc:/network/loopback:default (loopback network interface)
 State: online since Thu Dec 05 19:30:54 2013
   See: man -M /usr/share/man -s 1M ifconfig
   See: /system/volatile/network-loopback:default.log
Impact: None.
```

Running Basic Network Diagnostic Checks

Less obvious causes of network problems are those that degrade network performance. If the network is having problems, you can run a series of software checks to diagnose and fix basic problems. For example, you can use the `ping` command to quantify problems, such as the loss of packets by a system. Or, you can use the `netstat` command to display routing tables and protocol statistics. For more information about the various methods that you can use to troubleshoot these types of networking problems, see Chapter 2, "Using Observability Tools to Monitor Network Traffic Usage" and "Resources for Monitoring and Detecting Problems on a TCP/IP Network" on page 31.

Third-party network diagnostic programs also provide a number of tools for troubleshooting network issues. Refer to the third-party product documentation for specifics.

The following are some of the methods that you can use to perform basic network software checking:

- Use the `netstat` command to display network information.

  The `netstat` command displays a variety of useful information for troubleshooting network connectivity issues. The type of information that is displayed depends on the options that you specify. See "Monitoring and Analyzing the Network" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4* and the `netstat`(8) man page.
- Check the `/etc/inet/hosts` database to ensure that all of the entries are correct and current.

See the hosts(5) man page.

- Try to connect to the local system by using the telnet command.

  See the telnet(1) man page.
- Ensure that the inetd network daemon is running.

  ```
  # /usr/bin/pgrep inetd
  883
  ```

  The previous output indicates that the inetd daemon is running on the system with the process ID 883.
- If IPv6 is enabled on your network, verify that the svc:/network/routing/ndp SMF service is enabled.

  ```
  # svcs ndp
  STATE          STIME    FMRI
  online         Apr_20   svc:/network/routing/ndp:default
  ```

  The previous output indicates that the svc:/network/routing/ndp SMF service is enabled.
- Check the system's router and routing information.
  - Display a system's persistent route as follows:

    ```
    # route -p show
    ```

    See "Troubleshooting Issues When Adding a Persistent Route" on page 22.
  - Display the configuration that is in the routing table as follows:

    ```
    # netstat -nr
    ```

Troubleshooting Issues When Adding a Persistent Route

You use the route command to manage the network routing tables and to add, modify, and remove persistent routes. Always specify the -p option to ensure that any changes you make to the network routing tables persist across system reboots.

Note - It is important to use care when adding persistent routes to make sure that any routes that are being added do not conflict with any existing persistent configuration.

Check whether a route already exists in the persistent configuration as follows:

```
# route -p show
```

```
persistent: route add default 203.0.113.1 -ifp net0
```

If the route already exists in the persistent configuration, the information that is in the network routing tables (which is not persistent) might differ from the persistent configuration.

The following example illustrates this point further. In this example, an attempt is made to add a persistent route to the `net1` interface. However, the command fails because that persistent route already exists for the `net0` interface, per the previous example's output.

```
# route -p add default 203.0.113.1 -ifp net1
add net default: gateway 203.0.113.1
add persistent net default: gateway 203.0.113.1: entry exists
Warning: persistent route might not be consistent with routing table.
```

Running the `route -p show` command again reveals that the persistent route did not change and is still configured for `net0`, as shown in the following output:

```
# route -p show
persistent: route add default 203.0.113.1 -ifp net0
```

However, the command did change the routing tables in the kernel to use `net1`, as shown in the following output:

```
# netstat -nr

Routing Table: IPv4
  Destination          Gateway           Flags  Ref     Use Interface
-------------------- ------------------- -----  ----    --- ---------
default              203.0.113.1         UG       2       1 net1
203.0.113.0          203.0.113.78        U        3       0 net1
127.0.0.1            127.0.0.1           UH       2     466 lo0
.
.
.
```

Therefore, it is always a best practice to delete any existing persistent route configuration *prior* to adding a new route. See "Creating Persistent (Static) Routes" in *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer*.

Simulating Network Operating Conditions Within a Test Environment

The Network Conditions Simulator (NCS) is a STREAMS module/driver (`ncs`(4D)) that you can use to simulate network operating conditions within a test environment.

For example, you can simulate the following conditions by using NCS:

- Propagation delay
- Bandwidth
- Drop rate
- Packet reordering
- Corruption

How the NCS Works

NCS works in between the IP and datalink layer of the Oracle Solaris network protocol stack by capturing packets from the IP layer, then manipulating those packets to simulate networking conditions that you have configured for a specified network interface by using the `ncsconfig` command. See "Performing NCS Simulations" on page 24 for more information.

You can simultaneously activate the NCS module on multiple network interfaces. However, note that if two interfaces have NCS activated, and target address A is on the target list, then all of the packets that are sent to the target address A are impacted, regardless of which interface is used. The target list is then shared amongst all of the interfaces. This aspect of the feature can be useful for testing the Stream Control Transmission Protocol (SCTP).

Note - The NCS module only operates on output, so the feature can be viewed similarly to how a router functions on the outgoing path. The return path remains unchanged.

Support for NCS in the Oracle Solaris zones environment includes global zones and kernel zones.

Performing NCS Simulations

The following examples show some of the ways in which you might use the `ncsconfig` command to configure the NCS module on a given network interface and display configuration information about systems that are using NCS.

Note that you must first install the NCS software package and then activate the module on each network interface for which you want to perform simulations. Also, to use the `ncsconfig` command, you must be assigned the `Network Management` rights profile.

Tip - To avoid a potential performance issue, deactivate the NCS module on the given network interface whenever you are not performing simulations. See Example 3, "Deactivating the NCS Module for a Network Interface," on page 25.

The nscconfig command supports several additional options and arguments for configuring NCS parameters beyond what is described in this document. See the ncsconfig(8) man page.

EXAMPLE 1 Determining Whether a Network Interface Is Using the NCS Module

The following example shows how to use the ncsconfig command to determine whether a network interface (net0) is using NCS. The -Q option specifies the interface. You can specify multiple interfaces as a comma-separated list or by using multiple -Q options. The -t option specifies the address family (inet6).

```
# ncsconfig -Q net0 -t inet6
ncs module is active in net0
```

EXAMPLE 2 Activating the NCS Module for a Network Interface

The following example shows how to activate the NCS module for a given network interface (net1).

```
# ncsconfig -A net1
# ncsconfig -Q net1
ncs module is active in net1
```

where -A *interface* indicates to activate the specified interface (net1). You can specify multiple interfaces as a comma-separated list or by using multiple -A options.

EXAMPLE 3 Deactivating the NCS Module for a Network Interface

The following example shows how to deactivate the NCS module for a given network interface (net0).

```
# ncsconfig -D net0 -t inet6
# ncsconfig -Q net0
ncs module is not active in net0
```

where -D *interface* indicates to deactivate the specified interface (net0).

To specify multiple network interfaces, use a comma-separated list with the -D option as follows:

```
# ncsconfig -D net0,net1 -t inet6
```

Or, you can specify multiple -D options on the command line as follows:

```
# ncsconfig -D net0 -D net1 -t inet6
```

EXAMPLE 4 Displaying the NCS Configuration for a System

The following example shows how to display the NCS configuration for a system by specifying an IP address (203.0.113.3).

```
# ncsconfig -h 203.0.113.3
[ 203.0.113.3 ](203.0.113.3)
Total bytes processed: 0, total packets processed: 0
```

You can also use the -h option to specify a host name. When you specify a host name rather than an IP address, both the host name and the IP address of the system are included in the output of the command, as shown in the following example:

```
# ncsconfig -h systemA -t inet
[ systemA.example.com ](203.0.113.3)
Total bytes processed: 0, total packets processed: 0
```

If you do not include the -t option when specifying a host name, NCS uses IPv4, which is the default:

```
# ncsconfig -h systemA
[ systemA.example.com ] (203.0.113.3)
Total bytes processed: 0, total packets processed: 0
```

Note that you cannot specify multiple hosts as part of a single command when using the -h option.

EXAMPLE 5 Displaying a List of Systems That Are Using the NCS Module

The following example shows how to display all of the targets (hosts) that are currently using the NCS module.

```
# ncsconfig
ncs target list:     203.0.113.3
```

The output of this example indicates that there is one host with the IP address 203.0.113.3 that is currently using the NCS module.

Troubleshooting Interface Configuration Error Conditions

This section contains the following topics:

- “Cannot Assign IP address by Using the ipadm create-addr command” on page 27

- "Error Message: cannot create address object: Invalid argument provided" on page 27
- "Error Message: cannot create address: Persistent operation on temporary object" on page 28

Cannot Assign IP address by Using the ipadm create-addr command

With the traditional `ifconfig` command that is used for network configuration in previous Oracle Solaris releases, you could plumb and assign an IP address by using a single command. In this Oracle Solaris release, you use the `ipadm` command to configure both IP interfaces and IIP addresses.

The following example assumes that a static IP address is being assigned to an interface. This process involves two steps. First, create or plumb the IP interface by using the `ipadm create-ip` command. Then, assign an IP address to the interface by using the `ipadm create-addr` command:

```
# ipadm create-ip interface
# ipadm create-addr -T addr-type -a address addrobj
```

Error Message: cannot create address object: Invalid argument provided

This error message is displayed during IP address configuration.

The address object identifies a specific IP address that is bound to an IP interface. The address object is a unique identifier for each IP address on the IP interface. You must specify a different address object to identify a second IP address that you assign to the same IP interface. If you want to use the same address object name, you must delete the first address object instance before assigning it to a different IP address.

Use one of the following methods:

- Specify a different address object to identify a second IP address as follows:

```
# ipadm show-addr
ADDROBJ    TYPE     STATE    ADR
lo0        static   ok       127.0.0.1/10
```

```
net0/v4   static   ok      192.0.2.1
# ipadm create-addr -T static -a 192.0.2.5 net0/v4b
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADR
lo0       static   ok      127.0.0.1/10
net0/v4   static   ok      192.0.2.1
net0/v4b  static   ok      192.0.2.5
```

- Delete the first instance of the address object, then assign the same address object to a different IP address, as follows:

```
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADR
lo0       static   ok      127.0.0.1/8
net0/v4   static   ok      192.0.2.1
# ipadm delete-addr net0/v4
# ipadm create-addr -T static -a 192.0.2.5 net0/v4
# ipadm show-addr
ADDROBJ   TYPE     STATE   ADR
lo0       static   ok      127.0.0.1/8
net0/v4   static   ok      192.0.2.5
```

Error Message: `cannot create address: Persistent operation on temporary object`

This error message is displayed during IP interface configuration.

By default, you use the `ipadm` command to create persistent network configuration. If the IP interface that you are configuring was created as a temporary interface, you cannot use the `ipadm` command to configure persistent settings on that interface. After verifying that the interface you are configuring is temporary, delete the interface and recreate it persistently. You can then resume configuring the interface as follows:

```
# ipadm show-if -o all
IFNAME   CLASS      STATE   ACTIVE   CURRENT        PERSISTENT   OVER
lo0      loopback   ok      yes      -m46-v------   46--         --
net0     ip         ok      yes      bm4---------   ----         --
```

The absence of a `4` flag for an IPv4 configuration or a `6` flag for an IPv6 configuration in the `PERSISTENT` field indicates that `net0` was created as a temporary interface.

```
# ipadm delete-ip net0
# ipadm create-ip net0
# ipadm create-addr -T static -a 192.0.2.10 net0/v4
ipadm: cannot create address: Persistent operation on temporary object
```

Troubleshooting Issues With IPv6 Deployment

Refer to the following information if you encounter any issues while planning and deploying IPv6 at your site. For specific planning tasks, see Chapter 2, "Planning for Using IPv6 Addresses" in *Planning for Network Deployment in Oracle Solaris 11.4*.

IPv6 Interface Is Not Configured Correctly

The existence of an IPv6 interface does not necessarily mean the system is using IPv6. The interface is not brought up until you actually configure an IPv6 address on that interface.

For example, the following output of the `ifconfig` command shows that the `inet6 net0` interface has not been marked as `UP` and has an address of `::/0`,meaning an IPv6 interface is not configured.

```
# ifconfig net0 inet6
net0:
flags=120002000840<RUNNING,MULTICAST,IPv6,PHYSRUNNING> mtu 1500 index 2 inet6 ::/0
```

The `in.ndpd` daemon still runs on the system but does not operate on any IP interfaces that do not have an `addrconf` address configured.

Cannot Upgrade IPv4 Router to IPv6

If you cannot upgrade your existing equipment, you might need to purchase IPv6-ready equipment. Check the manufacturer's documentation for any equipment-specific procedures that you might be required to perform to support IPv6.

You cannot upgrade certain IPv4 routers for IPv6 support. If this situation applies to your topology, as an alternative, you can physically wire an IPv6 router next to the IPv4 router. Then, you can tunnel from the IPv6 router over the IPv4 router. For instructions on configuring IP tunnels, see Chapter 5, "Administering IP Tunnels" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Problems Encountered When Upgrading Services to Support IPv6

You might encounter the following issues when preparing services for IPv6 support:

- Certain applications, even after being ported to IPv6, do not turn on IPv6 support by default. You might have to configure these applications to turn on IPv6.
- A server that runs multiple services, some of which are IPv4 only and others that are both IPv4 and IPv6, can experience problems. Some clients might need to use both types of services, which can lead to confusion on the server side.

Current ISP Does Not Support IPv6

If you want to deploy IPv6, but your current Internet Service Provider (ISP) does not offer IPv6 addressing, consider the following alternatives:

- Hire another ISP to provide a second line for IPv6 communications from your site. This solution is expensive.
- Get a virtual ISP. A *virtual ISP* provides your site with IPv6 connectivity but no link. Instead, you create a tunnel from your site, over your IPv4 ISP, to the virtual ISP.
- Use a 6to4 tunnel over your ISP to other IPv6 sites. For an address, you can use the registered IPv4 address of the 6to4 router as the public topology part of the IPv6 address. For more information, see “How to Configure a 6to4 Tunnel” in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Security Issues When Tunneling to a 6to4 Relay Router

By nature, a tunnel between a 6to4 router and a 6to4 relay router is insecure. The following types of security problems are inherent in such a tunnel:

- Though 6to4 relay routers do encapsulate and decapsulate packets, these routers do not check the data that is contained within the packets.
- Address spoofing is a major issue on tunnels to a 6to4 relay router. For incoming traffic, the 6to4 router is unable to match the IPv4 address of the relay router with the IPv6 address of the source. Therefore, the address of the IPv6 system can easily be spoofed. The address of the 6to4 relay router can also be spoofed.
- By default, no trusted mechanism exists between 6to4 routers and 6to4 relay routers. Thus, a 6to4 router cannot identify whether the 6to4 relay router is to be trusted or even if it is

a legitimate 6to4 relay router. A trusted relationship between the 6to4 site and the IPv6 destination must exist. Otherwise, both sites leave themselves open to possible attacks.

These problems and other security issues that are inherent with 6to4 relay routers are explained in RFC 3964, Security Considerations for 6to4 (`http://www.rfc-editor.org/rfc/rfc3964.txt`). See also RFC 6343, Advisory Guidelines for 6to4 Deployment (`http://www.rfc-editor.org/rfc/rfc6343.txt`) for updated information about using 6to4.

Generally, you should consider enabling support for 6to4 relay routers for the following reasons only:

- Your 6to4 site intends to communicate with a private, trusted IPv6 network. For example, you might enable 6to4 relay router support on a campus network that consists of isolated 6to4 sites and native IPv6 sites.
- Your 6to4 site has a compelling business reason to communicate with certain native IPv6 systems.
- You have implemented the checks and trust models that are suggested in Security Considerations for 6to4 (`http://www.ietf.org/rfc/rfc3964.txt`) and Advisory Guidelines for 6to4 Deployment. (`http://www.ietf.org/rfc/rfc6343.txt`).

Resources for Monitoring and Detecting Problems on a TCP/IP Network

The following table describes tasks for monitoring and detecting problems on a TCP/IP network. For complete instructions, see *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

TABLE 1 Tasks for Monitoring TCP/IP Networks

Task	Command and/or Description	Task Information
Monitor network traffic for features at the various layers of the Oracle Solaris network protocol stack.	Depending on the feature and at which layer of the network protocol the feature is configured, you can use a variety of observability tools to gather statistics and monitor network traffic usage.	"Observing Network Configuration and Traffic Usage" on page 37
Log the IP addresses of all incoming TCP connections.	Transport layer protocols typically need no intervention to run properly. However, in some circumstances, you might need to log or modify services that run over the transport layer protocols.	"Logging IP Addresses of All Incoming TCP Connections" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Determine whether a remote system is running.	Use the `ping` command to determine the status of a remote system.	"Using the ping Command" in *Administering TCP/IP Networks,*

Task	Command and/or Description	Task Information
		IPMP, and IP Tunnels in Oracle Solaris 11.4
Detect whether a system is dropping packets	Use the `-s` option of the `ping` command to determine whether a remote system is running but losing packets.	"Investigating Dropped Packets" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Display network statistics on a per-protocol basis.	Use the `netstat` command to display statistics on a per-protocol basis for TCP, Stream Control Transmission Protocol (SCTP), and User Datagram Protocol (UDP) endpoints in table format.	"Monitoring and Analyzing the Network" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Perform TCP and UDP management.	Use the `netcat` (or `nc`) utility to open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, perform port scanning.	"Performing TCP and UDP Administration With the netcat Utility" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Trace the actions of the IPv4 routing daemon, including all packet transfers.	If you suspect a malfunction of the `routed` daemon, you can start a log that traces the daemon's activity. The log includes all packet transfers when you start the `routed` daemon.	"Logging Actions of the IPv4 Routing Daemon" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Discover the route to a remote system.	Use the `traceroute` command to discover the route to a remote system. The output displays the number of hops in the path a packet follows.	"Discovering the Route to a Remote Host" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Monitor IPv4 and IPv6 packet transfer processes.	Use the `snoop` command to monitor the state of package (data) transfers.	"Using the snoop Command" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Analyze network traffic.	Use the TShark command line interface (CLI) or the Wireshark graphical user interface (GUI) to analyze network traffic.	"Using tshark and Wireshark" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Monitor network traffic on a server.	Use the `ipstat` and `tcpstat` commands to monitor network traffic on a server.	"Using the ipstat and tcpstat Commands" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*
Monitor the status of IPMP on your system.	Use the `ipmpstat` command to gather different types of information about the status of IPMP. You can also use the command to display information about the underlying IP interfaces for each IPMP group, as well as data and test addresses are configured for the group.	"Monitoring IPMP Information" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*

Troubleshooting IPMP Configuration

This section contains the following topics:

- "Failure Detection in IPMP" on page 33
- "Disabling Outbound Load Spreading in Link-Based IPMP Configuration" on page 33
- "Error Message: `*ipadm: cannot add net0 to ipmp0: Underlying interface has addresses managed by dhcpagent(8)*`" on page 34

Failure Detection in IPMP

To ensure continuous availability of the network to send or receive traffic, IPMP performs failure detection on an IPMP group's underlying IP interfaces. Failed interfaces remain unusable until they are repaired. Any remaining active interfaces continue to function while any existing standby interfaces are deployed, as needed.

The `in.mpathd` daemon handles the following types of failure detection:

- Probe-based failure detection:
  - No test addresses are configured (transitive probing)
  - Test addresses are configured
- Link-based failure detection, if supported by the NIC driver

For more details, see "Failure Detection in IPMP" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Disabling Outbound Load Spreading in Link-Based IPMP Configuration

It is possible to disable outbound load spreading in link-based IPMP. If you mark an interface as standby, that interface is not used until an active interface fails, regardless of whether link-based or probe-based failure detection is used. Link-based failure detection is always enabled by the `in.mpathd` daemon.

Use the `ipadm` command as follows:

```
# ipadm set-ifprop -m ip -p standby=on interface
```

For information about how inbound and outbound load spreading in link-based IPMP works, see "Functions of an IPMP Configuration" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

Error Message: `*ipadm: cannot add net0 to ipmp0: Underlying interface has addresses managed by dhcpagent(8)*`

The following error message might be displayed when you attempt to add an IPMP group:

```
*ipadm: cannot add net0 to ipmp0: Underlying interface has addresses managed by
 dhcpagent(8)*
```

This message is displayed because you cannot add an IP interface with an address that is managed by `dhcpagent` into an IPMP group. As a workaround, disable the DHCP or stateful address configuration on `net0` before adding it to the IPMP group.

Troubleshooting Issues With VRRP and the Oracle Solaris Bundled Packet Filter

The Virtual Router Redundancy Protocol (VRRP) provides high availability of IP addresses, such as those that are used for routers and load balancers. Oracle Solaris supports both L2 and L3 VRRP. The standard VRRP multicast address (224.0.0.18/32) is used to ensure that VRRP functions properly. See `http://www.rfc-editor.org/rfc/rfc5798.txt` for more information. When you use VRRP with the Oracle Solaris bundled Packet Filter, you must explicitly check whether outgoing or incoming IP traffic is allowed for the multicast address.

Use the `pfctl -sr` command to check the IP traffic information:

```
# pfctl -sr
 #

    /* pfctl -sr shows rules, loaded to PF kernel module, of there
     * are no rules loaded output is empty
     */
```

If the output of the command indicates that traffic is not allowed for the standard multicast address, you must add the following rules to the Packet Filter configuration for each VRRP router:

```
# echo "pass quick on VRRP VIP Interface from VRRP VIP/32 to 224.0.0.18/3" | pfctl -f -
```

See Chapter 3, “Using Virtual Router Redundancy Protocol” in *Configuring an Oracle Solaris 11.4 System as a Router or a Load Balancer*.

♦ ♦ ♦ CHAPTER 2

Using Observability Tools to Monitor Network Traffic Usage

This chapter describes how to use Oracle Solaris network observability tools to display configuration information and monitor network traffic usage on each layer of the Oracle Solaris network protocol stack. This chapter contains examples for selected networking features and a particular network configuration scenario.

This chapter contains the following topics:

About Network Troubleshooting and Observability

The following tables describe the major features of the Oracle Solaris network protocol stack, grouped by layer. Information about the tools that you use to observe and monitor network traffic usage and display the configuration for each of these features is also provided. For the tools that you use to manage and observe various networking features, such as the `dladm` and `dlstat` commands, the subcommands that are specific to each feature are also provided.

TABLE 2 Networking Features on the Hardware Layer of the Network Protocol Stack

Feature	Functional Area	Administrative Interface
Physical network interface card (NIC)	Hardware connectivity	`dladm show-phys`, `dlstat show-phys`
Driver configuration	Driver connectivity	Managed through the `driver.conf` file and `dladm` properties (`dladm show-linkprop`)

TABLE 3 Networking Features on the Datalink Layer of the Network Protocol Stack

Feature	Functional Area	Administrative Interface
Aggregations (DLMP and trunk)	High availability	`dladm show-aggr` and `dlstat show-aggr`
Bridging protocols: - STP - TRILL	High availability, network virtualization	`dladm show-bridge` and `dlstat show-bridge`
Data Center Bridging (DCB)	Network storage, performance	`lldpadm`, `dladm`
Etherstubs	Network virtualization	`dladm show-etherstub`
Edge Virtual Bridging (EVB)	Network virtualization	`dladm`
Elastic Virtual Switch (EVS)	Network virtualization	`evsadm`, `evsstat`, `dladm`
Flows	Observability, resource management, security	`flowadm` and `flowstat`
IP tunnels	IP connectivity	`dladm show-iptun`, `ipadm`
Link Layer Datalink Protocol (LLDP)	Observability, network storage, network virtualization	`lldpadm`
Virtual local area networks (VLANs)	Network virtualization	`dladm show-vlan`, `dlstat`
Virtual network interface cards (VNICs)	Network virtualization	`dladm show-vnic`, `dlstat`
Virtual extensible area networks (VXLANs)	Network virtualization	`dladm show-vxlan`, `dlstat`

TABLE 4 Networking Features on the IP Layer of the Network Protocol Stack

Feature	Functional Area	Administrative Interface
Elastic Virtual Switch (EVS)	Network virtualization	`evsadm`, `evsstat`, `dladm`
Firewalls	Security	Packet filtering with `pfconf`,and `pfctl`
Flows	Observability, resource management, security	`flowadm`, `flowstat`
Integrated Load Balancer (ILB)	Performance	`ilbadm`, `ilbadm show-server`, `ilbadm show-servergroup`
IPMP	High availability	`ipadm`
IP tunnels	IP connectivity	`ipadm show-iptun`
Routing	IP connectivity	`route`, `netstat -r`, and with SMF commands
VNIs	IP connectivity	`ipadm`
VNICs	Network virtualization	`dladm show-vnic` and `dlstat`
VRRP	High availability	`dladm`, `vrrpadm`

Feature	Functional Area	Administrative Interface
VXLANs	Network virtualization	`dladm show-vxlan` and `dlstat`

TABLE 5 Networking Features on the Transport Layer of the Network Protocol Stack

Feature	Functional Area	Administrative Interface
Firewalls	Security	Packet filtering with `pf` and `ipnat`
Flows	Observability, resource management, security	`flowadm` and `flowstat`
Pluggable congestion control	Performance	`ipadm`
Socket filtering	Security	`soconfig` (`-F`)

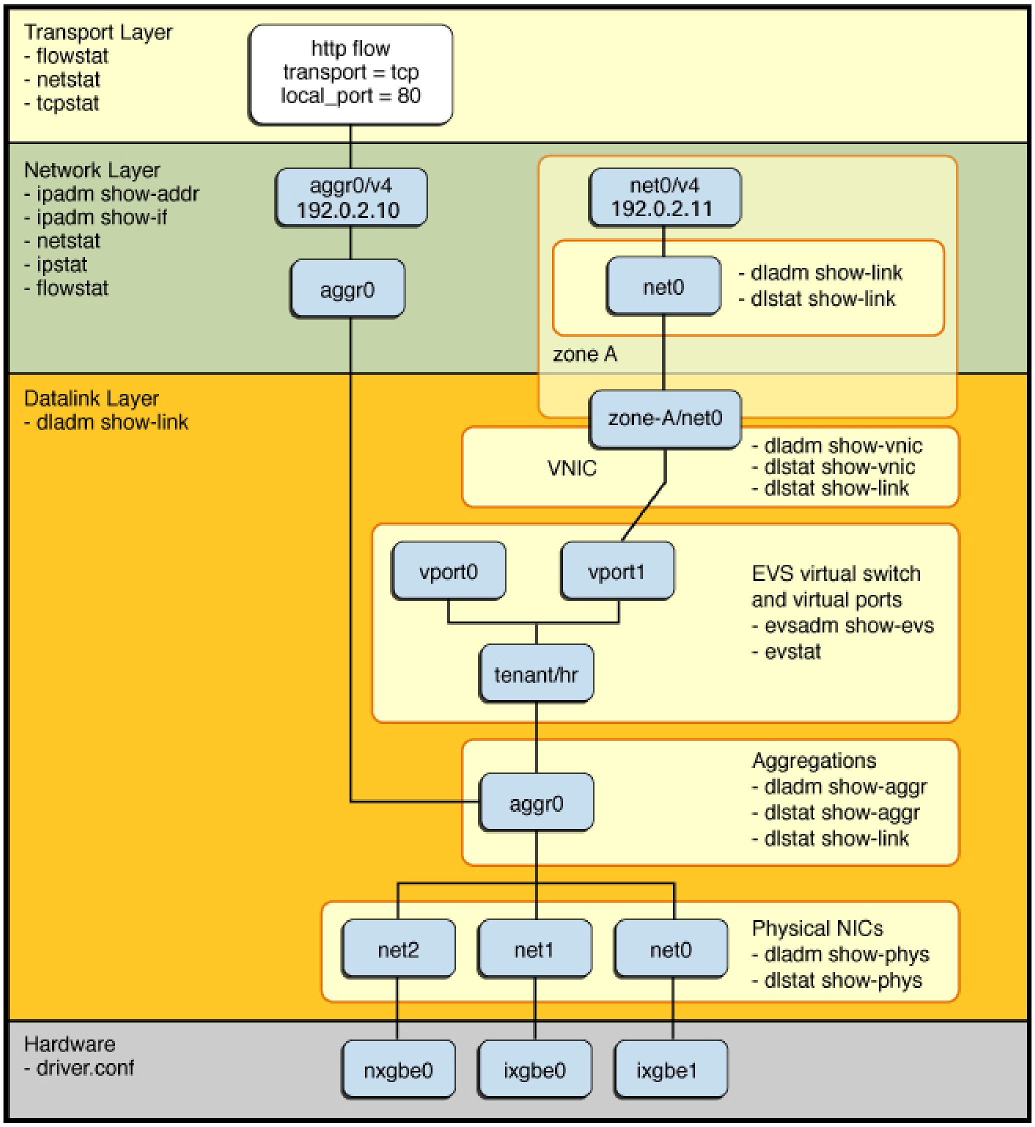
Observing Network Configuration and Traffic Usage

The following information further describes how to use the observability tools that are described in the tables in "About Network Troubleshooting and Observability" on page 35.

The following figure shows a common, hypothetical network configuration to illustrate how various networking features are configured at each layer of the Oracle Solaris network protocol stack.

Note - The figure does not include every possible datalink type that you can configure. For a more detailed description of all of the features that you can configure on each layer of the network protocol stack, see Chapter 1, "Summary of Oracle Solaris Network Administration" in *Strategies for Network Administration in Oracle Solaris 11.4*.

FIGURE 1 Network Configuration Within the Oracle Solaris Network Protocol Stack



The figure illustrates the following configuration:

- At the physical layer of the network stack, three physical NICs, `nxgbe0`, `ixgbe0`, and `ixgbe1`, are present in the system and appear as physical datalink instances, `net2`, `net1`, and `net0`, respectively.
- These physical NICs are then grouped together into an aggregation called `aggr0`.
- The aggregated datalink is then configured directly with an IP address (`aggr0/v4`) and is also used simultaneously as the uplink of a virtual switch, called `tenant/hr`, which is configured as an elastic virtual switch. The virtual switch has two virtual ports, `vport0` and `vport1`.
- An Oracle Solaris zone (`zone-A`) has a VNIC called `zone-A/net0`, which is connected to one of the virtual ports. Within the zone itself, the VNIC appears as a datalink (`net0`), which is configured with an IP address (`net0/v4`).
- A flow for the HTTP traffic is also created on top of the aggregation (`aggr0`).

Tools for Observing Network Configuration and Traffic Usage

You configure and administer datalinks by using the `dladm` command. You use the `dlstat` command to obtain statistics on network traffic usage for datalinks.

To display inbound and outbound traffic statistics per link, use one of the following commands:

```
# dlstat link
```

```
# dlstat show-link link
```

To display inbound and outbound traffic statistics per physical network device, use the following command:

```
# dlstat show-phys link
```

To display inbound and outbound traffic statistics per port and per link aggregation, use the following command:

```
# dlstat show-aggr link
```

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

You configure and administer *flows* by using the `flowadm` command. You use the `flowstat` command to obtain statistics on network traffic usage for flows. As shown in Figure 1, “Network Configuration Within the Oracle Solaris Network Protocol Stack,” on page 38, depending on the attributes that you specify, you can use flows to observe network traffic usage at different layers of the network stack. For more information, see the `flowadm`(8) and `flowstat`(8) man pages.

For more information about monitoring network traffic usage, see Chapter 8, "Monitoring Network Traffic and Resource Usage" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

Observing Network Configuration and Traffic Usage at the Hardware Layer

Troubleshooting network configuration and performance issues at the hardware layer of the network protocol stack might involve observing the following:

- Number of rings per physical NIC and number of packets being transmitted through those rings.
- Number of packet drops per physical NIC and per physical ring.
- NIC-specific counters that might be useful.
- Number of rings and number of descriptors that are configured per physical NIC.

For physical devices, you can use the `dladm show-phys` and `dlstat show-phys` commands to observe network traffic usage. These two commands display different output, depending on the type of information that you want to obtain.

For example, use the `dladm show-phys` command without any options to display the physical device and the attributes for all of the physical links on a system:

```
# dladm show-phys
LINK              MEDIA                STATE      SPEED    DUPLEX     DEVICE
net1              Ethernet             unknown    0        unknown    bge0
net0              Ethernet             up         1000     full       nge0
```

For more information, see Chapter 2, "Administering Datalink Configuration in Oracle Solaris" in *Configuring and Managing Network Components in Oracle Solaris 11.4* and the `dladm`(8) man page.

The `dlstat show-phys` command displays statistics about the packets and bytes that are transmitted and received per physical device. This subcommand operates on the hardware rings, which are at the hardware layer of the network stack.

The following example displays statistics for all of the physical links on a system. The output displays both incoming and outgoing traffic statistics for each link on a system. Information about the number of packets and byte sizes per packet is also displayed.

```
# dlstat show-phys
LINK    IPKTS   RBYTES    OPKTS   OBYTES
net1        0        0        0        0
net0    1.95M  137.83M   37.95K    3.39M
```

You can use the -r option to display receive-side statistics on each of the hardware rings for a device. The output of this command includes the bytes and packets that are received and the hardware and software drops, and so on, for the device. The following example shows that the net4 datalink has eight rings, which are identified under the INDEX field.

```
# dlstat show-phys -r net4
LINK  TYPE  INDEX    IPKTS   RBYTES
net4    rx      0      701   42.06K
net4    rx      1        0        0
net4    rx      2        0        0
net4    rx      3        0        0
net4    rx      4        0        0
net4    rx      5        0        0
net4    rx      6        0        0
net4    rx      7        0        0
```

To obtain similar information for transmitted traffic, use the -t option.

The following example displays the number of inbound dropped packets per physical link.

```
# dlstat show-phys -o idrops
IDROPS
     0
871.14K
```

The -o *field*[,...] option is used to specify a case-insensitive, comma-separated list of output fields to display.

In the following example, the number of inbound and outbound dropped packets and bytes per physical link are displayed.

```
# dlstat show-phys -o idrops,idropbytes,odrops,odropbytes
IDROPS    IDROPBYTES   ODROPS ODROPBYTES
0                0        0          0
871.14K          0        0          0
```

You must specify both the idrops and idropbytes options with the dlstat show-phys -o command. Note that one of these values can be zero, depending on the system's hardware capabilities, as shown in the previous output where the IDROPS field is non-zero, while the IDROBYTES field is zero.

For driver configuration, you can manage property values for specific drivers through the driver.conf file and through datalink properties. Driver configuration files enable you to provide device property values that override the default values that are provided by the driver devices themselves. For more information, see the driver.conf(5) man page.

Observing Network Configuration and Traffic Usage at the Datalink Layer

Several networking features are configured at the datalink layer (L2) of the network protocol stack. These features include both physical and virtual datalinks. Certain commands that you use to observe network traffic usage at this layer of the stack are generic and can be used for any type of configured datalink. Other subcommands are specific to the feature itself and therefore can be used to display additional information about the configuration of that feature.

The commands that you use at this layer of the stack also depend on the type of information that you want to observe. For example, at the datalink layer of the stack, you might want to display fan-out statistics or per-link statistics. You can use different commands to obtain each type of information.

For basic information about datalinks, use the `dladm show-link` command. This command displays link configuration information, either for all of the datalinks on the system or for a specified datalink.

```
# dladm show-link
LINK             CLASS     MTU    STATE    OVER
net1             phys      1500   unknown  --
net0             phys      1500   up       --
```

The output shows that this system has two datalinks, which are directly associated with their corresponding physical NICs. No special datalinks exist on the system, for example, an aggregation or a VNIC. These types of L2 entities are configured over the physical datalinks under the `phys` class.

You use the `dlstat show-link` command to observe network traffic usage at the datalink layer. The `show-link` subcommand operates at the datalink layer of the network protocol stack and provides statistics that refer to the lanes that are configured over the physical link.

The following output shows inbound and outbound traffic statistics per link:

```
# dlstat show-link
LINK      IPKTS    RBYTES     OPKTS    OBYTES
net1          0         0         0         0
net0      1.96M  137.97M     38.40K     3.29M
```

In the following example, receive-side traffic statistics for the `net4` device are reported. The statistics for `INTRS` and `POLLS` counters are also displayed, which indicates how many packets were received in the interrupt context versus the polling mode. The `IDROPS` counter indicates how many packets were dropped at the datalink layer of the network stack.

```
# dlstat show-link -r net4
LINK  TYPE      ID    INDEX    IPKTS  RBYTES    INTRS    POLLS   IDROPS
net4    rx     local      --        0      0        0        0        0
net4    rx     other      --        0      0        0        0        0
net4    rx        hw       0    7.46M  1.06G    5.62M    1.84M        0
net4    rx        hw       1        0      0        0        0        0
net4    rx        hw       2        0      0        0        0        0
net4    rx        hw       3        0      0        0        0        0
net4    rx        hw       4        2    196        2        0        0
net4    rx        hw       5        0      0        0        0        0
net4    rx        hw       6        0      0        0        0        0
net4    rx        hw       7        0      0        0        0        0
```

In the example output, only statistics for the named link, physical device (for the `show-phys` subcommand), or aggregation (for the `show-aggr` subcommand) is displayed. If *link* is not specified in the command, then statistics for all of the links, devices, and aggregations are displayed in the output.

In this example, the information that is displayed under the `ID` field is interpreted as follows:

- `local` – Denotes corresponding loopback traffic on layer 2 (L2) of the network stack.
- `other` – Includes broadcast and multicast traffic.

  Over the lifetime of a datalink, the hardware resources that are associated with a datalink might vary, depending on resource utilization, link configuration, or the assignment of physical NICs to link aggregations. The `rx` entries that are listed in the output of the `show-link -r` command correspond to the hardware resource that is currently assigned to the link. The output for the `other` row includes traffic for hardware resources that are no longer assigned to the datalink.
- `hw` – Denotes a hardware lane.
- `sw` – Denotes a software lane (as shown in the following example).

The distinction between hardware and software lanes is based on the ability of a NIC to support ring allocation. On hardware lanes, rings are dedicated to the packets that use those lanes. In contrast, rings on software lanes are shared amongst datalinks.

The following output displays statistics for outbound packets on the rings that are used by `net4`:

```
# dlstat show-link -t net4
LINK  TYPE      ID  INDEX    OPKTS   OBYTES   ODROPS
net4    tx   local     --        0        0        0
net4    tx   other     --        0        0        0
net4    tx      hw      0      372   15.67K        0
net4    tx      hw      1        1       98        0
```

```
net4    tx      hw      2        0        0        0
net4    tx      hw      3        0        0        0
net4    tx      hw      4        0        0        0
net4    tx      hw      5        0        0        0
net4    tx      hw      6        1       98        0
net4    tx      hw      7        0        0        0
```

You can use the `-f` option to display received and dropped fanout packets, as in the following example that shows how to display fanout statistics:

```
# dlstat show-link -f
LINK  INDEX    IPKTS   IDROPS
net0      0    1.38K        0
net0      1       36        0
net0      2       25        0
net0      3      147        0
net0      4       15        0
net0      5       45        0
net0      6       18        0
net0      7       39        0
```

Observing Network Configuration and Traffic Usage for Aggregations

Aggregations are also configured at the datalink layer (L2) of the network protocol stack. Depending on the type of information that you want to obtain, for example, the overall distribution of traffic between the physical NIC or aggregation statistics, use the following commands:

`dladm show-aggr` Displays aggregation configuration (the default), Link Aggregation Control Protocol (LACP) information or DLMP probe-based failure and recovery detection status, either for all of the aggregations or for a specified aggregation.

Specify the `-x` option to display detailed per-aggregation information for an aggregation.

`dlstat show-aggr` Displays statistics about the packets and bytes that are transmitted and received for an aggregation.

`dlstat show-link` Displays statistics about the packets and bytes that are transmitted and received for an aggregation (the aggregation being a datalink).

One difference between the `dlstat show-aggr` and `dlstat show-link` commands is that the `dlstat show-aggr` command displays per-port statistics, while the `dlstat show-link` command provides per-link statistics. Another important difference between these two commands is that the `dlstat show-aggr` command displays the overall statistics for the entire aggregation. Whereas, the `dlstat show-link` command displays only the statistics for the master client of the aggregation, for example IP.

Thus, if you create VNICs on top of an aggregation, the `dlstat show-aggr` command would report the total number of packets across all of the VNICs, plus the primary client (IP). This output is similar to the output of the `show-phys` subcommand compared to the output of the `show-link` subcommand, where `show-phys` displays the total traffic usage, while `show-link` displays only the traffic usage for the primary datalink.

For more information about administering aggregations, see Chapter 2, "Configuring High Availability by Using Link Aggregations" in *Managing Network Datalinks in Oracle Solaris 11.4*.

EXAMPLE 6 Displaying Aggregation Configuration Information

The output of the `dladm show-aggr -x` displays the status of the existing aggregations that are configured on a system.

```
# dladm show-aggr -x
LINK       PORT        SPEED    DUPLEX     STATE    ADDRESS             PORTSTATE
aggr1      --         1000Mb      full        up   0:14:4f:29:d1:9d        --
net1                  1000Mb      full        up   0:14:4f:29:d1:9d     attached
net3                     0Mb   unknown      down   0:14:4f:29:d1:9f     standby
```

EXAMPLE 7 Displaying Per-Port Statistics for Aggregations

The output for the `dlstat show-aggr` command displays per-port statistics for an aggregation. Both the packets and bytes that are transmitted and received for the aggregation are displayed.

```
# dlstat show-aggr
LINK            PORT    IPKTS      RBYTES    OPKTS   OBYTES
aggr1            --      99        2.18K     23       966
aggr1            net4    25        1.50K      8       336
aggr1            net5    74        10.68K    15       630
```

EXAMPLE 8 Displaying Per-Link Statistics for Aggregations

The output of the `dlstat show-link` command displays per-link statistics for an aggregation. Both the packets and bytes that are transmitted and received for the aggregation are displayed. The difference between this example and the previous example is that the `show-aggr` subcommand displays per-port statistics, while the `show-link` subcommand displays per-link statistics.

```
# dlstat show-link
LINK    IPKTS     RBYTES    OPKTS   OBYTES
net5        0          0        0        0
net2        0          0    5.60K    1.49M
net4        0          0        0        0
net6    4.43K      1.32M    6.39K    1.56M
net1    4.43K      1.32M    6.39K    1.56M
net0  387.10K     99.42M   59.43K    7.67M
net3        0          0    5.61K    1.50M
aggr1     150     18.65K       30    1.26K
```

EXAMPLE 9 Displaying Receive-Side Traffic Statistics for Hardware Rings of an Aggregation

This example displays receive-side statistics for each of an aggregation's (`aggr1`) hardware rings.

```
# dlstat show-aggr -r
LINK      PORT  INDEX    IPKTS   RBYTES
aggr1     --       --    7.93M    4.37G
aggr1     net4      0  541.37K  298.21M
aggr1     net4      1        0        0
aggr1     net4      2        0        0
aggr1     net4      3    1.79M  991.63M
aggr1     net4      4  780.47K  433.30M
aggr1     net4      5        1       98
aggr1     net4      6  530.05K  292.34M
aggr1     net4      7  548.36K  301.74M
aggr1     net5      0  524.10K  288.56M
aggr1     net5      1    1.34M  739.92M
aggr1     net5      2  818.08K  448.12M
aggr1     net5      3  235.62K  129.96M
aggr1     net5      4        0        0
aggr1     net5      5        0        0
aggr1     net5      6  554.44K  305.24M
aggr1     net5      7  260.61K  143.71M
```

EXAMPLE 10 Displaying Receive-Side Traffic Statistics for an Aggregation's Hardware Lanes

This example displays receive-side traffic statistics for each hardware lane of an aggregation (`aggr1`). Note that the statistics reported pertain to the traffic for the primary client of the aggregation only, for example, the IP traffic on top of the aggregation. The output does not include traffic statistics for other clients that are configured on top of the aggregation, for example VNICs. To display traffic usage for all of the clients of an aggregation, you must use the `dlstat show-aggr` command, as shown in Example 6, "Displaying Aggregation Configuration Information," on page 45 and Example 9, "Displaying Receive-Side Traffic Statistics for Hardware Rings of an Aggregation," on page 46.

```
# dlstat show-link -r aggr1
LINK   TYPE    ID    INDEX    IPKTS   RBYTES    INTRS   POLLS  IDROPS
aggr1   rx   local     --       0       0        0        0      0
aggr1   rx   other     --       0       0        0        0      0
aggr1   rx      hw      0     721   43.26K     721        0      0
aggr1   rx      hw      1       0       0        0        0      0
aggr1   rx      hw      2       0       0        0        0      0
aggr1   rx      hw      3       0       0        0        0      0
aggr1   rx      hw      4       0       0        0        0      0
aggr1   rx      hw      5       0       0        0        0      0
aggr1   rx      hw      6       0       0        0        0      0
aggr1   rx      hw      7       0       0        0        0      0
aggr1   rx      hw      8  22.23K   1.33M   22.23K        0      1
aggr1   rx      hw      9     693  63.76K      693        0      0
aggr1   rx      hw     10       0       0        0        0      0
aggr1   rx      hw     11       0       0        0        0      0
aggr1   rx      hw     12       0       0        0        0      0
aggr1   rx      hw     13  13.00K   4.45M    13.00K       0      2
aggr1   rx      hw     14       0       0        0        0      0
aggr1   rx      hw     15  10.40K 624.06K   10.40K        0      0
```

Observing Network Configuration and Traffic Usage for an EVS Switch

You configure and manage the elastic virtual switch (EVS) feature of Oracle Solaris at the datalink layer (L2) of the network protocol stack. Depending on the type of information that you want to display, use the following commands:

`evsadm`
Creates and manages EVS switches and their resources: IP networks (*IPnets*) and virtual ports (*VPorts*).

Use the `show-evs` subcommand to display information for all of the EVS switches that are managed by an EVS controller or for a specified EVS switch.

`evsstat`
Displays network traffic statistics for all of the VPorts in a data center or for all of the VPorts of a specified elastic virtual switch. The command also displays the statistics for any VNICs associated with the VPorts.

`dladm show-vnic -c`
Displays information about the VNICs that are connected to an EVS switch. The VPort and the EVS switch to which the VNIC is connected is determined by the EVS switch.

For more information about administering the EVS feature, see Chapter 6, "Administering Elastic Virtual Switches" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

EXAMPLE 11 Displaying Information About an EVS Configuration

In this example, the evsadm command is used to display basic information about an EVS configuration.

```
# evsadm
NAME          TENANT        STATUS  VNIC    IP           HOST
evs0          sys-global    busy    --      ipnet0       sysabc-02
sys-vport0    --            used    vnic0   203.0.113.2/24 sysabc-02
```

EXAMPLE 12 Displaying Inbound and Outbound Traffic Usage for VPorts That Are Connected to an EVS Switch

In this example the evsstat command is used to display incoming and outgoing network traffic statistics for the only VPort that is connected to an EVS switch.

```
# evsstat
VPORT        EVS     TENANT    IPKTS   RBYTES   OPKTS  OBYTES
sys-vport0  evs0 sys-tenant  101.88K   32.86M  40.16K  4.37M
sys-vport1  evs0 sys-tenant    4.50M    6.78G   1.38M 90.90M
```

EXAMPLE 13 Displaying Information About VNICs That Are Connected to an EVS Switch

You can create VNICs on top of an underlying link and you can also connect VNICs to an EVS switch. To obtain information about a VNIC that is connected to an EVS switch, use the dladm show-vnic command with the -c option, as shown in the following example.

```
# dladm show-vnic -c
LINK       TENANT        EVS     VPORT       OVER    MACADDRESS        IDS
vnic0      sys-global    evs0    sys-vport0  net     12:8:20:f2:46:22  VID:200
```

Observing Network Configuration and Traffic Usage for VNICs

VNICs are configured at the datalink layer (L2) of the network protocol stack. Use the following commands to display configuration information and observe network traffic usage for these L2 entities:

dladm show-vnic   Displays VNIC configuration information for all of the VNICs on a system, all of the VNICs on a link, or for a specified vnic-link.

`dlstat` Displays statistics about the packets and bytes that are transmitted and received per VNIC.

For more information about administering VNICs, see "Managing VNICs" in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*.

EXAMPLE 14 Displaying VNIC Configuration Information

The following example displays VNIC configuration information for the existing VNIC on a system (`vnic0`).

```
# dladm show-vnic
LINK        OVER        SPEED  MACADDRESS       MACADDRTYPE IDS
vnic0       net1        1000   2:8:20:f2:46:22  fixed       VID:200
```

EXAMPLE 15 Displaying Network Traffic Statistics for VNICs

In the following example, the `dlstat` command is used to display statistics of the packets and bytes that are transmitted and received by a specific VNIC (`vnic0`).

```
# dlstat vnic0
LINK    IPKTS   RBYTES    OPKTS    OBYTES
vnic0   1.53M   158.18M   154.22K  32.84M
```

Observing Network Configuration and Traffic Usage at the IP Layer

You observe network traffic usage at the IP layer (L3) of the network protocol stack by using a few different commands. Depending on the type of information that you want to display, use the following commands:

`ipadm` Displays general configuration information for IP interfaces and addresses.

`ipadm show-addr` Displays IP address information, either for the given address object (`addrobj`) or for all of the address objects that are configured on a specified interface, including those that are only in the persistent configuration.

`ipadm show-if` Displays network interface configuration information, either for all of the network interfaces that are configured on the system, including those that are only in the persistent configuration, or for a specified interface.

`ipstat` Displays statistics on IP traffic based on the selected output mode and sort order.

`netstat` Displays the contents of certain network-related data structures in various formats.

For more information about administering IP configuration, see "Monitoring IP Interfaces and Addresses" in *Configuring and Managing Network Components in Oracle Solaris 11.4*. See also the `ipstat(8)` and `netstat(8)` man pages.

EXAMPLE 16 Displaying General Information About IP Configuration

You use the `ipadm` command to display general information about IP configuration. This example shows how to display information about all of the IP interfaces and addresses that are configured on a system.

```
# ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
net0              ip         ok           --         --
   net0/v4        static     ok           --         203.0.113.140/24
```

EXAMPLE 17 Displaying Information About Configured IP Interfaces

You use the `ipadm` command with the `show-if` subcommand to display information about the configured IP interfaces on a system, as shown in the following example.

```
# ipadm show-if
IFNAME   CLASS      STATE   ACTIVE   OVER
lo0      loopback   ok       yes     ---
net0     ip         ok       yes     ---
```

EXAMPLE 18 Displaying Information About Configured IP Address Objects

The following example shows how to use the `ipadm` command with the `show-addr` subcommand to display information about configured IP address objects on a system.

```
# ipadm show-addr
ADDROBJ      TYPE       STATE   ADDR
lo0/v4       static     ok      127.0.0.1/8
```

EXAMPLE 19 Displaying Statistics About IP Traffic by Using the `ipstat` Command

The following example shows how to use the `ipstat` command to display statistics about IP traffic. The command provides options for reporting statistics about the IP traffic that matches a specified source or destination address, interface, and higher layer protocols.

```
# ipstat
SOURCE                    DEST                  PROTO      INT      RATE
abc11example-02           dhcp-sys.example       TCP      net0     145.6
dns1.example.com          abc11example-02        UDP      net0      66.0
abc11example-02           dns1.example.com       UDP      net0      10.4
dhcp-sys.example          abc11example-02        TCP      net0       4.0
foo1.example.com          all-sys.mcast.net     ICMP      net0       3.2
```

For more information, see "Using the ipstat and tcpstat Commands" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

EXAMPLE 20 Displaying Connected Sockets by Using the `netstat` Command

The `netstat` command displays network status and protocol statistics. You can display the status of TCP, SCTP, and UDP endpoints in a table format. You can also display routing table and interface information by using this command. The various types of network data that is reported depends upon the command option that you specify.

For example, you can use the `netstat` command without any options to display a list of active sockets for each protocol:

```
# netstat
 TCP: IPv4
   Local Address         Remote Address        Swind   Send-Q  Rwind   Recv-Q  State
  ------------------     -----------------    -------  ------  ------- ------  --------
   localsys.local.port1  remotesys1              65535  0      128592   0
 ESTABLISHED
   localsys.local.port2  localsys.local.port5   130880  0     139264   0
 ESTABLISHED
   localsys.local.port3  localsys.local.port6   139060  0     130880   0
 ESTABLISHED
   localsys.local.port4  remotesys2.remote.port2 65572 63     128480   0
 ESTABLISHED
```

Use the `netstat -P` *protocol* command to limit the display of statistics or state of just those sockets that are applicable to a specific protocol, as shown in this example:

```
# netstat -P tcp
TCP: IPv4
Local Address Remote Address               Swind  Send-Q  Rwind  Recv-Q  State
```

```
------------ --------------------          ------ ------ ------- ------  --------
sys3.48962   foo.com.ldaps                 49232   0     128872  0       ESTABLISHED
sys3.ssh     dhcp1-203-0-113-210.foo.com   64292  63     128480  0       ESTABLISHED
```

You can specify *protocol* as any of the following: `ip`, `ipv6`, `icmp`, `icmpv6`, `igmp`, `udp`, `tcp`, `rawip`.

The following example shows how to use the `netstat -s` command to display per-protocol statistics:

```
# netstat -s
RAWIP   rawipInDatagrams    =      2    rawipInErrors   =  0
        rawipInCksumErrs    =      0
        rawipOutDatagrams   =      2
        rawipOutErrors      =      0

UDP     udpInDatagrams      =   1023    udpInErrors    =   0
        udpOutDatagrams     =   1023
        udpOutErrors        =      0

TCP     tcpRtoAlgorithm     =      4    tcpRtoMin     =   200
        tcpRtoMax           =  60000
        tcpMaxConn          =     -1
        tcpActiveOpens      =    382
        tcpPassiveOpens     =     83
        tcpAttemptFails     =     81
        tcpEstabResets      =      1
        tcpCurrEstab        =      2
        tcpOutSegs          =   6598
        tcpOutDataSegs      =   5653
        tcpOutDataBytes     = 836393
        tcpRetransSegs      =     16
. . .
```

EXAMPLE 21 Displaying Statistics About Flows That Are Configured at the IP Layer of the Network Stack

You can create flows for the various IP addresses or subnets that are configured at the IP layer of the network protocol stack. You can then use the `flowstat` command to display statistics about these flows as shown in the following example.

```
# flowadm add-flow -l net0 -a transport=tcp tcpflow1
# flowadm add-flow -l net4 -a transport=tcp tcpflow2

# flowstat
FLOW      IPKTS   RBYTES   IDROPS    OPKTS   OBYTES   ODROPS
tcpflow2      0        0        0        0        0        0
```

```
tcpflow1     53   5.62K        0       45    5.52K        0
```

To display flow information for a particular device, specify the `-l` option as follows:

```
# flowstat -l net0
FLOW      IPKTS   RBYTES  IDROPS   OPKTS  OBYTES ODROPS
tcpflow1    108   11.19K       0     86   10.45K      0
```

EXAMPLE 22 Creating and Observing Flows for Specific IP Addresses

You can create flows for specific IP addresses by using the `local_ip` and `remote_ip` attributes with the `flowadm add-flow` command. You can then use the `flowstat` command display statistics for these flows, as shown in the following example.

```
# flowadm add-flow -l net0 -a local_ip=203.0.113.45 flow1
# flowadm add-flow -l net4 -a remote_ip=192.0.2.0/24 flow2
# flowstat
FLOW      IPKTS     RBYTES   IDROPS    OPKTS  OBYTES   ODROPS
flow2   528.54K    787.39M        0  179.39K  11.85M        0
flow1   742.81K      1.10G        0        0       0        0
```

Observing Network Configuration and Traffic Usage at the Transport Layer

You observe network traffic for features that are configured and administered at the transport layer (L4) of the network protocol stack by using the following commands:

`flowstat`
Displays runtime statistics about user-defined flows. Use the `flowadm show-flow` command to determine the flow name to specify with the `flowstat` command.

You can use flows as an observability tool rather than just for bandwidth control, for example, to measure the amount of traffic that a specific service consumes.

`netstat`
Displays the contents of certain network-related data structures in various formats. With no arguments, the `netstat` command displays the connected sockets for `PF_INET`, `PF_INET6`, and `PF_UNIX`, unless modified by using the `-f` option.

`tcpstat`
Displayss statistics on TCP and UDP traffic on a server based on the selected output mode and sort order that is specified in the command syntax.

For more information about administering TCP/IP networks with the `netstat` and `tcpstat` commands, see Chapter 1, “Administering TCP/IP Networks” in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4*.

For more information about the `flowstat` command, see “Displaying Network Traffic Statistics of Flows” in *Managing Network Virtualization and Network Resources in Oracle Solaris 11.4*. See also the `flowstat(8)` man page.

EXAMPLE 23 Displaying Runtime Statistics for Flows by Using the `flowstat` Command

The following example output displays a static display of traffic information for all of the configured flows that are on a system. The `flowadm` command is used to determine the name of the flow.

```
# flowadm
FLOW        LINK     PROTO LADDR             LPORT RADDR  RPORT DIR
tcpflow1    net1     tcp   --                --    -- --    bi
tcpflow0    net0     tcp   --                --    -- --    bi
udpflow0    net0     udp   --                --    -- --    bi

# flowstat
FLOW       IPKTS   RBYTES  IDROPS    OPKTS   OBYTES   ODROPS
tcpflow1      0        0       0        0        0        0
tcpflow0  1.39K  117.86K       0    2.16K  260.77K        0
udpflow0      5    1.43K       0        0        0        0
```

You can also use the `flowstat` command with the `-l` option to display statistics for all of the flows for a specified link or statistics for a specified flow, as shown in output of the following two examples:

```
# flowstat -l net0
FLOW      IPKTS   RBYTES   IDROPS    OPKTS    OBYTES   ODROPS
tcpflow0  1.51K  126.85K        0    2.43K   292.85K        0
udpflow0      9    2.80K        0        0         0        0

# flowstat -l net0 tcpflow0
FLOW      IPKTS    RBYTES   IDROPS     OPKTS   OBYTES  ODROPS
tcpflow0  1.66K   137.11K        0     2.69K  324.42K       0
```

EXAMPLE 24 Displaying Information About Transport Layer Data Structures by Using the `netstat` Command

You can use the `netstat` command to display information about data structures at the transport layer (L4) of the network protocol stack, for example TCP or UDP. In the following example, the `netstat -P` *transport-protocol* command is used to display information about TCP.

```
# netstat -p tcp
TCP: IPv4
  Local Address     Remote Address   Swind   Send-Q  Rwind  Recv-Q  State
  ----------------  ---------------  ------- ------ ------- ------ -----------
  localsys.ssh     remotesys1.port4   65380    63     128480   0    ESTABLISHED
  localsys.port1   remotesys2.ldaps   65535     0     128592   0    ESTABLISHED
  localsys.port2   localsys.port5    130880     0     139264   0    ESTABLISHED
  localsys.port3   localsys.port6    139060     0     130880   0    ESTABLISHED
```

EXAMPLE 25 Displaying Statistics for TCP and UDP Traffic by Using the `tcpstat` Command

You use the `tcpstat` command to observe network traffic at the transport layer of the network protocol stack, specifically for TCP and UDP. In addition to the source and destination IP addresses, you can observe the source and destination TCP or UDP ports, the PID of the process that is sending or receiving the traffic, and the name of the global zone in which that process is running.

The following example shows the type of information that is reported when you use the `tcpstat` command with the `-c` option. The `-c` option specifies to print newer reports after previous reports, without overwriting previous reports:

```
# tcpstat -c 3
ZONE         PID PROTO  SADDR             SPORT DADDR             DPORT   BYTES
global    100680 UDP    antares           62763 agamemnon          1023   76.0
global    100680 UDP    antares             775 agamemnon          1023   38.0
global    100680 UDP    antares             776 agamemnon          1023   37.0
global    100680 UDP    agamemnon          1023 antares           62763   26.0
global    104289 UDP    zucchini          48655 antares            6767   16.0
global    104289 UDP    clytemnestra      51823 antares            6767   16.0
global    104289 UDP    antares            6767 zucchini          48655   16.0
global    104289 UDP    antares            6767 clytemnestra      51823   16.0
global    100680 UDP    agamemnon          1023 antares             776   13.0
global    100680 UDP    agamemnon          1023 antares             775   13.0
global    104288 TCP    zucchini          33547 antares            6868    8.0
global    104288 TCP    clytemnestra      49601 antares            6868    8.0
global    104288 TCP    antares            6868 zucchini          33547    8.0
global    104288 TCP    antares            6868 clytemnestra      49601    8.0
Total: bytes in: 101.0  bytes out: 200.0
```

In the following output, the `tcpstat` command reports the five most active TCP traffic flows for a server:

```
# tcpstat -l 5
ZONE            PID PROTO  SADDR            SPORT DADDR            DPORT   BYTES
global        28919 TCP    achilles.exampl  65398 aristotle.exampl   443   33.0
zone1          6940 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
zone1          6940 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
```

```
global         8350 TCP    ajax.example.com  6868 achilles.exampl  61318    8.0
global         8350 TCP    achilles.exampl  61318 ajax.example.com  6868    8.0
Total: bytes in: 16.0  bytes out: 49.0
```

See "Using the ipstat and tcpstat Commands" in *Administering TCP/IP Networks, IPMP, and IP Tunnels in Oracle Solaris 11.4* and the `tcpstat`(8) man page.

♦ ♦ ♦ CHAPTER 3

Troubleshooting Naming Services Issues

This chapter describes basic naming services configuration in Oracle Solaris and how to manage and troubleshoot various related issues that could prevent your network from working properly.

This chapter contains the following topics:

- "About Naming Services Configuration" on page 57
- "Troubleshooting DNS Issues" on page 57
- "Troubleshooting NFS Issues" on page 59
- "Troubleshooting NIS Issues" on page 61

About Naming Services Configuration

Naming services configuration is managed by the Service Management Facility (SMF). Because the SMF repository is now the primary repository for all naming services configuration, you no longer modify a specific file to configure naming services. To make the configuration changes persistent, you must enable or refresh the appropriate SMF services.

If no network configuration exists after an installation, naming services default to `files only` behavior rather than `nis files`. To avoid potential configuration issues, make sure that the `svc:/system/name-service/cache` SMF service is enabled at all times. For more information, see Chapter 1, "About Naming and Directory Services" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

Troubleshooting DNS Issues

This section contains the following procedures:

- "How to Troubleshoot DNS Client Issues" on page 58

- "How to Troubleshoot DNS Server Issues" on page 58

How to Troubleshoot DNS Client Issues

In Oracle Solaris 11.4, you do not edit the `/etc/resolv.conf` file directly to make persistent changes to a DNS client. All DNS client configuration is managed by the `dns/client` SMF service instance. For information about how to enable a DNS client, see "How to Configure a DNS Client in Oracle Solaris" in *Configuring and Managing Network Components in Oracle Solaris 11.4*.

1. **Check the status of the DNS service.**

```
# svcs -xv dns/client:default
```

2. **Check the DNS client service log.**

```
# more /var/svc/log/network-dns-client:default.log
```

3. **Check the name server settings.**

```
# svcprop -p config/nameserver dns/client
```

4. **Check the search settings.**

```
# svcprop -p config/search dns/client
```

5. **Check all DNS settings.**

```
# svcprop -p config dns/client
```

How to Troubleshoot DNS Server Issues

1. **Check the status of the DNS service.**

```
# svcs -xv dns/server:default
```

2. **Check the DNS service log.**

```
# more /var/svc/log/network-dns-server:default.log
```

3. **Check for `syslog` messages.**

```
# grep named /var/adm/messages
```

4. **Start the `named` daemon.**

```
# named -g
```

5. **After resolving the issue, clear the DNS service.**

```
# svcadm clear dns/server:default
```

6. **Verify that the DNS service is back online.**

```
# svcs dns/server:default
```

Troubleshooting Issues With the Name Service Switch File

Check the current configuration for the name service switch file (`/etc/nsswitch.conf`) by running the following command:

```
# svccfg -s name-service/switch listprop config
```

Troubleshooting NFS Issues

This section contains the following procedures:

- "How to Troubleshoot NFS Client Connectivity Issues" on page 59
- "How to Check the NFS Server Remotely" on page 60
- "How to Troubleshoot Issues With the NFS Service on the Server" on page 61

How to Troubleshoot NFS Client Connectivity Issues

Troubleshooting issues with a client connecting to an NFS server can involve several steps, depending on the root cause. The following procedure describes the logical sequence that you might follow to resolve NFS client connectivity issues. If you do not resolve the problem by performing a given step, proceed to the next step until you have identified and corrected the issue.

1. **Check that the NFS server is reachable from the client system.**

```
# ping nfs-server
```

2. **If the NFS server is not reachable from the client, check that the local naming service is running.**

3. **If the local naming service is running, check that the client has the correct host information.**

   ```
   # getent hosts nfs-server
   ```

4. **If the host information on the client is correct, try to reach the NFS server by running the `ping` command from another client.**

5. **If the NFS server is reachable from the second client, check whether the first client can connect to any other systems on the local network by using the `ping` command:**

   ```
   # ping other-client-system
   ```

6. **If other clients are unreachable, follow the steps that are described in “Running Basic Network Diagnostic Checks” on page 21.**

How to Check the NFS Server Remotely

The following procedure describes the logical sequence that you can follow to check an NFS server remotely.

1. **Check whether the NFS services are running on the NFS server.**

   ```
   # rpcinfo -s bee|egrep 'nfs|mountd`
   ```

2. **Check whether the `nfsd` processes are running:**

   ```
   # rpcinfo -u nfs-server nfs
   ```

3. **Check that the `mountd` daemon is running on the NFS server.**

   ```
   # rpcinfo -u nfs-server mountd
   ```

4. **Check whether the local `autofs` service is used.**

   ```
   # cd /net/wasp
   ```

5. **Verify that the file system is shared as expected on the NFS server.**

   ```
   # showmount -e nfs-server
   ```

▼ How to Troubleshoot Issues With the NFS Service on the Server

The following procedure describes the logical sequence that you can follow to verify whether the NFS service is running on the server.

1. **Check whether the server can reach the clients.**

   ```
   # ping client
   ```

2. **If the client is not reachable, check that the local naming service is running.**

3. **If the naming service is running, check the networking software configuration on the server, for example, the `/etc/netmasks` file and the properties that are set for the `svc:/system/name-service/switch` SMF service.**

4. **Check whether the `rpcbind` daemon is running.**

   ```
   # rpcinfo -u localhost rpcbind
   ```

5. **Check whether the `nfsd` daemon is running.**

   ```
   # rpcinfo -u localhost nfs
   # ps -ef | grep mountd
   ```

Troubleshooting NIS Issues

The following information describes how to debug issues with the Network Information Service (NIS) (pronounced "niss" in this guide). Before attempting to debug a NIS server or a client problem, review Chapter 5, “About the Network Information Service” in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

This section contains the following topics:

- “Troubleshooting NIS Binding Issues” on page 62
- “Troubleshooting Issues That Affect a Single NIS Client” on page 62
- “Troubleshooting NIS Issues That Affect Multiple Clients” on page 66

Troubleshooting NIS Binding Issues

The following are common symptoms of NIS binding problems:

- Messages displaying that the ypbind daemon cannot find or communicate with a server.
- Messages that display server not responding.
- Messages that display NIS is unavailable.
- Commands on a client limp along in background mode or function much slower than normal.
- Commands on a client hang. Sometimes commands hang, even though the system as a whole seems fine and you can run new commands.
- Commands on a client crash with obscure messages or no messages at all.

Troubleshooting Issues That Affect a Single NIS Client

If only one or two clients are experiencing symptoms that indicate NIS binding difficulty, the problems probably are on those clients. However, if many NIS clients are failing to bind properly, the problem probably exists on one or more of the NIS servers. See "Troubleshooting NIS Issues That Affect Multiple Clients" on page 66.

The following are common NIS issues that affect a single client:

- ypbind **daemon not running on the NIS client**

  One client has problems, but the other clients on the same subnet are operating normally. On the problem client, run the ls -l command on a directory that contains files that are owned by many users (such as /usr), including files that are not in the client /etc/passwd file. If the resulting display lists file owners who are not in the local /etc/passwd file as numbers rather than names, the NIS service is not working on the client.

  These symptoms usually indicate that the client's ypbind process is not running. Verify whether the NIS client services are running as follows:

  ```
  client# svcs \*nis\*
  STATE          STIME    FMRI
  disabled       Sep_01   svc:/network/nis/domain:default
  disabled       Sep_01   svc:/network/nis/client:default
  ```

  If the services are in a disabled state, log in and become the root role, then start the NIS client service as follows:

```
client# svcadm enable network/nis/domain
client# svcadm enable network/nis/client
```

- **Missing or incorrect domain name**

  One client has problems and other clients are operating normally, but the `ypbind` daemon is running on that client. In this case, the client might have an incorrectly set domain.

  Run the `domainname` command on the client to determine which domain name is set:

  ```
  client# domainname
  example.com
  ```

  Compare the output with the actual domain name in the `/var/yp` directory on the NIS master server. As shown in the following example, the actual NIS domain is shown as a subdirectory in the `/var/yp` directory:

  ```
  client# ls -l /var/yp
  -rwxr-xr-x 1 root Makefile
  drwxr-xr-x 2 root binding
  drwx------ 2 root example.com
  ```

  If the domain name that is displayed in the output of the `domainname` command on the client is not the same as the server domain name that is listed as a subdirectory in the `/var/yp` directory, the domain name in the `config/domain` property of the `nis/domain` service is incorrect. Reset the NIS domain name. For instructions, see "How to Set a System's NIS Domain Name" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

  Note - The NIS domain name is case-sensitive.

- **NIS Client not bound to server**

  If your domain name is set correctly and the `ypbind` daemon is running, yet commands still hang, make sure that the client is bound to a server by running the `ypwhich` command. If you have just started the `ypbind` daemon, then run the `ypwhich` command. You might need to run the `ypwhich` command several times. Typically, the first time you run the command, it reports that the domain is not bound. The second time you run the command, it should proceed normally.

- **No NIS server available**

  If your domain name is set correctly and the `ypbind` daemon is running, but you receive messages indicating the client cannot communicate with the server, check the following:

  - Does the client have a `/var/yp/binding/`*domainname*`/ypservers` file that contains a list of NIS servers to bind to? To view the selected NIS servers, use the `svcprop`

  -p config/ypservers nis/domain command. If not, run the ypinit -c command to specify which NIS servers this client should bind to, in order of preference.

  - If the client does have a /var/yp/binding/*domainname*/ypservers file, are there enough servers listed, in case one or two servers become unavailable? To view the selected NIS servers, use the svcprop -p config/ypservers nis/domain command. If not, add additional servers to the list by running the ypinit -c command.
  - Do the selected NIS servers have entries in the /etc/inet/hosts file? To view the selected NIS servers, use the svcprop -p config/ypservers nis/domain command. If these servers are not in the local /etc/inet/hosts file, add the servers to the hosts NIS maps and rebuild your maps by running the ypinit -c or ypinit -s command. For information, see "Working With NIS Maps" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.
  - Is the name service switch set up to check the system's local hosts file, in addition to NIS? For more information, see Chapter 2, "About the Name Service Switch" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.
  - Is the name service switch set up to check files first for services and then rpc?

- ypwhich **displays are inconsistent**

  If you run the ypwhich command several times on the same NIS client, the resulting display varies because the NIS server changes. This behavior is normal. The binding of the NIS client to the NIS server changes over time when the network or the NIS servers are busy. Whenever possible, the network becomes stable at the point when all clients receive an acceptable response time from the NIS servers. As long as the client receives the NIS service, it does not matter where the service comes from. For example, one NIS server can receive NIS services from another NIS server on the network.

- **What to do when server binding is not possible**

  When local server binding is not possible, use the ypset option with the ypbind command to temporarily allow binding to another server on another network or subnet, if available. Note that to use the -ypset option, you must start the ypbind daemon by using either the -ypset or -ypsetme option. See the ypbind(8) man page.

  ```
  # /usr/lib/netsvc/yp/ypbind -ypset
  ```

  For another method, see "Binding to a Specific NIS Server" in *Working With Oracle Solaris 11.4 Directory and Naming Services: DNS and NIS*.

Caution - For security reasons, using the `-ypset` or `-ypsetme` option is not recommended. Only use these options for debugging purposes under controlled circumstances. Using the `-ypset` or `-ypsetme` option can result in serious security breaches. While the daemons are running, anyone can alter server bindings, which can permit unauthorized access to sensitive data. If you must start the `ypbind` daemon by using either of these options, kill the `ypbind` process after you have corrected the problem, then restart it without specifying these options.

Restart the `ypbind` daemon as follows:

```
# svcadm enable -r svc:/network/nis/client:default
```

See the `ypset`(8) man page.

- `ypbind` **daemon crashes**

  If the `ypbind` daemon crashes almost immediately each time you start it, look for a problem in the `svc:/network/nis/client:default` service log. Check the `rpcbind` daemon as follows:

  ```
  % ps -e |grep rpcbind
  ```

  If the `rpcbind` daemon is not available, is not running, or is not functioning normally, check the `svc:/network/rpc/bind:default` log file. See the `rpcbind`(8) and `rpcinfo`(8) man pages.

  From a system that is functioning normally, you can communicate with the `rpcbind` daemon on the problematic client.

  Run the following command from a functioning system:

  ```
  % rpcinfo client
  ```

  If the `rpcbind` daemon on the problematic system is fine, the following output is displayed:

  ```
  program    version netid     address           service    owner
  100007     3       udp6      ::.191.161        ypbind      1
  100007     3       tcp6      ::.135.200        ypbind      1
  100007     3       udp       0.0.0.0.240.221   ypbind      1
  100007     2       udp       0.0.0.0.240.221   ypbind      1
  100007     1       udp       0.0.0.0.240.221   ypbind      1
  100007     3       tcp       0.0.0.0.250.107   ypbind      1
  100007     2       tcp       0.0.0.0.250.107   ypbind      1
  100007     1       tcp       0.0.0.0.250.107   ypbind      1
  100007     3       ticlts    2\000\000\000     ypbind      1
  ```

```
100007    2       ticlts    2\000\000\000     ypbind     1
100007    3       ticotsord 9\000\000\000     ypbind     1
100007    2       ticotsord 9\000\000\000     ypbind     1
100007    3       ticots    @\000\000\000     ypbind     1
...
```

If no addresses are displayed (your system will have different addresses), the `ypbind` daemon was unable to register its services. Reboot the system and run the `rpcinfo` command again. If the `ypbind` processes are there and they change each time you attempt to restart the NIS service, reboot the system, even if the `rpcbind` daemon is running.

Troubleshooting NIS Issues That Affect Multiple Clients

If only one or two clients are experiencing symptoms that indicate NIS binding difficulty, the problems probably are on those clients. See "Troubleshooting Issues That Affect a Single NIS Client" on page 62. However, if several NIS clients are failing to bind properly, the problem most likely exists on one or more of the NIS servers.

The following are common NIS issues that can affect multiple clients:

- **`rpc.yppasswdd` considers a non-restricted shell that begins with `r` to be restricted**

  To resolve this problem, do the following:

  1. Create a `/etc/default/yppasswdd` file that contains a special string: `"check_restricted_shell_name=1"`.
  2. If the `"check_restricted_shell_name=1"` string is commented out, the `r` check does not occur.

- **Network or servers unreachable**

  NIS can hang if the network or NIS servers are so overloaded that the `ypserv` daemon cannot receive a response back from the client's `ypbind` process within the timeout period. NIS can also hang if the network is down.

  Under both of these circumstances, every client that is on the network experiences the same or similar problems. In most cases, the condition is temporary. The messages usually go away when the NIS server reboots and restarts the `ypserv` daemon, when the load on the NIS servers or the network decreases, or when the network resumes normal operations.

- **Server malfunction**

  Make sure the servers are up and running. If you are not physically near the servers, use the `ping` command to determine if the server is reachable.

- **NIS daemons not running**

If the servers are up and running, try to find a client that is behaving normally and run the ypwhich command on it. If the ypwhich command does not respond, kill it. Then, become the root role on the NIS server and check whether the NIS process is running as follows:

```
# ptree |grep ypbind
100759 /usr/lib/netsvc/yp/ypbind -broadcast
527360 grep yp
```

If neither the ypserv daemon (NIS server) nor the ypbind daemon (NIS client) daemons are running, restart them as follows:

```
# svcadm restart network/nis/client
```

If both the ypserv and ypbind processes are running on the NIS server, then run the ypwhich command. If the command does not respond, the ypserv daemon is probably hung and should be restarted.

On the server, restart the NIS service as follows:

```
# svcadm restart network/nis/server
```

- **Servers have different versions of a NIS map**

  Because NIS propagates maps among servers, occasionally you might find different versions of the same map on various NIS servers that are on the network. This version discrepancy is normal and acceptable if the differences do not last too long.

  The most common cause of map discrepancy is when normal map propagation is prevented. For example, a NIS server or router that is located between NIS servers is down. When all NIS servers and the routers between them are running, the ypxfr command should succeed.

  If the servers and routers are functioning properly, proceed as follows:

  - Check the ypxfr log output. See Example 26, "Logging ypxfr Command Output," on page 68.
  - Check the svc:/network/nis/xfr:default log file for errors.
  - Check the crontab file and yupxfr shell script.
  - Check the ypservers map on the master server.

- ypserv **process crashes**

  When the ypserv process crashes almost immediately and does not stay up even after repeated activations, the debugging process is virtually the same as the debugging process for ypbind crashes.

  First, run the following command to see if any errors are being reported:

  ```
  # svcs -vx nis/server
  ```

  Check for the existence of the rpcbind daemon as follows:

```
# ptree |grep rpcbind
```

Reboot the server if you do not find the daemon. Otherwise, if the daemon is running, run the following command and look for similar output:

```
# rpcinfo -p ypserver


program vers     proto  port  service
100000  4        tcp    111   portmapper
100000  3        tcp    111   portmapper
100068  2        udp   32813  cmsd
...
100007  1        tcp   34900  ypbind
100004  2        udp   731    ypserv
100004  1        udp   731    ypserv
100004  1        tcp   732    ypserv
100004  2        tcp  32772   ypserv
```

In the previous example, the following four entries represent the `ypserv` process:

```
100004     2     udp    731    ypserv
100004     1     udp    731    ypserv
100004     1     tcp    732    ypserv
100004     2     tcp   32772    ypserv
```

If there are no entries, and `ypserv` is unable to register its services with `rpcbind`, reboot the system. If there are entries, deregister the service from `rpcbind` before restarting `ypserv`. For example, you can deregister the service from `rpcbind` as follows:

```
# rpcinfo -d number 1
# rpcinfo -d number 2
```

number is the ID number that is reported by `rpcinfo` (`100004` in the preceding example).

EXAMPLE 26 Logging `ypxfr` Command Output

- If a particular slave server has problems updating the maps, log in to that server and run the `ypxfr` command interactively.

  If the command fails, a message about why it failed is displayed to enable you to fix the problem. If the command succeeds, but you suspect it has occasionally failed, create a log file on the slave server to enable the logging of messages as follows:

```
ypslave# cd /var/yp
ypslave# touch ypxfr.log
```

The output of the log file resembles the output of the `ypxfr` command when you run it interactively, with the exception that each line in the log file is time stamped. If you notice unusual ordering in the timestamps that is because it shows each time that the `ypxfr` command was actually run. If copies of `ypxfr` ran simultaneously but took differing amounts of time to finish, each copy might write a summary status line to the log file in a different order than when the command was run. Any pattern of intermittent failure shows up in the log.

Note - When you have resolved the problem, turn off logging by removing the log file. If you forget to remove it, the file continues to grow without limit.

- Check the `crontab` file and `ypxfr` shell script.

  Inspect the `root` `crontab` file and check the `ypxfr` shell script that it invokes. Typographical errors in these files can cause propagation problems. Failures to refer to a shell script within the `/var/spool/cron/crontabs/root` file or failures to refer to a map within any shell script can also cause errors.
- Check the `ypservers` map.

  Also, make sure that the NIS slave server is listed in the `ypservers` map on the master server for the domain. If it is not listed, the slave server still operates perfectly as a server, but `yppush` does not propagate map changes to the slave server.
- Update the maps on a broken slave server.

  If the NIS slave server problem is not obvious, you can perform a workaround while debugging the problem by using the `scp` or `ssh` command. These commands copy a recent version of the inconsistent map from any healthy NIS server.

  The following example shows how to transfer the problem map:

  ```
  ypslave# scp ypmaster:/var/yp/mydomain/map.\* /var/yp/mydomain
  ```

  In the previous example, the `*` character has been escaped in the command line so that it will be expanded on `ypmaster` instead of locally on `ypslave`.

♦ ♦ ♦ CHAPTER 4

Performing Network Diagnostics With the `network-monitor` Transport Module Utility

This chapter describes how to use the network diagnostics monitoring utility to detect misconfigured network resources and error conditions on your Oracle Solaris system.

This chapter contains the following topics:

What's New in Performing Diagnostics With the Network Monitor

For existing customers, this section highlights the key changes in this release.

In this release, an FMA alert is raised whenever a misconfigured datalink, which could be either a mismatch of VLAN ID or MTU, is detected. An FMA alert is also the link state for a datalink changes. The alert notification is performed by the standard FMA notification agents, for example, `syslog`, email, and so on. Use the `fmadm list-alert` command to obtain a list of the active FMA alerts on the system. For more information, see "Obtaining Information About the Active FMA Alerts on a System" on page 74 and the `fmadm`(8) man page.

Overview of the `network-monitor` Transport Module Utility

The `network-monitor` (also referred to as the *monitor* in this chapter) is a fault manager daemon (`fmd`) transport module utility that you use to perform network diagnostics on your Oracle Solaris system. The utility monitors network resources and reports conditions that might lead to limited or degraded network functionality. When the monitor utility detects an abnormal network condition, an alert is generated. You can retrieve information about alerts by using the `fmadm` command. See "Obtaining Information About the Active FMA Alerts on a System" on page 74. The monitor utility does not perform any further diagnosis of the error condition, nor does it perform any additional recovery actions. For more information, see the `network-diagnostics`(5) man page.

You control the monitor utility by setting property values that are stored in the `svc:/network/diagnostics` SMF service instance. For more information, see "Controlling the Use of Probes Through the `svc:/network/diagnostics` SMF Service" on page 75.

How Datalink MTU Mismatch Errors Are Detected

This error condition occurs when there is a mismatch between the MTUs of two peer datalinks. This type of mismatch can result in dropped frames because one datalink might transmit frames that are larger than the peer datalink can receive. The monitor utility attempts to detect any datalinks on the local system with MTUs that are set too high. Datalinks are verified upon system start-up and then again when an MTU change occurs.

MTU verification is performed by using either the Link-Layer Discovery Protocol (LLDP) or the Internet Control Message Protocol (ICMP) probe method. A peer host that has the LLDP service enabled can include MTU details in the information exchange. The utility performs MTU verification by extracting peer MTU information. When LLDP information is unavailable, the monitor utility attempts to verify the MTU by transmitting a series of ICMP probes of different sizes until the datalink MTU is reached. A mismatch is flagged if the utility consistently fails to reach a target by using maximum-sized probes.

How Datalink VLAN ID Mismatch Errors Are Detected

VLANs are used to group end-system hosts into the same broadcast domain. The hosts on a VLAN might not reside on the same LAN, but even if they do, each host can communicate with another host by using Layer 2 (L2) protocols. Conversely, hosts that reside on the same LAN but different VLANs cannot communicate by using L2 protocols. Each host that resides on a VLAN uses a network interface to communicate with the other hosts on the VLAN. VLANs are identified by VLAN identifiers (VIDs) that are exported by LLDP daemons over the relevant network interfaces to their peers. These peers are typically network devices, for example, switches that use a VID to forward data packets to respective hosts.

Hosts might not receive the intended packets if the VIDs are not configured correctly on the relevant network interfaces. The monitor utility captures this type of VLAN ID mismatch because it verifies the VLAN ID information periodically and whenever the VLAN information is modified at system boot time. If the VLAN ID for an interface changes, the appropriate alerts are generated. Because the VLAN information is verified by using LLDP packets, the peer host needs to have the LLDP service enabled. For information about LLDP, see Chapter 7, "Exchanging Network Connectivity Information With Link Layer Discovery Protocol" in *Managing Network Datalinks in Oracle Solaris 11.4*.

Managing the `network-monitor` Module

The `fmadm` command reports the current status of the monitor utility. The status is displayed as `active` when the utility is performing fault monitoring, as shown in the following example.

```
# fmadm config

  MODULE                    VERSION STATUS  DESCRIPTION
  cpumem-retire             1.1     active  CPU/Memory Retire Agent
  disk-diagnosis            0.1     active  Disk Diagnosis engine
  ...
  network-monitor           1.0     active  Network monitor
```

The `/usr/lib/fm/fmd/plugins/network-monitor.conf` configuration file has a property called `enable` that controls the state of the `network-monitor`. The monitor utility is enabled by default. To disable the monitor, set the `enable` property in the file to `false`, as shown in the following example.

```
# enable
#
# Enable/disable the network-monitor.
#
# setprop enable false
```

The monitor utility will be disabled upon reboot.

Obtaining Information About the Active FMA Alerts on a System

Starting with Oracle Solaris 11.4, datalink FMA alerts are supported. To obtain a list of the active FMA alerts on a system, use the `fmadm` command as follows:

```
# fmadm list-alert
--------------- ------------------------------------  -------------- ---------
TIME            EVENT-ID                              MSG-ID         SEVERITY
--------------- ------------------------------------  -------------- ---------
Mar 29 17:35:47 a8413d9a-1200-49e9-ab89-ceac07842505  NET-8000-39    Minor

Problem Status    : open
Diag Engine       : software-diagnosis / 0.1
System
    Manufacturer  : unknown
    Name          : unknown
    Part_Number   : unknown
    Serial_Number : unknown

System Component
    Manufacturer  : Sun Microsystems
    Name          : Sun Fire X4600 M2
    Part_Number   : To Be Filled By O.E.M.
    Serial_Number : 0807AM0213
    Firmware_Manufacturer : XXX Megatrends Inc.
    Firmware_Version      : (BIOS)08140111
    Firmware_Release      : (BIOS)04.14.2016
    Host_ID               : 00824c5c
    Server_Name           : TestSystem

----------------------------------------
Suspect 1 of 1 :
   Problem class : alert.oracle.solaris.net.datalink.link_down
   Certainty   : 100%

   Resource
     FMRI             : "obj:///:id=datalink/ixgbe0"
     Status           : Active

Description : Network connectivity via datalink ixgbe0 has been lost.

Response    : No automated response.

Impact      : Network communication via datalink ixgbe0 is no longer possible.

Action      : Check the networking cable, switch port, and switch
```

```
configuration. Please refer to the associated reference document
at http://support.oracle.com/msg/NET-8000-39 for the latest
service procedures and policies regarding this diagnosis.
```

Viewing Statistics for the network-monitor Module With the fmstat Command

The fmstat command reports fault management module statistics. You can also use the command to view statistics for diagnosis engines and agents that are currently participating in fault management such as the network-monitor transport module utility.

To view statistics that are kept by a specific fault management module, use the following command syntax:

```
# fmstat -m module
```

where -m *module* specifies the fault management module.

For example, to view the statistics for the network-monitor utility, use the following command:

```
# fmstat -m network-monitor
    linkstate.enabled true             operating status for linkstate
                NAME VALUE            DESCRIPTION
mtu-mismatch.allocerr 0                memory allocation errors
mtu-mismatch.enabled true             operating status for mtu-mismatch
mtu-mismatch.nprobes 7                number of transmitted ICMP probes
mtu-mismatch.procerr 0                errors processing datalinks
          sysev_drop 0                number of dropped sysevents
vlan-mismatch.enabled true             operating status for vlan-mismatch
```

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

To obtain a list of the modules that participate in fault management, use the fmadm command. For information, see the fmadm(8) man page.

Controlling the Use of Probes Through the svc:/network/diagnostics SMF Service

The types of diagnostics that the monitor performs are controlled by values that are stored in the policy/allow_probes property of the svc:/network/diagnostics SMF service. This property determines whether probe packets can be transmitted by diagnostic agents for the purpose of monitoring and reporting network problems. To set or change values for this property, use the svccfg command. Valid values are true and false. By default, the property is set to true. For information, see the svccfg(8) and network-diagnostics(5) man pages.

EXAMPLE 27 Disabling the Transmission of Diagnostic Probes

The following example shows how you can disable the transmission of diagnostic probes by setting the policy/allow_probes property of the svc:/network/diagnostics SMF service to false. You must refresh the SMF service after changing the default value for the changes to take effect.

```
# svccfg -s network/diagnostics setprop policy/allow_probes = boolean: false
# svccfg -s network/diagnostics refresh
```

Index