SunFDDI™ SBus Adapter 5.0 User's Guide

Sun

The Network Is the Computer™

Please
Recycle

Adobe PostScript™

Regulatory Compliance Statements

Your Sun product is marked to indicate its compliance class:

- Federal Communications Commission (FCC) — USA
- Department of Communications (DOC) — Canada
- Voluntary Control Council for Interference (VCCI) — Japan

Please read the appropriate section that corresponds to the marking on your Sun product before attempting to install the product.

FCC Class A Notice

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference.
2. This device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Shielded Cables: Connections between the workstation and peripherals must be made using shielded cables in order to maintain compliance with FCC radio frequency emission limits. Networking connections can be made using unshielded twisted-pair (UTP) cables.

Modifications: Any modifications made to this device that are not approved by Sun Microsystems, Inc. may void the authority granted to the user by the FCC to operate this equipment.

FCC Class B Notice

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference.
2. This device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/television technician for help.

Shielded Cables: Connections between the workstation and peripherals must be made using shielded cables in order to maintain compliance with FCC radio frequency emission limits. Networking connections can be made using unshielded twisted pair (UTP) cables.

Modifications: Any modifications made to this device that are not approved by Sun Microsystems, Inc. may void the authority granted to the user by the FCC to operate this equipment.

DOC Class A Notice - Avis DOC, Classe A

This Class A digital apparatus meets all of requirements the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

DOC Class B Notice - Avis DOC, Classe B

This Class B digital apparatus meets all of requirements the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

VCCI 基準について

第一種VCCI基準について

第一種VCCIの表示があるワークステーションおよびオプション製品は、第一種情報装置です。これらの製品には、下記の項目が該当します。

> この装置は、第一種情報装置(商工業地域において使用されるべき情報装置)で商工業地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会(VCCI)基準に適合しております。したがって、本製品を、住宅地域または住宅地域に隣接した地域でご使用になりますと、ラジオ、テレビジョン受信機等に受信障害を与えることがあります。
>
> 取り扱い説明書に従って正しくお取り扱いください。

第二種VCCI基準について

第二種VCCIの表示 VCI があるワークステーションおよびオプション製品は、第二種情報装置です。これらの製品には、下記の項目が該当します。

> この装置は、第二種情報装置(住宅地域または住宅地域に隣接した地域において使用されるべき情報装置)で住宅地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会(VCCI)基準に適合しております。しかし、本製品を、ラジオ、テレビジョン受信機に近接してご使用になりますと、受信障害の原因となることがあります。
>
> 取り扱い説明書に従って正しくお取り扱いください。

Contents

Figures

Tables

Preface

The *SunFDDI™ SBus Adapter 5.0 User's Guide* describes how to turn your system into an FDDI station attached to an FDDI network. It is intended for experienced network administrators who are familiar with the Solaris 1.*x* and 2.*x* operating environments. The SunFDDI SBus Adapter 5.0 is supported on Sun-4d, Sun-4m and Sun-4u system architectures.

The *SunFDDI™ SBus Adapter 5.0 User's Guide* is divided into two parts:

- *Part 1 — "Installing and Configuring SunFDDI,"* describes how to turn your machine into a SunFDDI station. It tells you how to install the SunFDDI SBus card, how to install and configure the SunFDDI device driver, and how to tune your FDDI station to get the best performance. It also includes troubleshooting advice to help you detect and resolve problems with your FDDI network.

- *Part 2 — "Planning and Implementing SunFDDI Networks,"* includes a brief overview of the FDDI protocols, and describes common FDDI network topologies. It tells you how to use the SunFDDI network utilities, how to use a SunNet Manager™ Console to manage your SunFDDI stations, how to develop applications over SunFDDI, and how to boot a diskless station across an FDDI connection.

The connector type has been changed for SunFDDI 5.0 to an SC-type connector. If you are connecting the SunFDDI 5.0 card to a network that has a MIC connector, an SC-MIC converter cable is required.

If you are unfamiliar with FDDI network terminology, or if you are implementing an FDDI network for the first time, read Part 2 first.

How This Book Is Organized

The SunFDDI SBus Adapter 5.0 User's Guide is organized as follows:

Part 1 — "Installing and Configuring SunFDDI"

Chapter 1, "SunFDDI Overview," describes the SunFDDI 5.0 implementation of the FDDI protocols and includes a list of the specifications to which it conforms.

Chapter 2, "Hardware Installation," tells you where to find detailed instructions on how to install your SunFDDI SBus card in your machine and describes how to connect the FDDI cable.

Chapter 3, "Installing SunFDDI in a Solaris 2.x Environment," describes how to install the SunFDDI device driver and utilities on machines running a Solaris™ 2.*x* environment. It includes a description of the post-installation scripts that configure your machine as an FDDI station.

Chapter 4, "Installing SunFDDI in a Solaris 1.x Environment," describes how to install the SunFDDI device driver and utilities on machines running a Solaris 1.*x* environment. It includes a description of the post-installation scripts that configure your machine as a FDDI station.

Chapter 5, "Improving Network Performance," describes the configurable network parameters and provides general advice on how to obtain the best performance from your network.

Chapter 6, "Troubleshooting and Diagnostics," tells you how to detect and resolve problems with your FDDI network.

Part 2 — "Planning and Implementing SunFDDI Networks"

Chapter 7, "FDDI Network Architecture," provides an overview of the FDDI network model and describes how FDDI networks operate.

Chapter 8, "FDDI Network Topologies," describes various pure FDDI network topologies and discusses their relative strengths and weaknesses. It also describes how to implement routing between mixed FDDI and Ethernet networks.

Chapter 9, "Using the SunFDDI Network Utilities," describes the network utilities delivered with SunFDDI and tells you how to use them to recover network statistics.

Chapter 10, "Managing FDDI Stations Using SunNet Manager," describes how to install the SunNet Manager agents for SunFDDI and how to set up the SunNet Manager console to manage them.

Chapter 11, "Developing Applications over SunFDDI," describes how to create applications that run over SunFDDI, using the DLPI interface for a Solaris 2.*x* environment.

Chapter 12, "Setting Up Servers and Diskless Clients," describes how to set up a server so that you can boot and run diskless clients over an FDDI network.

Appendix A, "Frequently Asked Questions About SunFDDI," contains a list of common questions and their responses.

UNIX Commands

This document may not include specific software commands or procedures. Instead, it may name software tasks and refer you to operating system documentation or the handbook that was shipped with your new hardware.

The type of information that you might need to use references for includes:

- Shutting down the system
- Booting the system
- Configuring devices
- Other basic software procedures

See one or more of the following:

- *Solaris 2.x Handbook for SMCC Peripherals* contains Solaris™ 2.x software commands.

- On-line AnswerBook™ for the complete set of documentation supporting the Solaris 2.x software environment.

- Other software documentation that you received with your system.

Typographic Conventions

The following table describes the typographic changes used in this book.

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`
AaBbCc123	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

Shell	Prompt
C shell	`machine_name%`
C shell superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`

Related Documents

For a more detailed description of FDDI technology and the relevant FDDI protocols, see these documents:

Title	Author/Publisher	Part Number
Handbook of Computer-Communications Standards, Volume 2,	William Stallings, Macmillan Publishing Company: 1987	NA
FDDI Technology and Applications	Edited by Sonu Mirchandani and Raman Khanna, John Wiley & Sons: 1993	NA
FDDI A High Speed Network	Amit Shah and G. Ramakrisnan, Prentice Hall, Inc.: 1994	NA
STREAMS DLPI Specification		800-6915

Ordering Sun Documents

The SunDocs Order Desk is a distribution center for Sun Microsystems technical documentation. You can use major credit cards and company purchase orders. You can order documentation in the following ways:

Fax 1-800-944-0661	Call 1-800-873-7869
World Wide Web: `http://www.sun.com/sunexpress/`	

Sun Welcomes Your Comments

Please use the *Reader Comment Card* that accompanies this document. We are interested in improving our documentation and welcome your comments and suggestions.

If a card is not available, you can email or fax your comments to us. Please include the part number of your document in the subject line of your email or fax message.

- Email: `smcc-docs@sun.com`

- Fax: SMCC Document Feedback
           1-415-786-6443

*Part 1 — Installing and Configuring
SunFDDI*

SunFDDI Overview 1≣

This chapter describes the Sun FDDI (Fiber Distributed Data Interface) implementation of the FDDI protocols, including a list of the ANSI/FDDI standards to which it conforms. See Chapter 7, "FDDI Network Architecture" for more information on FDDI architecture.

Feature Summary

The SunFDDI SBus Adapter 5.0 is a combination of hardware and software that turns your system into an FDDI station. There are two hardware options:

- **SunFDDI** — single attached station (SAS) on fiber
- **SunFDDI Dual** — dual attached station (DAS) on fiber

SunFDDI is a high-speed networking product that provides significantly greater bandwidth (up to 10 times) compared to 10-megabit Ethernet networks. It is designed for Sun-4d, Sun-4m and Sun-4u system architectures running Solaris 1.*x* and Solaris 2.3, 2.4, 2.5, 2.5.1, or a later release.

SunFDDI does the following:

- Implements FDDI single-attached station (SunFDDI) and FDDI dual-attached station (SunFDDI Dual)

- Provides connection to multimode fiber (SunFDDI and SunFDDI Dual) networks

- Supports data transfer rates of up to 100 Mbps

- Provides full support for existing SunOS 5.*x* and 4.*x* operating system network services and utilities.

- Provides power-up self test and system diagnostic tests

- Provides high reliability and availability through distributed, automatic recovery

- Supports up to four SunFDDI SBus cards installed in one SBus and up to eight SunFDDI SBus cards installed in one machine

- Supports 12.5 Mbps and 25 Mbps SBus transfer rates

- Supports 62.5/125 micrometer, multimode fiber (MMF)

- Provides SunNet Manager agents for the effective management of FDDI stations and networks from a centralized SunNet Manager Console; (requires SunNet Manager 2.2 or later in addition to SunFDDI)

- Complies with the ANSI X3T9.5 standard for single-attached LAN fiber rings and with revision 7.3 of the ANSI/FDDI SMT specification

- Complies with revision 1496-1993 of the IEEE SBus specification

FDDI Conformance

SunFDDI conforms to the following standards and specifications:

- ANSI/FDDI Media Access Control (MAC) X.3.139-1987
- ANSI/FDDI physical sublayer (PHY) X3.148-1988
- ANSI/FDDI physical medium dependent (PMD) X3.166-1990
- ANSI/FDDI Station Management (SMT) X3.299 R7.3 (formally R7.2.99)
- IEEE SBus 1496-1993

Hardware Installation 2≣

This chapter tells you where to find step-by-step procedures for installing SBus cards. It describes the convention used to assign device names to SunFDDI SBus cards, and tells you how to connect fiber optic cables.

Warning – For your personal safety and to protect your machine, please observe the following precautions when you install an SBus card in a SPARC™ workstation or server:

Before opening your machine, always ensure that the power switch is off (O position). When the power is off, the green light on the front of the machine is unlit and the fan in the power supply is not running. Always take care to protect your equipment from the effects of static electricity.

A lithium battery is molded into the `NVRAM–TOD` chip on the main logic board. This battery is not a customer replaceable part. Do not attempt to disassemble it or recharge it.

≡ *2*

Device Instances and Device Names

You can install a SunFDDI SBus card in any available master SBus slot. SunFDDI supports up to four SunFDDI SBus cards installed in one SBus, and up to eight SunFDDI SBus cards installed in one machine.

SunFDDI 5.0 SBus cards are assigned device names of the form nf*<inst>*, where the instance number *<inst>* is determined by the number and relative positions of the cards installed. When you install a SunFDDI card in your machine for the first time, it has instance number 0 and a device name nf0. If multiple cards are installed, the device name nf0 is associated with the card in the lowest numbered slot, the device name nf1 is associated with the card in the next highest numbered slot, and so on.

Refer to the hardware installation manual that came with your system for detailed instructions on how to identify the SBus slots in your machine.

Note – For a machine running Solaris 2.*x*, if a SunFDDI SBus card is moved from its initial slot to a different one, the operating system will find the device in the new slot and assume that it is a new instance of the device. As a result, the operating system will assign the next instance number to it and there will appear to be two devices installed in the machine.

Installation Items

Before proceeding with the installation, make sure you have received the following items:

- A SAS or DAS SunFDDI SBus adapter
- FDDI converter cable
- Disposable grounding wrist strap
- CD-ROM containing SunFDDI 5.0 software
- SunFDDI SBus Adapter 5.0 User's Guide

Contact your sales representative if any of these items are missing.

Installing SunFDDI SBus Cards

! **Caution** – The SunFDDI SBus card is sensitive to static electricity. Always use the wrist strap supplied with SunFDDI when handling the card and ensure that the wrist strap is properly grounded. Handle the SunFDDI SBus card by the edges, and avoid touching any of the components.

Refer to the hardware installation manual that came with your system for detailed instructions on how to install an SBus card in your machine.

There are no user-configurable jumpers or switches on the SunFDDI SBus card. Altering any of the components on the card may affect its operation adversely and will render any product warranty void.

▼ To Remove the Plastic Retainer from the Card

1. **Bend one leg of the plastic retainer slightly to the outside until it unhooks from the card.**

2. **Unhook the other leg of the plastic retainer from the card.**
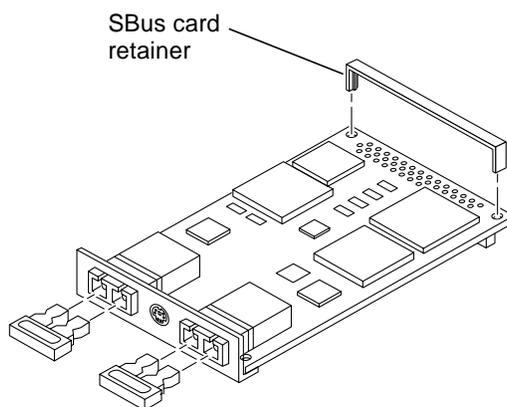
Figure 2-1   Removing Plastic Retainer From Rear of Card

Connecting Fiber Optic Cables

Fiber optic cable is fragile and should not be subjected to mechanical stress or impact. Always observe the following precautions when handling fiber optic cable and connectors:

- Fiber optic cable connectors consist of a ceramic or steel ferrule through which the optical fiber passes. Do not touch the tip of the ferrule, as marring or depositing oil or dirt on its delicate surface will destroy its ability to transmit light.

- Use the protective caps provided on the fiber optic cable connectors and optic data links whenever the cable is not connected.

- Fiber optic cable has a limited turn radius. Do not bend it in a way that could damage the fiber or cause signal loss.

1. **Unpack the cable, remove the plastic protective caps or plugs from each end, and remove the plug from the transceiver unit on the card.**

2. **Make sure the fiber optic cable is properly oriented with its connector on the SunFDDI card.**
   Most fiber optic cables have raised keys on one side of the connectors (see Figure 2-2). There are also keyway notches on the inside of the connector on the card; the fiber optic cable should be oriented so that the keys on the cable connectors line up with the keyway notches on the card. Also make sure to connect the cable to the proper port, either Port A or Port B (see Figure 2.2).

Warning – Some cables do not possess raised keys to help you properly orient the connection.
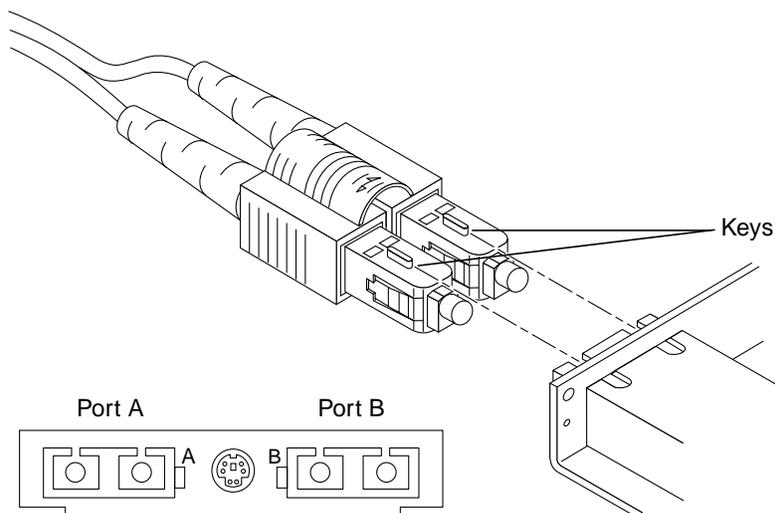
Figure 2-2   Aligning SC Type Fiber Optic Cable Connectors

3. **Once you have verified that the fiber optic cable and the connector on the SunFDDI card are properly oriented, plug the cable into the card connector.**
   You should hear a distinctive "click" sound, telling you that the cable is properly mated with the card connector.

Note – If you are connecting the SunFDDI 5.0 card to a network that has a MIC connector, a converter cable is required. Earlier versions of SunFDDI used MIC connectors.

• Sun has provided converter cables that will allow existing MIC-based networks to connect to the SunFDDI 5.0 adapter without making cabling changes. Networks that have already been designed for SC connections do not need the converter cable. There will be under one decibel of signal integrity loss when using the adapter. Borderline applications may need to reconfigure the network directly for SC connection if the signal loss materially affects the error rate.

Cleaning the Fiber Optic Cable

Fiber optic cable connectors can be cleaned if they become contaminated with dirt, dust or oil, which can cause signal noise on the network.

1. **Gently wipe the end of the ferrule with an alcohol-moistened cotton swab.**
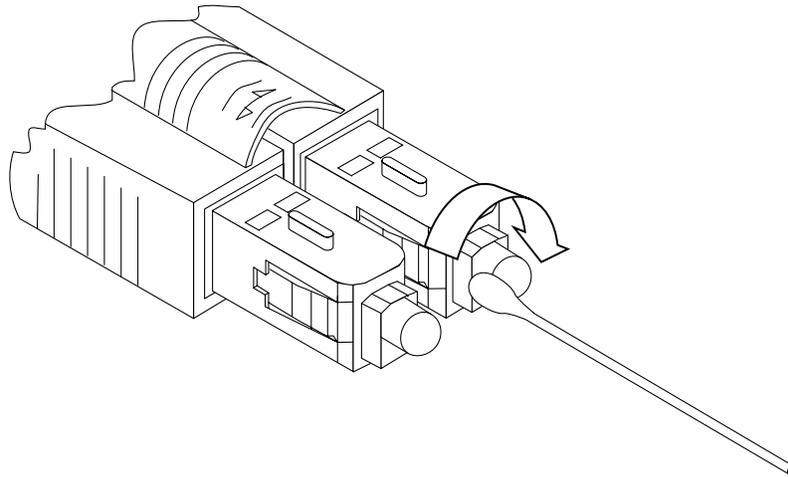
Figure 2-3   Cleaning the End of a Fiber Optic Cable

2. **Blow clean, dry compressed air around the end of the connector and around the ferrule to dislodge loose dust and grit.**

3. **Clean the adapter by moistening the end of a pipe cleaner in alcohol and running the pipe cleaner though the adapter.**
   Repeat with a dry pipe cleaner.

4. **Blow the adapter with clean, dry compressed air to finish drying the clean adapter.**

Link Status Indicator (Diagnostic LED)

A link status indicator is mounted on the SunFDDI SBus card. The color of the LED indicates the current status of the connection to the FDDI network as follows:

- **red**
  Either the driver is loaded and the interface is *not* configured
  or the driver has not been loaded.

- **amber**
  Interface configured. No valid line state to an active FDDI network exists.

- **green**
  Connection established to an active FDDI network.

- **blinking green**
  In a dual-ring environment: Invalid connection.
  In a dual-homing environment: Normal. The connection is in standby mode.

Refer to Chapter 6, "Troubleshooting and Diagnostics," for more information on diagnosing problems with your FDDI network.

≡ 2

Installing SunFDDI in a Solaris 2.x Environment 3≡

Installation Requirements	*page 12*
Declaring IP Addresses for SunFDDI	*page 12*
Installing and Configuring SunFDDI	*page 13*
Installing SunNet Manager Agents for SunFDDI	*page 20*
Removing SunFDDI	*page 22*

This chapter assumes that you have already installed the SunFDDI SBus card in your machine and that you are loading the software from a local CD-ROM drive.

This chapter tells you how to install the SunFDDI device drivers, utilities, and SunNet Manager agents in a Solaris 2.*x* environment. It includes instructions on how to check your installation and how to remove the SunFDDI software, if necessary. If your machine is running a Solaris 1.*x* environment, see Chapter 4, "Installing SunFDDI in a Solaris 1.x Environment" for instructions.

Note – If you installed the SunFDDI 5.0 SBus card into a Netra™ 150 server, *do not* use the software procedures in this user's guide. After you have installed the SunFDDI 5.0 card in the Netra 150 server, follow the instructions given in the Netra User's Manual to complete the software portion of the installation process.

11

≡ *3*

Installation Requirements

Hardware and Software Platforms

SunFDDI is designed for Sun-4d, Sun-4m and Sun-4u system architectures (32 Mbytes RAM minimum) running a Solaris 1.*x* or Solaris 2.3, 2.4, 2.5, 2.5.1 or later environment.

Disk Space

You need a machine with 1600 KBytes of disk space to install SunFDDI in a Solaris 2.*x* environment.

Host names and IP Addresses

You must assign a unique host name and IP address to each SunFDDI device installed in your machine. If you are installing more than one interface in a single machine, each interface must be connected to a different subnet of a network.

Declaring IP Addresses for SunFDDI

You must assign a unique IP address to each SunFDDI interface. If you have more than one networking interface installed on your machine—either multiple SunFDDI interfaces or some other LAN interface (for example, an Ethernet interface)—each interface must be connected to a different subnet.

Before installing the SunFDDI software, update the files on your NIS/NIS+ server to assign IP addresses and host names for the SunFDDI interfaces that you are going to configure. If you are not using a name service, update the `/etc/hosts` file on each machine in the network to add the IP addresses and host names of the stations attached to the FDDI ring.

The installation script updates the `/etc/hosts` file on the local machine and enters the IP addresses and host names assigned to the SunFDDI interfaces in which it is installed.

Installing and Configuring SunFDDI

Use `pkgadd(1M)` to install unbundled software on machines running a Solaris 2.*x* environment. See the *Solaris 2.5 Software and AnswerBook Packages Administration Guide* for detailed information on installing software products using `pkgadd`.

▼ **To Load and Mount the CD-ROM**

1. **Log in as** `root` **or become** `superuser`.

2. **Place the CD-ROM in its caddy and load the caddy into the CD-ROM drive.**

3. **Mount the CD-ROM on a local directory.**

- If the Volume Manager (`vold`) is running on your machine, then the CD-ROM is mounted automatically under `/cdrom/sun_fddi_5_0_sparc`.

- If the Volume Manager (`vold`) is *not* running on your machine, create a directory called `/cdrom/sun_fddi_5_0_sparc` and mount the CD-ROM manually:

```
# mkdir /cdrom/sun_fddi_5_0_sparc
# /usr/sbin/mount -o ro -F hsfs /dev/dsk/c0t6d0s0 \
/cdrom/sun_fddi_5_0_sparc
```

For detailed instructions on how to load software from a CD-ROM drive mounted on a remote directory, see the *Solaris 2.5 Software and AnswerBook Packages Administration Guide.*

After mounting the CD-ROM, you will see the following directories in the /cdrom/sun_fddi_5_0_sparc directory:

- sol_1.1.x
  Contains the FDDI software packages for SunOS 4.1.3, 4.1.3_U1, 4.1.4 and the examples directory.

- sol_2.3
- sol_2.4 -> sol_2.5
- sol_2.5
  Contain the FDDI software packages for Solaris 2.3, 2.4, 2.5, 2.5.1 and the examples directory.

▼ To Install SunFDDI Using pkgadd

The instructions below are for SunFDDI on a Solaris 2.5 environment. If you are installing SunFDDI on an environment other than Solaris 2.5, replace "**sol_2.5**" with your specific operating environment, as in **sol_2.3** or **sol_2.4**.

1. **Log in as** root **or become** superuser.

2. **Start** pkgadd **by typing:**

```
# /usr/sbin/pkgadd —d /cdrom/sun_fddi_5_0_sparc/sol_2.5
```

3. **Enter the number that corresponds to the package you want to install and press Return. Respond to any prompts that appear on the command-line.**
   There are two packages associated with SunFDDI. They contain the device drivers and utilities used to manage your SunFDDI station.

```
The following packages are available:
    1 SUNWnfr   SunFDDI (Driver)
                (sparc) 5.0
    2 SUNWnft   SunFDDI (Man Pages/Utilities)
                (sparc) 5.0

Select package(s) you wish to process (or "all" to process all
packages). (default: all) [?,??,q]:
```

▼ To Install the Device Driver (SUNWnfr)

The package SUNWnfr contains the SunFDDI device driver and the SunNet Manager agent daemon and start-up scripts. By default, the base directory *<basedir>* for this package is /. You can change the default base directory for this package when you start pkgadd(1M) with the –R option. This package must be installed on every machine that runs SunFDDI.

An installation script is executed automatically when you install the SunFDDI device driver (SUNWnfr). This script checks that the hardware is installed and displays the number of SunFDDI interfaces that it finds. If there is no hardware installed, the script halts the installation process without installing the software.

1. Indicate whether you are installing the driver for a diskless client.

- Type y if you want to install the driver in the /export/root/*<client>* file system for a diskless client
- Type n if you are installing the driver for the machine on which you are running the script.
  You must use pkgadd with the –R option to change the base directory when you install the driver for a diskless client.

```
Is this install for a diskless client [n] [y,n,?]
```

(See Chapter 12, "Setting Up Servers and Diskless Clients" for more information on booting diskless clients across an FDDI network.)

The number of SunFDDI interfaces found is displayed and you are asked how many interfaces you want to configure.

2. Enter a number in the range 1 to 16 or press Return to accept the default.

```
How many FDDI (nf) interfaces do you want to configure?
```

3. Type the host name for which you want the interface associated.

```
What host name do you want to use for interface 1: hysop-nf
```

4. Type the IP address that corresponds to the host name or press Return to accept the default IP address.
If an IP address is already associated with the host name (either in the NIS/NIS+ database or in the /etc/hosts file) the script offers it as a default IP address. You must assign a unique IP address to each SunFDDI interface. If you have multiple IP interfaces installed on one machine, each interface must be attached to a different FDDI subnetwork.

```
What ip address do you wish to use for hysop-nf [<ip_address>]:
```

5. Repeat Steps 3 and 4 until you have assigned a host name and IP address for each SunFDDI SBus card installed in your machine.

6. Indicate whether you want the SunNet Manager agents for SunFDDI to start automatically when you reboot the machine.

- Type y to start the SunNet Manager agents for SunFDDI automatically

- Type n, or press Return, if you are not using SunNet Manager to manage your network; if you are running SunNet Manager agents on another machine; or if you want to start the SunNet Manager agents manually.

```
Do you want to start the SunNet Manager daemons for SunFDDI
at boot time? [n] [y,n,?,q] y
```

7. Type y to accept the execution of the installation scripts with superuser permission.

```
This package contains scripts which will be executed with superuser
permission during the process of installing the package.

Do you want to continue with the installation [y,n,?] y
```

The script displays a list of the files that it installs, loads the driver, and configures the IP interfaces automatically.

If the installation is successful, your SunFDDI interfaces are now active. The link status indicator (diagnostic LED) will be:

- **Amber** if the interface is configured and no valid line state to an active FDDI network exists

- **Green** if the interface is configured and connected to an active network

▼ **To Install the Utilities, man Pages, and SNM Agents (**SUNWnft**)**

The SUNWnft package contains the SunFDDI utilities and man pages. By default, the base directory *<basedir>* for this package is /opt. You can change the default base directory when you start pkgadd(1M) with the −R option. This package can be installed on a server and shared between multiple machines, if required.

An installation script is executed automatically when you install the SunFDDI utilities and man pages (SUNWnft):

1. **Type** y **to accept the installation of files with** setuid/setgid **permission.** If you are installing the SUNWnft package for the first time, this step will not occur.

```
Do you want to install these setuid/setgid files [y,n,?,q] y
```

2. **Type y to accept the execution of the installation scripts with superuser permission.**

```
This package contains scripts which will be executed with superuser
permission during the process of installing the package.

Do you want to continue with the installation [y,n,?] y
```

≡ *3*

▼   To Finish the Installation

1. **Type** q **to stop** pkgadd **when the list of available packages is displayed again.**

```
The following packages are available:
    1 SUNWnfr   SunFDDI (Driver)
                (sparc) 5.0
    2 SUNWnft   SunFDDI (Man Pages/Utilities)
                (sparc) 5.0

Select package(s) you wish to process (or "all" to process all
packages). (default: all) [?,??,q]:
```

2. **Eject the CD-ROM from the CD-ROM drive.**

- If Volume Manager (vold) is running on your machine, then the CD-ROM is unmounted automatically when it is ejected:

```
# eject cdrom
```

- If Volume Manager (vold) is not running on your machine, unmount the CD-ROM before you eject and remove the directory that you created:

```
# /usr/sbin/umount /cdrom/sun_fddi_5_0_sparc
# eject cdrom
# rmdir /cdrom/sun_fddi_5_0_sparc
```

3. **Reboot your machine to ensure correct system operation.**

▼ To Check the Installation

1. Use `netstat(1M)` **to check for SunFDDI interfaces.**

SunFDDI Interface
nf*<inst>* ---->

```
% /usr/bin/netstat -i
Name Mtu    Net/Dest Address   Ipkts Ierrs Opkts Oerrs Collis
Queue
lo0  8232  loopback localhost  21    0     21    0     0      0
le0  1500  our-lan  hysop      2146  0     950   1     13     0
nf0  4352  our-fddi hysop-nf   1086  0     907   0     0      0
```

2. Use `ifconfig(1M)` **to check the current state of the SunFDDI interfaces.**

```
% /usr/sbin/ifconfig nf0
nf0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 4352
     inet <host_address> netmask <netmask> broadcast <ip_address>
     ether <mac_address>
```

3. Use `ping(1M)` **to verify that you can send and receive data across the FDDI connection.**
If your SunFDDI station is attached to an active FDDI network, try to access a remote station.

```
% /usr/sbin/ping hemlock-nf
hemlock-nf is alive
```

Your SunFDDI station should now be active, and you should be able to send and receive data across an FDDI connection.

To obtain optimum performance from your FDDI network, you may need to tune the configurable network parameters. See Chapter 5, "Improving Network Performance," for detailed instructions.

If you have problems configuring or using your SunFDDI station, see Chapter 6, "Troubleshooting and Diagnostics," for help.

≡ *3*

▼ Unloading Drivers

The driver for the SunFDDI SBus Adapter can be unloaded while the system is running by using the `modunload(1M)` command. If the SunNet Manager daemons are running on your system, you must kill that process *before* you unload the SunFDDI 5.0 driver.

♦ **Perform the following steps if SunNet Manager is configured and running on your system:**

```
% ps -ef | grep nf_snmd
% kill -9 <process IDs from above command>
% modinfo | grep smt
% modunload -i <module ID for smt driver>
```

♦ **Perform the following steps to unload the SunFDDI driver:**

```
% modinfo | grep nf
% modunload -i <module ID for nf driver>
```

Installing SunNet Manager Agents for SunFDDI

Use `nf_install_agents` to install the SunNet Manager agents for SunFDDI. This script copies the FDDI schema files to the directory in which the standard agents are installed and updates the configuration files for SunNet Manager. Run `nf_install_agents` on each SunNet Manager Console, and on each SunFDDI station you want to manage using SunNet Manager.

▼ To Use `nf_install_agents` in a Solaris 2.*x* Environment

1. **Log in as** `root` **or become** `superuser`.

2. **Start** `nf_install_agents` **by typing:**

```
# /opt/SUNWconn/bin/nf_install_agents
```

3. Type the host names of the Console machine(s) that you want to receive SMT Request Frame (SRF) trap reports from the local station.
You can send SRF trap reports to more than one Console. Each Console must have the schemas for SunFDDI installed, for example:

```
Enter hostnames (if any) to send SRF Trap reports to:
(enter blank to terminate)
          enter hostname:opus
          enter hostname:rigel
          enter hostname:
```

Host name of Console ------->
Host name of Console ------->
Press Return to terminate --->

4. Specify the destination directory for the SunFDDI schemas.
SunNet Manager expects to find the schemas in the directory *<basedir>*/SUNWconn/snm/agents. The variable *<basedir>* is set to /opt by default. If you installed the SunNet Manager agents and libraries on the FDDI station, or if you used getagents to recover the agents from the Console, then this directory exists.

Enter the base directory under which you installed the SunNet Manager agents as the destination directory for the SunFDDI schemas.

```
What is the name of the root directory under which the SunNet
Manager is installed? [/opt]: [?]
```

5. Specify the current location of the SunFDDI schemas.
This is the base directory under which you installed the SunFDDI software package (SUNWnf) using pkgadd(1M).

```
What is the name of the root directory under which the nf FDDI
software is installed? [/opt]: [?]
```

The script copies the SunFDDI schemas to the destination directory and modifies the local configuration files for SunNet Manager.

6. Start the SunNet Manager agents for SunFDDI by typing:

```
# /usr/sbin/nf_snmd
```

≡ *3*

Note – If you typed n in response to the question "Do you want to start the SunNet Manager daemons for SunFDDI at boot time?" when you installed the SunFDDI software, you must start the SunNet manager agents for SunFDDI *manually* each time you reboot your machine. Alternatively, you can install the start-up script into the `/etc/rc2.d` directory.

```
# cd /opt/SUNWconn/SUNWnf/utilities
# cp nf_fddidaemon /etc/rc2.d/S98nf_fddidaemon
```

Removing SunFDDI

If you remove the SunFDDI hardware interfaces from your machine, you should also remove the SunFDDI software to prevent the device driver from starting each time you reboot the machine.

Use `pkgrm(1M)` to remove unbundled software from machines running a Solaris 2.*x* environment. Refer to *Installing Solaris Software* for detailed information on removing software packages using `pkgrm`.

▼ To Remove SunFDDI:

1. **Log in as** `root` **or become** `superuser`.

2. **Use** `pkgrm` **to remove SunFDDI.**
   The post-remove script removes the modifications made to the system files and updates the package information on your machine.

```
# /usr/sbin/pkgrm SUNWnfr
# /usr/sbin/pkgrm SUNWnft
```

3. **Modify the** `/etc/hosts` **file to remove the IP addresses and host names for the SunFDDI interfaces that were inserted by the post-installation script.**

4. **Reboot the machine to ensure correct system operation.**

Installing SunFDDI
in a Solaris 1.x Environment 4≡

This chapter assumes that you have already installed the SunFDDI SBus card in your machine and that you are loading the software from a local CD-ROM drive.

This chapter tells you how to install the SunFDDI device drivers, utilities, and SunNet Manager agents in a Solaris 1.*x* environment. It includes instructions on how to check your installation and how to remove the SunFDDI software, if necessary.

If your machine is running a Solaris 2.*x* environment, see Chapter 3, "Installing SunFDDI in a Solaris 2.x Environment" for instructions.

≡ *4*

Installation Requirements

Hardware and Software Platforms

SunFDDI was designed for SPARCstations and SPARCservers
(32 Mbytes RAM minimum) running a Solaris 1.1.1 environment or later.

Disk Space

You require 3600 KBytes of disk space to install SunFDDI in a Solaris 1.x
environment.

host Names and IP Addresses

You need to assign a unique host name and IP address to each SunFDDI device
installed in your machine. If you are installing more than one interface in a
single machine, each interface must be on a different subnet.

Declaring IP Addresses for SunFDDI

You must assign a unique IP address to each SunFDDI interface. If you have
more than one IP interface installed on your machine—either multiple
SunFDDI interfaces or some other LAN interface (for example, an Ethernet
interface)—each interface must be connected to a different subnetwork.

Before installing the SunFDDI software, update the files on your NIS/NIS+
server to assign IP addresses and host names for the SunFDDI interfaces that
you are going to configure. If you are not using a name service, update the
`/etc/hosts` file on each machine in the network to add the IP addresses and
host names of the stations attached to the FDDI ring.

The installation script updates the `/etc/hosts` file on the local machine and
enters IP addresses and host names assigned to the SunFDDI interfaces in
which it is installed.

Installing and Configuring SunFDDI

Table 4-1 lists the files and directories created on your machine when you install SunFDDI on a machine running a Solaris 1.*x* environment:

Table 4-1   Files and Directories (Solaris 1.*x*)

Files and Directories	Description
`/usr/sunlink/nf/install`	Installation script
`/usr/sunlink/nf/sys/<arch>/OBJ`	Kernel object files
`/usr/sunlink/nf/sys/sunif`	Source configuration file
`/usr/sunlink/nf/usr/etc`	FDDI utilities
`/usr/sunlink/nf/usr/snm`	Product-specific SNM schemas
`/usr/sunlink/nf/man/man8`	Product-specific man pages

The directory *<arch>* is dependent on the architecture of your machine. Use `arch(1B)` to display your kernel architecture:

```
% `arch -k`
```

▼ To Load and Mount the CD-ROM

1. **Log in as** `root` **or become** `superuser`.

2. **Place the CD-ROM in its caddy and load the caddy into the CD-ROM drive.**

3. **Create a directory on which to mount the CD-ROM.**

```
# mkdir /<mnt_dir>
```

4. **Mount the CD-ROM on the local directory.**

```
# /etc/mount -r —t hsfs /dev/sr0 /<mnt_dir>
```

▼ To Install SunFDDI Using `cdm`

Use `cdm` to install unbundled software on systems running a Solaris 1.*x* environment. The `cdm` utility is delivered on the CD-ROM with the SunFDDI software.

1. **Log in as** `root` **or become** `superuser`**.**

2. **Change to the directory that contains the SunFDDI software and start** `cdm` **by typing:**

```
# cd /<mnt_dir>/sol_1.1.x
# ./cdm
```

3. **Type** `1` **and press Return to select an application for installation.**
   When you start `cdm`, no applications are selected and the category is set to `all`.

```
Current program environment:

    Application: none
    Category: all
    Directory: /<mnt_dir>/Sol_1.1.x

----->>>>> CDM <<<<<-----

1.   Select Application
2.   Show Current Application
3.   Install Application
4.   Display Application Text File
5.   Print Application Text File
6.   List Applications
7.   List Categories
8.   Change Current Category
9.   Change Current Directory
10.  Show Program Environment

Please enter a number or q to quit: 1
```

4. Type 1 and press Return to select SunFDDI (NF) for installation.
This is the only application that can be installed from this CD-ROM.

```
Applications available:

    1: NF

Please enter a number or q to quit: 1
```

**5. Type 2 and press Return to display information about the application you
are installing; press Return to continue.**

```
"NF" selected as new application

----->>>>> CDM <<<<<-----

1.  Select Application
2.  Show Current Application
3.  Install Application
4.  Display Application Text File
5.  Print Application Text File
6.  List Applications
7.  List Categories
8.  Change Current Category
9.  Change Current Directory
10. Show Program Environment

Please enter a number or q to quit: 2

Current Application:

    Application name:NF
    Software version:5.0
    Company:        Sun Microsystems, Inc
    OS Required:    SunOS 4.1.3, SunOS 4.1.3_U1, SunOS 4.1.4
    Architectures:  sun4d, sun4m
    Text files:     none
    Install programs: install_SunLink

Press <Return> to continue:
```

6. Type 3 **and press Return to install SunFDDI.**

```
----->>>>> CDM <<<<<-----

1.   Select Application
2.   Show Current Application
3.   Install Application
4.   Display Application Text File
5.   Print Application Text File
6.   List Applications
7.   List Categories
8.   Change Current Category
9.   Change Current Directory
10.  Show Program Environment

Please enter a number or q to quit: 3
```

7. Type y **and press Return to continue the installation.**
This displays a description of the software you are about to install and a
copyright message.

If you type n and press Return, the installation process is halted without
displaying this information and without installing the software.

8. Type y **and press Return to continue the installation.**
The cdm script calls extract_unbundled, which extracts the installation
scripts from the CD-ROM. From this point on, a history of the installation is
logged in /var/tmp/unbundled/5.0_FDDI_S.log.

If you type n and press Return, the installation process is halted without
extracting the install scripts.

9. Type y **and press Return to display a description of the installation.**
This displays the name of the software you are about to install and a list of
the required and optional hardware and software.

If you type n and press Return, the installation continues without displaying
this information.

10. **Check that you have enough disk space to install the software and type y to continue the installation.**
    The installation script displays the current space available in each of the target file systems.

    If you type n and press Return, the installation is halted without installing the software. The installation scripts and log file are left in `/var/tmp/unbundled`.

11. **Type the system type and press Return to continue the installation.**
    You can install SunFDDI on a standalone machine or on a server that will share the software with multiple clients.

    ```
    Enter system type [standalone | server]:
    ```

    By default, SunFDDI is installed under `/usr`. The installation script checks for conflicting files in this directory.

12. **Type n and press Return to continue the installation in** `/usr`**.**
    If you type y and press Return, you can specify a different destination directory.

    ```
    Enter the partition you would like the software installed in:
    ```

    The script checks for an existing version of the software in the destination directory. If the software exists, you can halt the installation.

13. **Type y and press Return to continue the installation.**
    If you type n and press Return, the installation is halted without installing the software. The installation scripts extracted from the CD-ROM and the log file are left in `/var/tmp/unbundled`.

    A list of the files that are installed is displayed and a message indicates whether the installation was successful.

    If the installation is not successful, refer to Chapter 6, "Troubleshooting and Diagnostics" for help.

    ```
    install_SunLink : **** Installation Completed ****
    ```

The installation script launches the configuration script automatically.

```
Running configuration script: /usr/sunlink/nf/install/install_nf
```

A check is made of the hardware installation and the number of SunFDDI interfaces is displayed.

If the hardware is not installed, the script halts the configuration process.

14. **Install the hardware and launch** install_nf **manually.**

15. **Type the host name for which you want the interface associated.**
    Each SunFDDI interface is assigned a device name of the form nf*<inst>*, where the instance number *<inst>* is determined by the number and relative positions of the SunFDDI SBus cards installed, for example:

```
What host name do you want to use for nf<inst>: hysop-nf
```

Indicate whether you want to use this host name as the primary (official) IP address for the machine.

Each machine has a primary IP address, which is usually the IP address assigned to its Ethernet interface.

16. **If you want the primary IP address to be assigned to this SunFDDI interface, type** y **in response to the question, otherwise, type** n.

```
Do you want to use hysop-nf as the official hostname? (y|n) [y]:
```

The script compiles the configuration file and links it with the driver.

17. **Type** y **if you want the SunNet Manager agents for SunFDDI to be started automatically each time you reboot the machine.**

    a. **Type** n **if you are not using SunNet Manager to manage your network, or if you want to start the SunNet manager agents manually.**

```
Do you want to run the SunNet Manager daemon? (y|n) y
```

The driver is loaded and the IP interface is configured. If the installation is successful, your SunFDDI interfaces are now active. The Link Status Indicator (diagnostic LED) will be **amber** (the driver is installed and configured), or **green** (the interface is configured and connected to an active network).

```
FDDI has been installed and configured successfully
```

If the driver is not configured correctly or if the Link Status Indicator is **red**, see Chapter 6, "Troubleshooting and Diagnostics" for help.

18. Type q **to quit** cdm **when the installation is finished.**

```
----->>>>> CDM <<<<<-----

1.   Select Application
2.   Show Current Application
3.   Install Application
4.   Display Application Text File
5.   Print Application Text File
6.   List Applications
7.   List Categories
8.   Change Current Category
9.   Change Current Directory
10. Show Program Environment

Please enter a number or type q to quit: q
```

19. Unmount the CD-ROM and eject it from the CD-ROM drive.

```
# /etc/umount /<mnt_dir>
# eject cdrom
```

20. Remove the mount directory that you created in Step 3 on page 25.

```
# rmdir /<mnt_dir>
```

21. Reboot your machine to ensure correct system operation.

▼ To Check the Installation

1. Use `netstat(8C)` **to check for SunFDDI interfaces.**

SunFDDI interface
nf*<inst>* ---->

```
% /usr/ucb/netstat -i
Name Mtu    Net/Dest Address   Ipkts Ierrs Opkts Oerrs Collis
Queue
lo0  8232  loopback localhost  21    0     21    0     0      0
le0  1500  our-lan  hysop      2146  0     950   1     13     0
nf0  4352  our-fddi hysop-nf   1086  0     907   0     0      0
```

2. Use `ifconfig(8C)` **to check the current state of the SunFDDI interface.**

```
% /etc/ifconfig nf0
nf0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 4352
      inet <host_address> netmask <netmask> broadcast <ip_address>
```

3. Use `ping(8C)` **to check that you can send and receive data across the FDDI connection.**
If your SunFDDI station is attached to an active FDDI network, try to access a remote station through its FDDI interface.

```
% /usr/etc/ping hemlock-nf
hemlock-nf is alive
```

Your SunFDDI station should now be active, and you should be able to send and receive data across an FDDI connection.

To obtain the optimum performance from your FDDI network, you may need to tune the configurable network parameters. Refer to Chapter 5, "Improving Network Performance," for detailed instructions.

If you have problems configuring or using your SunFDDI station, see Chapter 6, "Troubleshooting and Diagnostics," for help.

Installing SunNet Manager Agents for SunFDDI

Use `nf_install_agents` to install the SunNet Manager agents for SunFDDI. This script copies the FDDI schema files to the directory in which the standard agents are installed and updates the configuration files for SunNet Manager. Run `nf_install_agents` on each SunNet Manager Console and each SunFDDI station you want to manage using SunNet Manager.

▼ To Use `nf_install_agents` in a Solaris 1.*x* Environment

1. **Log in as** `root` **or become** `superuser`.

2. **Start** `nf_install_agents` **by typing:**

```
# /usr/sunlink/nf/install/nf_install_agents
```

3. **Type the host names of Console machines that will receive SMT Request Frame (SRF) trap reports when an event occurs at the local station.**
   You can send SRF trap reports to more than one Console. Each Console must have the schemas for SunFDDI installed, for example:

```
Enter hostnames (if any) to send SRF Trap reports to:
(enter blank to terminate)
            enter hostname:opus
            enter hostname:rigel
            enter hostname:
```

host name of Console-------->
host name of Console-------->
Press Return to terminate --->

4. Specify the destination directory for the SunFDDI schemas.
SunNet Manager expects to find the schemas in the directory
<basedir>/`sunlink/nf/sys/snm/agents`. The variable *<basedir>* is set to
`/usr` by default. If you installed the SunNet Manager agents and libraries
on the client, or if you used `getagents` to recover the agents from the
Console, then this directory exists on the client machine.

Enter the base directory under which you installed the SunNet Manager
agents as the destination directory for the SunFDDI schemas.

```
What is the name of the root directory under which the SunNet
Manager is installed? [/usr]: [?]
```

5. Specify the current location of the SunFDDI schemas.
This is the base directory under which you installed the SunFDDI software
using `cdm`.

```
What is the name of the root directory under which the nf FDDI
software is installed? [/usr]: [?]
```

The SunFDDI schemas are copied to the destination directory and the local
configuration files are modified for SunNet Manager.

**6. Start the SunNet Manager agents for SunFDDI and start the SunNet
Manager agents for SunFDDI by typing:**

```
# /usr/etc/nf_snmd
```

Note – If you typed n in response to the question "Do you want to start the SunNet Manager daemons for SunFDDI at boot time?" when you installed the SunFDDI software, you must start the SunNet manager agents for SunFDDI *manually* each time you reboot your machine. Alternatively, you can add the following lines to the file /etc/rc.local:

```
if [ -f /usr/etc/nf_snmd ]; then
        /usr/etc/nf_snmd
fi
```

Removing SunFDDI

If you remove the SunFDDI hardware interfaces from your machine, you should also remove the SunFDDI software.

The Solaris 1.*x* version of SunFDDI does not include a script for removing the software. You need to remove the files, directories, and modifications to the system files manually.

▼ To Remove SunFDDI

1. **Log in as** root **or become** superuser.

2. **Remove the directory tree below** /usr/sunlink/nf.
   Be careful when you use the rm(1) command with the –r option and be sure that you type the name of the directory tree correctly.

```
# rm -r /usr/sunlink/nf
```

3. **Remove the** /dev/nf.AUTOLOAD **file and links to**
   /usr/etc/nf.INSTALL**.**

```
# cd /dev
# rm nf.AUTOLOAD
# rm nf.INSTALL nf.LOAD nf.MKDEV nf.RMDEV nf.UNLOAD nf.o
```

4. **Remove the files associated with the** nf **device driver from** /usr/etc**.**

```
# cd /usr/etc
# rm nf.o nf_macid nf_smtmon nf_stat nf.INSTALL nf_snmd*
```

5. **Modify the** /etc/loadable **script to remove references to the** nf **device
   driver.**

```
if [ -f /dev/nf.LOAD ]; then
 if [ -f /dev/nf.AUTOLOAD ]; then
 /dev/nf.LOAD
 fi
fi
```

6. **Modify the** /etc/hosts **file to remove the IP addresses and host names
   assigned to the SunFDDI interfaces.**

7. **Remove the** /etc/hostname.nf*<inst>* **files.**

```
# rm /etc/hostname.nf*
```

8. **If you assigned the primary host name to a SunFDDI interface, you must
   modify** /etc/hostname.official **to enter the host name assigned to
   the Ethernet interface** le0**.**

9. **Modify the file** `/etc/rc.local` **to remove the references to**
   `/etc/loadable` **and the SunFDDI (**`nf`**) driver.**

```
if [ `modstat | grep nf | wc -l` -eq 0 ]; then
    echo "The fddi driver has not been properly loaded."
else
    ifconfig nf<inst> <hostname> netmask + -trailers up
fi
```

10. **Remove the following lines from the** `/etc/rc.local` **file if you installed
    SunNet Manager agents for SunFDDI:**

```
if [-f /usr/etc/nf-snmd]; then
    /usr/etc/nf-snmd
fi
```

≡ *4*

Improving Network Performance 5≡

This chapter tells you how to obtain the best performance from your FDDI network. It assumes that you are familiar with FDDI network architecture and related terminology. See Chapter 7, "FDDI Network Architecture" for a detailed description of the FDDI protocols, and Chapter 8, "FDDI Network Topologies" for a detailed description of the FDDI network architecture.

⚠ **Caution** – There are a number of ways to improve network performance, some of which are discussed in this chapter; however, these should be undertaken only by experienced system administrators. Tuning your network incorrectly can affect network performance adversely.

High Performance FDDI Networks

Think of an FDDI network as having the potential to carry *more* information, rather than as being a faster connection. If the applications running over the network do not use the available bandwidth efficiently, you will not see much improvement in the performance of your network by using FDDI.

You can obtain optimum performance by balancing the complementary objectives of maximum throughput and minimum access delay:

- **Throughput** is a measure of the ring utilization. Network efficiency is defined as the ratio of actual throughput to theoretical maximum bandwidth (100 Mbps).

- **Access Delay** is the time a station has to wait before it can transmit on the network. This is largely dependent on how fast the permission token rotates around the ring.

For some applications (for example, bulk data transfers), throughput is critical; other applications (for example, voice or video applications) are more sensitive to access delay.

Ring Latency Improvement

The Ring latency is the physical delay inherent in the FDDI network. It is dependent on the number of stations in the ring and the overall length of the ring. The speed of the microprocessor in the FDDI station also has a significant effect on ring latency.

The FDDI specifications define a network that supports up to one thousand physical connections with a total ring length of 200 km (100 km per ring in a dual-ring topology). There are two physical connections for each dual-attached station; therefore, in theory, up to five hundred dual-attached stations could be connected in a single ring. In reality, optimum performance is obtained by limiting the number of connections to less than one hundred stations in an unsegmented ring.

It is possible to realize performance improvements by changing the topology of the network, trading off the total number of stations in comparison to the overall length of the ring, and selecting faster machines to act as FDDI stations. However, these modifications are rarely practical in an established network.

Data Throughput Improvement

The most significant improvements in throughput are achieved by maximizing the rate at which data is transferred to and from the FDDI network. If you are running TCP/IP or UDP/IP applications, you can improve throughput by modifying the parameters that control the rate at which these protocols transfer data to and from the SunFDDI device driver—that is, between user space and kernel space.

Send and Receive Buffers

The send and receive buffers control the maximum rate at which data is transferred between user space and kernel space by applications that use Sockets. The size of these buffers defines the amount of data that can be transferred at one time.

To optimize performance, consistent values must be assigned for the send and receive buffers. The default value (4 Kbytes) assigned to these parameters is optimized for transmission across Ethernet connections. For applications running over FDDI, the send and receive buffers should be set to 32 Kbytes.

Performance Across Bridges

The *maximum transfer unit* (MTU) for each physical layer in the network defines the maximum size (in octets) of the transmitted packets. A larger MTU means that more data can be transmitted in a single packet; a larger MTU usually means higher throughput.

For SunFDDI, the MTU is set to 4352 by default[1], and for an Ethernet interface, the MTU is 1500. This difference can cause performance problems in mixed FDDI/Ethernet networks.

Packets are transmitted between the two networks across a bridge, which handles the translation between the two protocols. When data is transmitted across the bridge from the FDDI network, the larger FDDI packets need to be

---

1. This value makes allowance for the frame header and has been demonstrated to be the optimum value of MTU for pure FDDI networks (see RFC 1390)

fragmented so that they do not exceed the MTU of the Ethernet network. Some low-cost bridges that do not support fragmentation will reject the 4352-byte packets.

Fragmenting and reassembling the packets introduces a considerable overhead that affects performance. It also increases the risk of out-of-sequence or dropped packets. Therefore, if there is heavy network traffic across the bridge, it may be more efficient to force the MTU in the FDDI network to be 1500 to match the MTU on the Ethernet side of the bridge. The above argument can also be applied to a mixed FDDI/NetWare® network. In this case, the MTU in the NetWare LAN is 4178 octets.

Target Token Rotation Time (TTRT)

⚠ **Caution** – Modifying the target token rotation time (TTRT) can affect network performance adversely. In the worst case, it can reduce network throughput to zero. In general, the TTRT should not be set to less than 8000 microseconds (ms).

The target token rotation time (TTRT) is the key FDDI parameter used for network performance tuning. In general, increasing the TTRT increases throughput and increases access delay.

For SunFDDI, the TTRT must be between 4000 and 165000 microseconds, and is set to 8000 ms by default. The optimum value for the TTRT is dependent on the application and the type of traffic on the network:

- If the network load is very irregular (*bursty* traffic), the TTRT should be set as high as possible to avoid lengthy queueing at any one station.

- If the network is used for the bulk transfer of large data files, the TTRT should be set relatively high to obtain maximum throughput, without allowing any one station to monopolize the network resources.

- If the network is used for voice, video, or real-time control applications, the TTRT should be set low to decrease access delay.

The TTRT is established during the *claim process.* Each station on the ring bids a value (`T_req`) for the operating value of the TTRT (`T_opr`) and the station with the lowest bid wins the claim. Setting the value of `T_req` on a single station does not guarantee that this bid will win the claim process.

Improving Performance in a Solaris 2.x Environment

The following section describes how to modify the user-configurable network parameters in a Solaris 2.*x* environment. To obtain optimum network performance, you may need to tune some, or all, of these parameters, depending on your network configuration and the type of network traffic.

Changing the High and Low Water Marks

The maximum rate at which data is transferred between user space and kernel space by applications that use STREAMS is controlled by the *high water marks*. These parameters define the maximum amount of data that can be queued for transmission to the TCP or UDP STREAMS modules. There are independent send and receive high water marks for TCP and UDP applications.

When the number of bytes queued exceeds the high water mark, transmission is halted temporarily to allow the backlog to be cleared. The *low water mark* specifies the level to which the queue must drop before transmission is restarted.

▼ To Tune the High Water Mark

To optimize performance, assign consistent values for the transmit and receive high water marks. The default value (8192) assigned to these parameters is optimized for transmitting across Ethernet connections. For applications running over FDDI, the high water marks should be set to 32 Kbytes.

Use ndd(1M) to change the TCP and UDP high water marks:

1. **Log in as** root **or become** superuser**.**

2. **Use** ndd −get **(the default) to check the current value of the TCP high water marks (**tcp_xmit_hiwat **and** tcp_recv_hiwat**).**

```
# ndd /dev/tcp tcp_xmit_hiwat
8192
# ndd /dev/tcp tcp_recv_hiwat
8192
```

3. **Use** ndd −set **to modify the current value of the TCP high water marks**
   (tcp_xmit_hiwat **and** tcp_recv_hiwat**).**
   For optimum performance over FDDI connections, the TCP high water
   marks should both be set to 32 Kbytes.

```
# ndd −set /dev/tcp tcp_xmit_hiwat 32768
# ndd −set /dev/tcp tcp_recv_hiwat 32768
```

4. **Use** ndd −get **(the default) to check the current value of the UDP high**
   water marks (udp_xmit_hiwat **and** udp_recv_hiwat**).**

```
# ndd −get /dev/udp udp_xmit_hiwat
8192
# ndd −get /dev/udp udp_recv_hiwat
8192
```

5. **Use** ndd −set **to modify the current value of the UDP high water marks**
   (udp_xmit_hiwat **and** udp_recv_hiwat**).**
   For optimum performance over FDDI connections, the UDP high water
   marks should both be set to 32 Kbytes.

```
# ndd −set /dev/udp udp_xmit_hiwat 32768
# ndd −set /dev/udp udp_recv_hiwat 32768
```

▼ To Tune the Low Water Mark

By default, the low water mark is set to 2 Kbytes. Provided the high water
mark is also tuned, increasing the low water mark can improve the
performance slightly. The low water mark must never be set higher than the
high water mark.

Use ndd(1M) to change the TCP and UDP low water marks:

1. **Log in as** root **or become** superuser**.**

2. **Use** `ndd -set` **to modify the current value of the TCP and UDP low water marks (**`tcp_xmit_lowat` **and** `udp_xmit_lowat`**).**
   For optimum performance over FDDI connections, set the TCP and UDP low water marks to 24 Kbytes.

```
# ndd -set /dev/tcp tcp_xmit_lowat 24576
# ndd -set /dev/udp udp_xmit_lowat 24576
```

▼ To Tune the High and Low Water Marks at Boot Time

Each time you reboot your machine, the high and low water marks are reset to their default values. If you want to configure your system so that the high water marks are modified automatically each time you reboot the machine, you must put these commands in a start-up script.

An example script is delivered on the SunFDDI CD-ROM. Once installed on your machine, this script sets the TCP and UDP high and low water marks each time you reboot. The TCP and UDP high water marks are set to 32 Kbytes; the TCP and UDP low water marks are set to 24 Kbytes. You can modify the example script to enter your own values, if required.

▼ To Install the Example Script on Your Machine

1. **Log in as** `root` **or become** `superuser`**.**

2. **Place the CD-ROM in the caddy and load the caddy into the CD-ROM drive.**

3. **Mount the CD-ROM on a local directory.**

- If the Volume Manager (`vold`) is running on your machine, then the CD-ROM is mounted automatically under `/cdrom/sun_fddi_5_0_sparc` when it is loaded in the drive.

- If the Volume Manager (vold) is not running on your machine, then you must create a directory called /cdrom/sun_fddi_5_0_sparc and mount the CD-ROM manually:

```
# mkdir /cdrom/sun_fddi_5_0_sparc
# /usr/sbin/mount -r /dev/dsk/c0t6d0s0 \
/cdrom/sun_fddi_5_0_sparc
```

4. Copy the start-up script to the directory /etc/rcS.d.

```
# cd /cdrom/sun_fddi_5_0_sparc/sol_2.5/examples
# cp ./S31ndd_fddi.sh /etc/rcd.S
```

5. Change the permissions and ownership for this file.

```
# chmod 755 s31ndd_fddi.sh
# chown root s31ndd_fddi.sh
# chgrp sys s31ndd_fddi.sh
```

▼ To Tune the Socket Options

For applications that use Sockets, the application developer can make more efficient use of the increased bandwidth provided by an FDDI connection by increasing the size of the send and receive buffers used by a specific application. This manipulation is restricted to applications and do not affect the other Socket connections.

Both Solaris 1.*x* and 2.*x* environments support program calls to getsockopt(3N) and setsockopt(3N). The options SO_SNDBUF and SO_RCVBUF are used to set the size of the TCP send and receive buffers for a specified Socket. See the man page for more detailed information.

To optimize performance, assign consistent values for the transmit and receive buffers. The default value (4 Kbytes) assigned to these parameters is optimized for transmitting across Ethernet connections. For applications running over FDDI, the send and receive buffers should be set to 32 Kbytes.

Code examples that demonstrate how `sockopt` can be used to optimize a TCP/IP application for an FDDI network are provided on the SunFDDI CD-ROM. To access the code examples:

♦ **Insert the CD-ROM in a CD-ROM drive and mount the CD-ROM on a local directory.**
The Solaris 2.*x* examples are located in *<mnt_dir>*/`sol_2.5/examples`. Instructions for running the example programs are contained in the `README` file in this directory.

MTU Path Discovery

Machines running a Solaris 2.*x* environment support MTU path discovery, which allows the optimum MTU to be negotiated. Under most circumstances, this ensures efficient use of the network resources. However, to enable MTU path discovery to work, the "`don't fragment`" bit in the packet is set, which causes a problem with some bridges and routers that do not support this feature. In this case, packets are not transmitted across the bridge, and the error messages generated as a result can be misleading.

Refer to "To Disable MTU Path Discovery" for detailed instructions on how to accommodate bridges and routers that do not support this feature.

▼ To Tune the Maximum Transfer Unit (MTU)

For SunFDDI, the maximum transfer unit (MTU) has been optimized for pure FDDI networks. Follow these steps to change the value of the MTU on a station running SunFDDI in a Solaris 2.*x* environment:

1. **Log in as** `root` **or become** `superuser`**.**

2. **Edit the** `/etc/system` **file to add the following line:**

```
set nf:nf_mtu=<packet_size>
```

The variable *<packet_size>* is set to the desired value for MTU entered in octets. The default value for MTU is 4352 octets. It must be in the range 1 to 4500 octets.

3. **Reboot the machine to apply the changes to the system.**

Note – Additionally, the `ifconfig(1M)` command can be used to change the MTU.

▼ To Disable MTU Path Discovery

Many bridges do not support MTU path discovery.

♦ **To disable this feature in order to transmit packets across a bridge, type:**

```
# ndd −set /dev/ip ip_path_mtu_discovery 0
```

MTU path discovery is re-enabled each time the machine is rebooted.

▼ To Tune the Target Token Rotation Time (TTRT)

The target token rotation time (TTRT) for the network is the lowest value of `T_req` bid during the claim process. Follow these steps to change the value of `T_req` bid by a station running a Solaris 2.*x* environment:

1. **Log in as** `root` **or become** `superuser`.

2. **Edit the** `/etc/system` **file to add the following line:**

```
set nf:nf_treq=<bid>
```

3. **Reboot the machine to apply the changes to the system.**

Improving Performance in a Solaris 1.x Environment

The following section describes how to modify the user-configurable network parameters in a Solaris 1.*x* environment. To obtain optimum network performance, you may need to tune some, or all, of these parameters, depending on your network configuration and the type of network traffic.

▼ To Tune the Send and Receive Buffers

The maximum rate at which data is transferred between user space and kernel space by applications that use Sockets is controlled by the send and receive buffers. The size of these buffers defines the amount of data that can be transferred at one time.

To optimize performance, consistent values must be assigned for the send and receive buffers. The default value (4 Kbytes) assigned to these parameters is optimized for transmitting across Ethernet connections. For applications running over FDDI, the send and receive buffers should be set to 32 Kbytes.

Follow these steps to change the default size of the TCP send and receive buffers used by all TCP/IP applications running on your machine:

1. **Log in as** `root` **or become** `superuser`.

2. **Edit the** `/sys/netinet/in_proto.c` **file to modify the following lines:**

Set to 32 Kbytes for FDDI ---->
Set to 32 Kbytes for FDDI ---->

```
/*
* Default TCP buffer sizes (in bytes)
*/
int    tcp_sendspace = 1024*4*8
int    tcp_recvspace = 1024*4*8
```

3. **Make a new kernel that is optimized for applications running over FDDI.**
   For example, to make a new kernel that is based on generic `vmunix`:

```
#  cd /sys/`arch -k`/conf
#  cp GENERIC FDDI
#  config FDDI
#  cd ../FDDI
#  make
#  cp /vmunix /vmunix.ether
#  cp vmunix /vmunix
```

4. **Reboot your machine to invoke the changes to the kernel.**

▼ To Patch the Buffer Size

If you want to experiment with various buffer sizes, you can patch the kernel temporarily using the general-purpose debugging tool adb(1). These changes will be lost when you reboot the machine.

Follow these steps to patch the kernel using adb:

1. **Log in as** root **or become** superuser.

2. **Pass the desired buffer size as input to the debugger.**
   Enter the value in hexadecimal. For example, to increase the TCP buffer size to 32 Kbytes:

```
#  echo "tcp_sendspace/W 0x8000 | adb -w /vmunix /dev/kmem
#  echo "tcp_recvspace/W 0x8000 | adb -w /vmunix /dev/kmem
```

▼ Tune the Socket Options

For applications that use Sockets, the application developer can make more efficient use of the increased bandwidth provided by an FDDI connection by increasing the size of the send and receive buffers used by a specific application. This manipulation is restricted to the application and does not affect the other Socket connections.

Both Solaris 2.*x* and Solaris 1.*x* environments support program calls to getsockopt(2) and setsockopt(2). The SO_SNDBUF and SO_RCVBUF options are used to set the size of the TCP send and receive buffers for a specified Socket. (See the man page for more information.)

To optimize performance, consistent values must be assigned for the transmit and receive buffers. The default value (4 Kbytes) assigned to these parameters is optimized for transmitting across Ethernet connections. For applications running over FDDI, the send and receive buffers should be set to 32 Kbytes.

Code examples that demonstrate how sockopt can be used to optimize a TCP/IP application for an FDDI network is provided on the SunFDDI CD-ROM. To access the code examples:

♦ **Insert the CD-ROM in a CD-ROM drive and mount the CD-ROM on a local directory.**
The Solaris 1.x examples are located in *<mnt_dir>*/sol_1.1.x/examples. Instructions for running the example programs are located in the README file.

▼ To Tune the Maximum Transfer Unit (MTU)

For SunFDDI, the maximum transfer unit (MTU) has been optimized for pure FDDI networks. Follow these steps to change the value of the MTU on a station running SunFDDI in a Solaris 1.*x* environment:

1. **Log in as** root **or become** superuser.

2. **Edit the file** /usr/sunlink/nf/sys/sunif/nf_conf.c **to modify the following line:**

```
int     nf_mtu = <packet_size>
```

The variable *<packet_size>* is set to the desired value for MTU entered in octets. The default value for MTU is 4352 octets. It must be in the range of 1 to 4500 octets.

3. **Change the directory to** /usr/sunlink/nf/sys/*<arch>*/OBJ**, where** *<arch>* **is dependent on your machine architecture.**

```
#   cd /usr/sunlink/nf/sys/<arch>/OBJ
```

4. **Compile the file** /usr/sunlink/nf/sys/sunif/nf_conf.c **using the following compile options:**

```
# cc -O -I/sys -DKERNEL -sparc -c ../../sunif/nf_conf.c
```

5. **Link the object file** nf_conf.o **with the other SunFDDI kernel object files:**

```
# ld -r -o nf.o if_nf.o smt.o nf_smtcore.o nf_conf.o
```

     b. If the system has a multicast kernel, link the object file `nf_conf.o` **as follows:**

```
# ld -r -o nf.o if_nf.mcast.o smt.mcast.o nf_smtcore.o nf_conf.o
```

6. Move the SunFDDI driver `nf.o` **to** `/usr/etc.`

```
#  mv nf.o /usr/etc
```

7. Reboot the machine to apply the changes to the system.

▼ To Tune the Target Token Rotation Time (TTRT)

The target token rotation time (TTRT) for the network is the lowest value of `T_req` bid during the claim process.

Follow these steps to change the value of `T_req` bid by a station running SunFDDI (with loadable drivers) in a Solaris 1.*x* environment:

1. Log in as `root` **or become** `superuser.`

2. Edit the `/usr/sunlink/nf/sys/sunif/nf_conf.c` **file to modify the following line:**

```
int     nf_treq = <bid>
```

The variable *<bid>* is set to the desired value for `T_req` entered in microseconds. The default value for `T_req` is 8000 microseconds. It must be in the range 4000 to 165000 microseconds.

3. Change directory to `/usr/sunlink/nf/sys/<arch>/OBJ`**, where** *<arch>* **is** `sun4c` **or** `sun4m`**, depending on your machine architecture.**

```
#  cd /usr/sunlink/nf/sys/<arch>/OBJ
```

4. **Compile the** `/usr/sunlink/nf/sys/sunif/nf_conf.c` **file using the following compile options:**

```
# cc -O -I/sys -DKERNEL -sparc -c ../../sunif/nf_conf.c
```

5. **Link the object file** `nf_conf.o` **with the other SunFDDI kernel object files:**

```
# ld -r -o nf.o if_nf.o smt.o nf_smtcore.o nf_conf.o
```

If the system has a multicast kernel, link the object file `nf_conf.o` as follows:

```
# ld -r -o nf.o if_nf.mcast.o smt.mcast.o nf_smtcore.mcast.o nf_conf.o
```

6. **Move the SunFDDI driver** `nf.o` **to** `/usr/etc`**.**

```
#  mv nf.o /usr/etc
```

7. **Reboot the machine to apply the changes to the system.**

☰ *5*

_____

Troubleshooting and Diagnostics 6 ≡

This chapter describes how to detect and resolve common problems with your FDDI network. It includes instructions on how to load and configure the SunFDDI device driver manually, if required. Note that SunFDDI supports almost all of the standard SunOS network diagnostic utilities, including `snoop(1M)`, `netstat(1M)`, `ping(1M)`, and `etherfind(8C)`.

For a Solaris 2.*x* environment, the default base directory *<basedir>* is:

```
/opt/SUNWconn/bin
```

For a Solaris 1.*x* environment, the default base directory *<basedir>* is:

```
/usr/sunlink/nf/usr/etc
```

Troubleshooting Checklist

Use the following checklist to verify the major components of your SunFDDI station, and to check that it is installed, configured, and attached to the network correctly.

▼ To Check the Connection to the Station

♦ **Check that the cable connector is seated firmly into the plug on the SunFDDI SBus card. You should feel the connector "click" into place.**

▼ To Check the Connections Between Stations

♦ **Check that the cables are connected correctly between ports of different types.**
The keys provided with most FDDI cables will help you to avoid "crossing" or "twisting" the ring. See "Connecting Fiber Optic Cables" on page 6 for a detailed description.

- For single attached stations (SAS), cables should be connected by either S↔M (to a concentrator) or S↔S (back-to-back).

- For dual attached stations (DAS), cables should be connected by either A↔B B↔A (dual-ring) or A↔M M↔B (dual-homing).

▼ To Check the Link Status Indicator

♦ **Check the status of the SunFDDI interface by looking at the Link Status Indicator (diagnostic LED) mounted on the backplate of the SunFDDI SBus card.**

- If the LED is **green**, the SunFDDI driver is loaded, the nf interface is configured, and the station is attached to an active FDDI network. This is the operating state.

- If the LED is **amber**, the SunFDDI driver is loaded, the `nf` interface is configured, but no valid line state to an active FDDI network exists.

  a. Check the physical connections between the station and the ring, or the station and the concentrator.

  b. Check the status of the neighboring station.
  If this station is down, it will bring down the link between the stations.

- If the LED is **red**, either the SunFDDI driver is loaded and the interface is *not* configured, or the driver has not been loaded.

  a. Configure the `nf` **interface with the** `ifconfig(1M)` **command of the form:**

```
# ifconfig plumb nf<inst> <hostname> netmask <mask> up
```

- If the LED is **red**, the SunFDDI driver is not loaded, or is loaded incorrectly.

  a. Check that the software is installed.

  b. Remove the existing version, if necessary, and reload the software.
  See Chapter 3, "Installing SunFDDI in a Solaris 2.x Environment," or Chapter 4, "Installing SunFDDI in a Solaris 1.x Environment," for instructions.

▼ To Check the MAC Address

By default, the first SunFDDI interface installed in the machine adopts the host-resident MAC address stored on the motherboard. Each subsequent SunFDDI interface takes the card-resident MAC address stored in its IDPROM.

A potential conflict exists between the first SunFDDI 5.0 interface and a SunFDDI 2.0 interface because both revisions of the product take the host-resident MAC address. You must change the MAC address assigned to one of these interfaces. See "Changing the Default MAC Address (`nf_macid`)" on page 112 for detailed instructions.

▼ To Check the SunFDDI Driver

♦ **Use** `netstat(1M)` **to check that the SunFDDI (`nf`) driver is installed correctly, and to check for an excessive number of errors and collisions:**

SunFDDI interface
nf*<inst>* ---->

```
% netstat -i
Name Mtu    Net/Dest Address   Ipkts Ierrs Opkts Oerrs Collis
Queue
lo0  8232  loopback localhost  21    0     21    0     0      0
le0  1500  our-lan  hysop      2146  0     950   1     13     0
nf0  4352  our-fddi hysop-nf   1086  0     907   0     0      0
```

A sudden increase in the number of errors could indicate a noisy connection caused by a dirty cable.

▼ To Check the IP Addresses

You must assign a unique IP address to each SunFDDI interface. If you have more than one IP interface installed in your machine—either multiple SunFDDI interfaces, or some other LAN interface (for example, an Ethernet interface)—each interface must be attached to a different subnetwork.

♦ **Use** `ifconfig(1M)` **to check the IP addresses assigned to the two SunFDDI interfaces:**

```
# ifconfig -a
nf0: flags=842(UP,BROADCAST,RUNNING,MULTICAST) mtu 4352
     inet 123:123:201:71 netmask ffffff00 broadcast
123:123:201:255
nf1: flags=843(UP,BROADCAST,RUNNING,MULTICAST) mtu 4352
     inet 123:123:201:75 netmask ffffff00 broadcast
123:123:201:255
```

If the interfaces have the same subnetwork number, as shown in this example, you must change the IP address assigned to one of the interfaces.

Ensure that the IP addresses and host names for each SunFDDI interface are entered in the NIS map, or in `/etc/hosts` on the local machine if you are not using NIS/NIS+.

▼ To Check the IP Routing

1. **Check the IP routing table using the** netstat(1M) **command:**

```
% netstat —nr
```

2. **Check that the FDDI subnetwork is featured in the routing table.**
   If the routing table is empty, check that the routing daemon (in.routed) is
   running on your machine.

▼ To Check the Protocol Statistics

♦ **Use** netstat(1M) **to check the per-protocol (IP, TCP, UDP, etc.) statistics
   for errors:**

```
# netstat -s
UDP
    udpInDatagrams      =  1423udpInErrors         =     0
    udpOutDatagrams     =   540

TCP tcpRtoAlgorithm     =      4tcpRtoMin          =    200
    tcpRtoMax           = 60000tcpMaxConn          =     -1
    tcpActiveOpens      =     38tcpPassiveOpens     =      1
    tcpAttemptFails     =      0tcpEstabResets      =     33
    tcpCurrEstab        =      2tcpOutSegs          =    427
    tcpOutDataSegs      =    264tcpOutDataBytes     = 15917
    tcpRetransSegs      =     38tcpRetransBytes     =      0
    tcpOutAck           =    143tcpOutAckDelayed    =     56
    tcpOutUrg           =      1tcpOutWinUpdate     =      0
    tcpOutWinProbe      =      0tcpOutControl       =    111
    tcpOutRsts          =     36tcpOutFastRetrans   =      0
    tcpInSegs           =    606
    tcpInAckSegs        =    292tcpInAckBytes       = 15949
    tcpInDupAck         =     62tcpInAckUnsent      =      0
    tcpInInorderSegs    =    311tcpInInorderBytes   = 99169
    tcpInUnorderSegs    =      0tcpInUnorderBytes   =      0
    .
    .
    .
```

♦ **Use** `netstat(1M)` **to check the driver statistics:**

```
# netstat -k
nfo
    ipacket = x, ierror = y, opackets = z
    .
    .
    .
```

▼ To Check the SMT Traffic

If the target station is not running SunFDDI, it does not necessarily support the same set of SMT frames. If the target station receives an SMT request for an unsupported service, it issues a Request Denied Frame (RDF).

♦ **Use** `nf_smtmon(1M)` **to examine the SMT frames:**

```
# <basedir>/nf_smtmon [-i nf<inst>] [−x] [−h] [<frameclass>]
```

Solving Common Problems

Use the following information to diagnose and resolve some of the common problems that can occur when installing, configuring, or running SunFDDI.

Table 6-1   Problems Installing SunFDDI in a Solaris 2.x Environment

Problem	Action
The SunFDDI software package (`SUNWnfr`) cannot be found. The SunFDDI man pages/utilities package (`SUNWnft`) cannot be found.	Check that you have inserted the CD-ROM in the CD-ROM drive and that the CD-ROM is mounted on a local directory. If the Volume Manager (`vold`) is running on your machine, the SunFDDI software is located in the following directory: `/cdrom/sun_fddi_5_0_sparc`. If the Volume Manager (vold) is not running on your machine, you must create a directory and mount the CD-ROM as described on page 13.
The configuration script cannot configure the IP address. "What ip address do you want to use for xyz [] [?] 0.0.0.0 Illegal ip address. . . . Try again"	Check that you have entered the valid IP address. The script will keep prompting you until you enter a valid address.
The configuration script cannot find the hardware devices installed in the machine.	Check that the SunFDDI SBus card is installed correctly and is seated firmly in the SBus slot. Use the `prtconf` command to see if the `nf` card exists.
The configuration script cannot load the SunFDDI driver. The procedure fails with the message, "`no available major numbers.`"	The number of major numbers—that is, the maximum number of device drivers that can be installed—is limited to 127 in a Solaris 2.*x* environment. This number is quickly exhausted by the large number of drivers installed in a typical system. Remove an existing driver before loading SunFDDI. This limitation does not exist in other versions of the operating system.
The configuration script cannot load the device driver.	Check whether the driver is already installed on the system. If you removed a previous version of the package using `pkgrm(1M)`, you must reboot the system before attempting another `add_drv`.
The configuration script cannot configure the device driver. The procedure fails with the message, "`Duplicate MAC address.`"	Two or more interfaces have been assigned the same MAC address. The most likely conflict lies between the first SunFDDI 5.0 interface (`nf0`) and one or more SunFDDI 2.0 (`bf`) interfaces installed in the same machine.
The `netstat -k` command shows non-zero "norxsmtbufs"	Add the following line to the `/etc/system` file and reboot the system: **`set nf:nf_num_smt_rx = 15`**

Table 6-2   Problems Installing SunFDDI in a Solaris 1.x Environment

Problem	Action
The device driver cannot be loaded.	Ensure that the following options are enabled in your kernel configuration file /sys/<*arch*>/conf/<*kernel*>: `options  VDDRV   #loadable modules`

The device driver cannot be loaded.	Ensure that the following options and pseudo devices are enabled in your kernel configuration file /sys/<*arch*>/conf/<*kernel*>:
	
	`options  IPCMESSAGE  #System V IPC message`
	`facility`
	`options  IPCSEMAPHORE #System V IPC semaphore`
	`facility`
	`options  IPCSHMEM    #System V IPC shared-memory`
	`facility`
	
	`pseudo-device  snit    #streams NIT`
	`pseudo-device  pf      #packet filter`
	`pseudo-device  nbuf    #streams NIT buffering`
	`module`
	`pseudo-device  clone   #clone device`

The configuration script cannot configure the device driver. The procedure fails with the message "`Duplicate MAC address`".	Two or more interfaces have been assigned the same MAC address. The most likely conflict lies between the first SunFDDI 4.0 interface (`nf0`) and one or more SunFDDI 2.0 (`bf`) interfaces installed in the same machine

Table 6-3   Problems Running SunFDDI

Problem	Action
The link status indicator (diagnostic LED) remains red.	Check that the SunFDDI software is installed correctly and that the driver is loaded and configured.
The link status indicator (diagnostic LED) remains amber.	Check that the SunFDDI SBus card is connected to the network.
The link status indicator (diagnostic LED) remains amber, even after the local station is connected to the network and the driver is configured and loaded.	Check that the other end of the cable is connected and that the neighboring station or concentrator is configured correctly. Some concentrators have diagnostic LEDs that indicate if the ring is operating correctly. Check that the ring is not crossed.

Table 6-3   Problems Running SunFDDI

Problem	Action
The SunFDDI interface (`nf`) is not displayed by `netstat`.	Check that the SunFDDI device driver is loaded, using `modinfo(1M)` to display information about loaded kernel modules. If the device driver is not loaded, see "Loading the Device Driver Manually (Solaris 2.x)" on page 68 for detailed instructions.
The SunFDDI interface (`nf`) is not displayed by `ifconfig`.	Check that the SunFDDI device driver is loaded, using `modinfo(1M)` to display information about loaded kernel modules. If the device driver is not loaded, see "Loading the Device Driver Manually (Solaris 2.x)" on page 68 for detailed instructions.
Frequent errors occur when running NFS over FDDI in a Solaris 2.3 environment.	You must install the NFS "jumbo" patch (`101344-05`) on machines running Solaris 2.3 to run NFS over FDDI. These errors have been fixed in subsequent releases of the operating system.
The local station can reach FDDI stations located on the same subnetwork, but these stations cannot reach the local station.	Check that the IP address and host name of the local station is entered in the NIS map or NIS+ tables (or in `/etc/hosts` on each remote station if you are not running NIS or NIS+).
The local station cannot reach other FDDI stations located on the same subnetwork.	Check that the link status indicator is **green**, indicating that the SunFDDI interface is attached to an active network. Use `ifconfig(1M)` to check that the SunFDDI interface (`nf`) is up. Check that the IP address and host name of the remote station is entered in the NIS map or NIS+ tables (or in `/etc/hosts` on the local station if you are not running NIS or NIS+).
Running `nf_stat` without the −m option (to display information about the local station) shows the ring alternating frequently between `UP` and `DOWN`.	Check that the link status indicator is **green**, indicating that the SunFDDI interface is attached to an active network. Check the connections to the concentrator. Some concentrators have diagnostic LEDs that indicate if the ring is operating correctly.

Table 6-3   Problems Running SunFDDI

Problem	Action
Running `nf_stat` without the –m option (to display information about the local station) shows that the ring is `ISOLATED`.	Check that the link status indicator is **green**, indicating that the SunFDDI interface is attached to an active network. Check the connections to neighboring stations or the concentrator. Some concentrators have diagnostic LEDs that indicate if the ring is operating correctly. Check that the neighboring stations are configured correctly. If everything else seems to be working, check the cable itself. Fiber optic cable is fragile and should not be subjected to stress or impact. If the turning curve around objects is too tight, the light path is interrupted. Clean the connectors.
The local station cannot reach FDDI stations located on a remote FDDI network.	Check that the IP address and host name of the remote station is entered in the NIS map or NIS+ tables (or in `/etc/hosts` on each remote station if you are not running NIS or NIS+). Check that the routing tables are using `netstat` –r. Check for an `nf` interface that shows a route to a gateway with the `UG` flags set. If you can reach the local gateway, then the problem probably lies with the IP routing. Try to confirm that the remote station is configured and running, and if possible try to reach the local station from the remote station. If you can, it indicates that the problem lies with the local IP routing. Try to reach the router nearest the remote station. If you can, it indicates that the fault lies between the remote station and the router. If you cannot, try to reach a router that is closer to the local station. Continue in this way until you have isolated the router that is dropping or misrouting packets. If the dynamic routing protocol is not adding routes, try adding a static route to the remote station. This method is not recommended for large networks with a large number of nodes.

Table 6-3   Problems Running SunFDDI

Problem	Action
The local station cannot reach FDDI stations located on a remote Ethernet network.	Check that the IP address and host name of the remote station is entered in the NIS map or NIS+ tables (or in `/etc/hosts` on each remote station if you are not running NIS or NIS+). If you are operating in a Solaris 2.*x* environment, which uses MTU path discovery, check that packets are being transmitted across the bridge between the networks. Many bridges do not yet support MTU path discovery and do not fragment the large packets sent by the FDDI stations. In this case you may need to disable this feature on your station by typing: # **ndd —set /dev/ip ip_path_mtu_discovery 0** Check that the routing tables are using `netstat —r`. Check for an `nf` interface that shows a route to a gateway with the `UG` flags set. If you can reach the local gateway, then the problem probably lies with the IP routing. Try to confirm that the remote station is configured and running, and if possible try to reach the local station from the remote station. If you can, it indicates that the problem lies with the local IP routing. Try to reach the router nearest the remote station. If you can, it indicates that the fault lies between the remote station and the router. If you cannot, try to reach a router that is closer to the local station. Continue in this way until you have isolated the router that is dropping or misrouting packets. If the dynamic routing protocol is not adding routes, try adding a static route to the remote station. This method is not recommended for large networks with a large number of nodes.
Running `nf_stat` with the —m option (to display information about the neighboring stations) shows frequent `error` frames and `lost` frames.	This could indicate a noisy network. For fiber connections, noise is probably caused by dirt or grease on the optical surface or by a damaged cable. Unshielded twisted-pair is sensitive to electrical and electromagnetic interference. Note that the receipt of `error` frames and `lost` frames does not give any indication of where the problem is occurring on the ring.

Table 6-3   Problems Running SunFDDI

Problem	Action
Running `nf_smtmon` shows the frequent occurrence of request denied frames (RDF).	The SMT entity on one of the remote stations does not support the same set of SMT services. This should not occur on a station running SunFDDI. You can use this facility to detect problems communicating with a remote station using the SunNet Manager proxy agent.

Table 6-4   Problems Using the SNM Agents for SunFDDI

Problem	Action
Cannot select `nf_fddi` or `nf_fddismt7_2` agent for quick dump.	Check that the SNM agents for SunFDDI are enabled in the properties window for the proxy station. See "To Use the SunFDDI Proxy Agent (`nf_fddismt7_2`)" on page 142 for detailed instructions.
Cannot recover SMT statistics from the proxy agent. The procedure fails with the error message, "`Cannot create RPC client: program is not registered.`"	Check that you specified the host name or MAC address for the target machine in the properties window for the target station. See "To Use the SunFDDI Proxy Agent (`nf_fddismt7_2`)" on page 142 for detailed instructions. Check that the SNM daemons are started on the proxy and the target machines: `# `**`ps -ef	grep snm`** `root `*`<pid>`*` `*`<date&time>`*` `*`<path>`*`/nf_snmd` `root `*`<pid>`*` `*`<date&time>`*` `*`<path>`*`/nf_snmd` `root `*`<pid>`*` `*`<date&time>`*` `*`<path>`*`/nf_snmd`  On a Solaris 2.*x* client, you can start the SunNet Manager daemons for SunFDDI by typing:  `# `**`/usr/sbin/nf_snmd`**  On a Solaris 1.*x* client, you can start the SunNet Manager daemons for SunFDDI by typing:  `# `**`/usr/etc/nf_snmd`**

Running the Hardware Self-Test Program

If you suspect that there may be a problem with the SunFDDI SBus card, you can use the built-in hardware self-test to check the state of its primary components and the connection to the network.

▼ To Run the Hardware Self-Test:

1. Log in as root **or become** superuser.

2. Halt the machine.

```
# sync;sync;halt
```

3. At the boot prompt, type:

```
ok test /sbus/nf

    Local MAC Address 0:80:d8:10:3:ed (Canonical)
    Interrupt register read/write test .... PASS
    CMT Processor read/write test ......... PASS
    S Port register test .................. PASS
    MAC register test ..................... PASS
    DMA register test ..................... PASS
    DMA bus master test ................... PASS
    Performing FDDI path test ............. PASS
    Connection Management (CMT) test ...... PASS
    Local loopback packet test ........... PASS
    Network loopback packet test .......... connection not active

/sbus/sbf selftest failed. Return code = -1
```

This example shows that the primary components on the board are working, but that the board is not currently connected to the network.

a. For Sun4m systems, at the boot prompt, type:

```
ok test /iommu/sbus/nf
```

Loading the Device Driver Manually (Solaris 2.x)

Normally, the SunFDDI device driver is loaded and configured by the post-installation script, which is launched automatically when the software is installed. If you encounter problems when running this script, or if you want to customize the installation, you may need to load the device driver manually.

▼ To Configure and Load the Device Driver

1. Log in as `root` **or become** `superuser`**.**

2. Check that there are no partially installed drivers for SunFDDI.

a. Check for existing SunFDDI devices in the `/dev` directory.

```
# ls /dev | grep nf
nf
# ls /dev | grep smt
smt
```

b. Use `rem_drv(1M)` to remove any installed SunFDDI devices and remove the device directories.

```
# /usr/sbin/rem_drv smt
# /usr/sbin/rem_drv nf
# /bin/rm −f /dev/smt
# /bin/rm −f /dev/nf
```

c. Check the `/etc/name_to_major` file for entries of the form `nf`*<num>* and `smt`*<num>* and remove these if they exist.

3. Use `add_drv(1M)` **to inform the system about the new drivers.**

```
# /usr/sbin/add_drv nf
# /usr/sbin/add_drv smt
```

4. Check that the device drivers have been added to the system correctly.

a. Look in the /devices/pseudo directory for entries of the form:

```
ls -l /devices/pseudo | grep nf
crw------- 1 root sys 11,119 <date> clone@0:nf
ls -l /devices/pseudo | grep smt
crw------- 1 root sys 11,119 <date> clone@0:smt
```

b. Look in the /dev directory for links to these entries:

```
ls -l /dev | grep nf
lrwxrwxrwx 1 root other <date> nf -> ../devices/pseudo/clone@0:nf
ls -l /dev | grep nf
lrwxrwxrwx 1 root other <date> smt ->
../devices/pseudo/clone@0:smt
```

5. **Create a file called** /etc/hostname.nf<*inst*> **for each SunFDDI IP interface that you configure.**
Each file must contain the host name assigned to the IP interface. These files are used to configure the interfaces when the system is rebooted. If you assign the primary host name to one of the FDDI interfaces, this host name must be entered in the file /etc/nodename.

6. **Use** ifconfig(1M) **to configure the SunFDDI IP interfaces (**nf<*inst*>**).**
The host name assigned to the IP interface must be entered in the NIS map, NIS+ tables, or in /etc/hosts on the local machine. Configure the interface using an ifconfig command of the form:

```
# /usr/sbin/ifconfig nf<inst> plumb <hostname> netmask + up
```

Note that the modifier netmask + takes the netmask defined in the /etc/netmasks file. If this entry is missing, you must specify the netmask explicitly using dot notation.

Provided the installation and configuration was successful, your SunFDDI station is now active and you can send and receive data transparently across an FDDI connection.

≡ *6*

Loading the Device Driver Manually (Solaris 1.x)

Normally, the SunFDDI device driver is loaded and configured by the post-installation script which is launched automatically when the software is installed. If you encounter problems when running this script or if you want to customize the installation, you may need to load the device driver manually.

▼ To Configure and Load the Device Driver

1. **Log in as** `root` **or become** `superuser`**.**

2. **Change directory to** `/usr/sunlink/nf/sys/<arch>/OBJ`**, where** *<arch>* **is dependent on your machine architecture.**

```
#  cd /usr/sunlink/nf/sys/<arch>/OBJ
```

3. **Compile the** `/usr/sunlink/nf/sys/sunif/nf_conf.c` **file using the following compile options:**

```
# cc -O -I/sys -DKERNEL -sparc -c ../../sunif/nf_conf.c
```

4. **Link the** `nf_conf.o` **object file with the other SunFDDI kernel object files:**

```
# ld -r -o nf.o if_nf.o smt.o nf_smtcore.o nf_conf.o
```

If the system has a multicast kernel, link the object file `nf_conf.o` as follows:

```
# ld -r -o nf.o if_nf.mcast.o smt.mcast.o nf_smtcore.mcast.o nf_conf.o
```

5. **Move the SunFDDI driver** `nf.o` **to** `/usr/etc`**.**

```
#  mv nf.o /usr/etc
```

6. Create the `/dev/nf.AUTOLOAD` **file.**

```
#   touch /dev/nf.AUTOLOAD
```

7. Create the following soft links that must point to the script
`/usr/etc/nf.INSTALL` **script.**

```
#   ln −s /usr/etc/nf.INSTALL /dev/nf.LOAD
#   ln −s /usr/etc/nf.INSTALL /dev/nf.UNLOAD
#   ln −s /usr/etc/nf.INSTALL /dev/nf.INSTALL
#   ln −s /usr/etc/nf.INSTALL /dev/nf.MKDEV
#   ln −s /usr/etc/nf.INSTALL /dev/nf.RMDEV
```

8. Create the `/etc/loadable` **script, which is used to load the driver
module.**

```
if [ −f /dev/nf.LOAD ]; then
     if [ −f /dev/nf.AUTOLOAD ]; then
          /dev/nf.LOAD
     fi
fi
```

9. Execute the `/etc/loadable` **script to load the driver module.**

10. Use `ifconfig(1)` **to configure the SunFDDI IP interfaces (**`nf`<*inst*>**).**
The host name assigned to the IP interface must be entered in the NIS map,
or in `/etc/hosts` on the local machine. Configure the interface using an
`ifconfig` command of the form:

```
#   /usr/etc/ifconfig nf<inst> <hostname> netmask + up
```

Note that the modifier `netmask +` takes the netmask defined in the file
`/etc/netmasks`. If this entry is missing, you must specify the netmask
explicitly using dot notation.

Provided the installation and configuration was successful, your SunFDDI
station is now active and you can send and receive data transparently across an
FDDI connection.

■ 6

Part 2 — *Planning and Implementing SunFDDI Networks*

FDDI Network Architecture 7≣

This chapter provides a brief introduction to the Fiber Distributed Data Interface (FDDI), and the network architecture described by the following specifications:

- ANSI/FDDI Physical Media Dependent (PMD) X3.166-1990
- ANSI/FDDI Physical Layer (PHY) X3.148-1988
- ANSI/FDDI Medium Access Control (MAC) X.3.139-1987
- ANSI/FDDI Station Management (SMT) X3.299 R7.3 (formally R7.2.99)

FDDI Network Model

The Fiber Distributed Data Interface (FDDI) provides high-performance, multistation networking at data transfer rates of up to 100 Mbps. It is based on a Token-Ring network architecture, and provides communication over optical fiber or copper twisted-pair connections.

The ANSI/FDDI specifications define a network model that consists of the following components:

- Physical Media Dependent (PMD) Layer
- Physical (PHY) Layer
- Medium Access Control (MAC) Layer
- Station Management (SMT) Protocol

Figure 7-1 shows the organization of these components, and their relationship with other network protocols that use them.
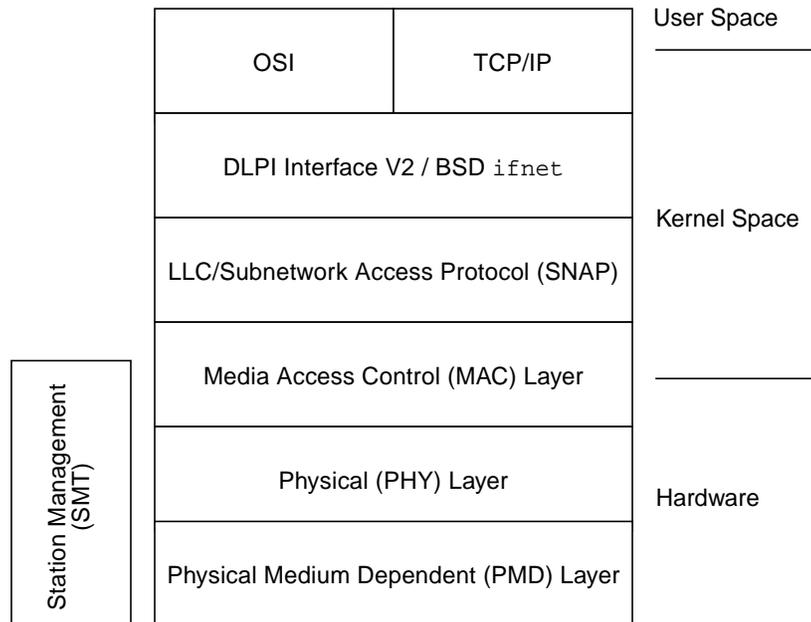
Figure 7-1   FDDI Architectural Model

*Physical Medium Dependent (*PMD*) Layer*

This portion of the FDDI network model defines the physical medium used to carry the encoded digital signal. It is also referred to as the *media* layer.

The PMD layer determines the characteristics of the transmitters, receivers, connectors, and cables used to attach the FDDI station to the network. SunFDDI supports the following PMD options:

- **Multimode fiber (MMF)**—single and dual-attached
- **Unshielded twisted-pair (UTP)**—single-attached *only*

Multi-Mode Fiber (MMF)

A fiber optic FDDI connection supports data transmission rates of up to 100 Mbps and, because attenuation in the fiber is very low, larger distances between FDDI stations are possible without the use of repeaters. In addition, fiber does not emit electromagnetic radiation and is electrically nonconductive. Therefore, it neither causes, nor suffers from, problems due to signal interference between adjacent equipment or cabling.

SunFDDI supports 62.5/125 micrometer, multimode fiber (MMF). Multimode fiber cannot support such large distances between FDDI stations as single-mode fiber (SMF), which is not supported by SunFDDI. However, multimode fiber is cheaper, and uses lower cost terminations, connectors, and light-sources.

*Physical (*PHY*) Layer*

The physical (PHY) layer handles the efficient encoding and decoding of digital data. It is also referred to as the *signal* layer.

At the transmitting end, the PHY layer encodes the digital data into FDDI *symbols* and passes them to the physical medium dependent (PMD) layer for transmission. At the receiving end, the PHY layer decodes the FDDI symbols and passes the digital data to the medium access control (MAC) layer. An FDDI symbol is the basic transmission unit in an FDDI network.

The PHY layer also provides the network synchronization mechanism. It uses a distributed clocking scheme under which each FDDI station has its own local clock to synchronize the transmission of outgoing data. The PHY layer extracts clocking information from incoming data as it is received.

*Media Access Control (*MAC*) Layer*

The media access control (MAC) layer specifies the access mechanism used to transmit and receive data on the FDDI network. It packages digital data in *frames.*

The MAC layer specifies three classes of digital data traffic:

- Synchronous (guaranteed) traffic
- Asynchronous (priority-based) traffic
- Restricted (dialogue-based) traffic

The MAC layer uses a *timed token rotation protocol* that regulates how much digital data can be sent at one time. This protocol ensures that the network bandwidth is used predictably and efficiently. It also gives the FDDI network its distinctive "ring" topology.

The FDDI stations connected on the network use a token to control the right to transmit data for a predefined time, determined by the local *timed target rotation timer* (TTRT) on each station. When an FDDI station completes transmission, it releases the token for use by downstream stations. In this way, the token rotates continuously around the ring.

Each attachment to an FDDI network is identified by a unique MAC address. The first SunFDDI SBus card installed in a machine takes its identity from the host-resident MAC address that is stored in nonvolatile memory on the motherboard of the machine in which it is installed. Each subsequent SunFDDI SBus card takes its identity from the card-resident MAC address stored in its own IDPROM.

Use the nf_macid(1M) utility to display the card-resident MAC address. See Chapter 9, "Using the SunFDDI Network Utilities" for detailed instructions.

*Station Management (*SMT*) Layer*

The station management (SMT) layer provides services to manage, control, and configure the FDDI network. It is also referred to as the *network management* layer.

At its lowest level, the SMT protocol handles *connection management* (CMT). This includes station initialization, the insertion and removal of stations on the network, and connection compatibility between stations. At a higher level, the SMT protocol handles *ring management* (RMT). This includes the detection of duplicate addresses and the isolation of fault conditions.

The SMT protocol also defines the FDDI *management information base* (MIB). This is a set of managed objects and associated attributes that includes the MAC entity (MAC), data path (PATH), attachment type (ATTACHMENT), and port identifier (PORT).

Communication services are used to exchange objects and information between peer SMT entities through special management frames that are carried at the same time as normal network traffic. This frame-based management is used to:

- Examine and modify FDDI station configuration

- Schedule synchronous (guaranteed) and asynchronous (priority-based) traffic

- Gather network statistics

- Generate status reports

SunFDDI includes SunNet Manager agents that access the SMT entity to collect and return FDDI statistics to a SunNet Manager console. See Chapter 10, "Managing FDDI Stations Using SunNet Manager" for detailed instructions on how to manage FDDI networks using a SunNet Manager console.

Communication Between FDDI Layers

The MAC layer transfers data between peer entities in the form of frames that are encoded as FDDI symbols by the PHY layer. The PMD layer transmits FDDI symbols across the network as a bit stream of light pulses (for fiber connections) or electrical signals (for twisted-pair connections).

The communication between layers in the FDDI architectural model is summarized in Figure 7-2.
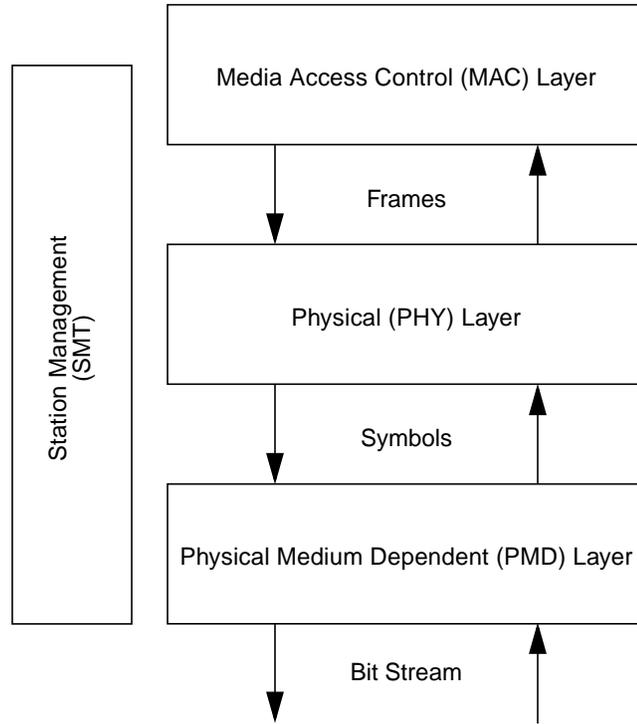
Figure 7-2   Communication Between FDDI Layers

FDDI Network Architecture

A typical FDDI network is based on a dual, counter-rotating ring, as illustrated in Figure 7-3. Each FDDI station is connected in sequence to two rings simultaneously—a primary ring and a secondary ring. Data flows in one direction on the primary ring, and in the other direction on the secondary ring.
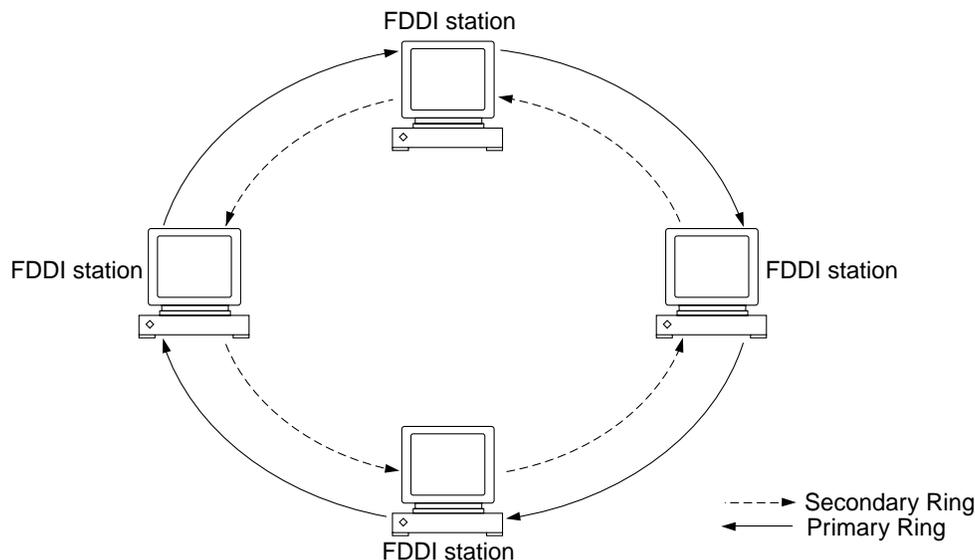
Figure 7-3   Basic FDDI Network Architecture

The secondary ring serves as a redundant path. It is used during station initialization, and may be used as a backup to the primary ring in the event of a station or cable failure. When a failure occurs, the dual-ring is "wrapped" around to isolate the fault and to create a single, one-way ring. The components of a typical FDDI network and the failure recovery mechanism are described in more detail in the following sections.

FDDI Stations

An FDDI station is any device that can be attached to a fiber or copper twisted-pair FDDI network through an FDDI interface. The FDDI protocols define two types of FDDI stations:

- Single-attached station (SAS)
- Dual-attached station (DAS)

Single-Attached Station (SAS)

A single-attached station (SAS) is attached to the FDDI network through a single connector called the S-port. The S-port has a primary input (Pin) and a primary output (Pout). Data from an upstream station enters through Pin and exits from Pout to a downstream station, as shown in Figure 7-4.

Single-attached stations are normally attached to single- and dual-attached concentrators, as described in "FDDI Concentrators" on page 84.
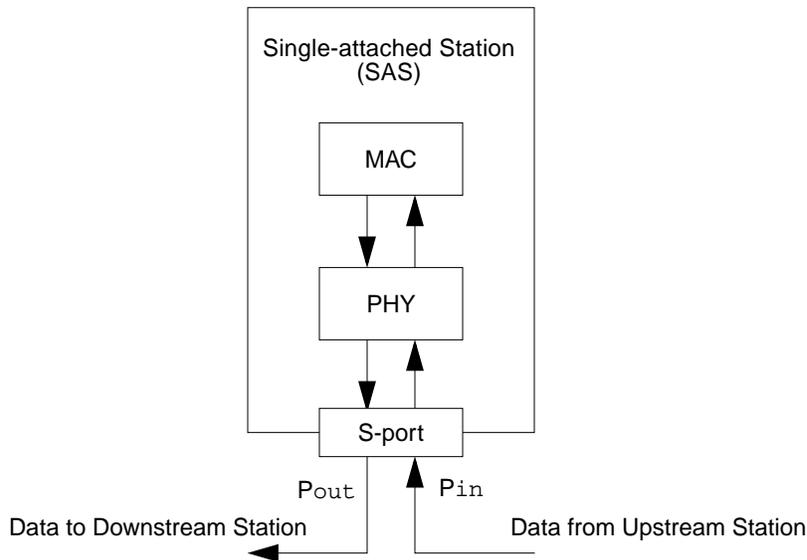
Figure 7-4   Single-Attached Station (SAS)

Dual-Attached Station (DAS)

A dual-attached station (DAS) is attached to the FDDI network through two connectors called the A-port and the B-port, respectively. The A-port has a primary input (`Pin`) and a secondary output (`Sout`); the B-port has a primary output (`Pout`) and a secondary input (`Sin`).

The primary input/output is attached to the primary ring and the secondary input/output is attached to the secondary ring. The flow of data during normal operation is shown in Figure 7-5.

To complete the ring, you must ensure that the B-port of an upstream station is always connected to the A-port of a downstream station. For this reason, most FDDI DAS connectors are keyed to prevent connections between two ports of the same type.
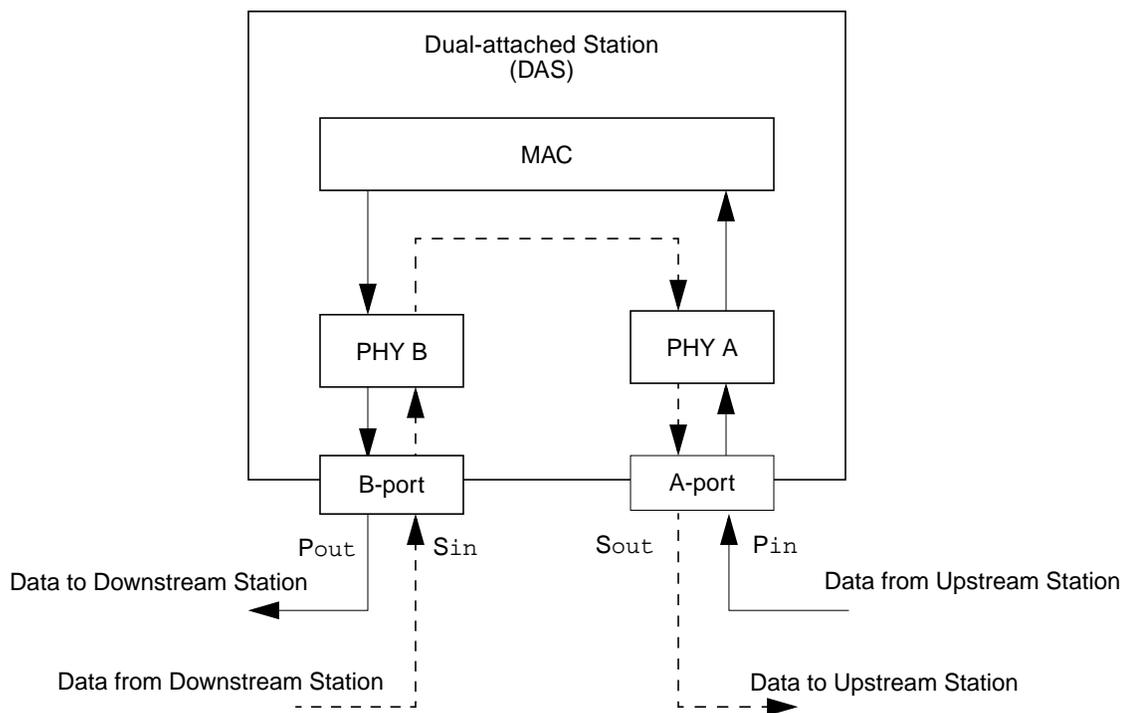
Figure 7-5   Dual-Attached Station (DAS)

FDDI Concentrators

FDDI concentrators are multiplexers that attach multiple single-attached stations to the FDDI ring. An FDDI concentrator is analogous to an Ethernet hub.

The FDDI protocols define two types of concentrators:

- Single-attached concentrator (SAC)
- Dual-attached concentrator (DAC)

Single-Attached Concentrator (SAC)

A single-attached concentrator (SAC) is attached to the FDDI network through a single connector, which is identical to the S-port on a single-attached station. It has multiple M-ports, to which single-attached stations are connected, as shown in Figure 7-6.
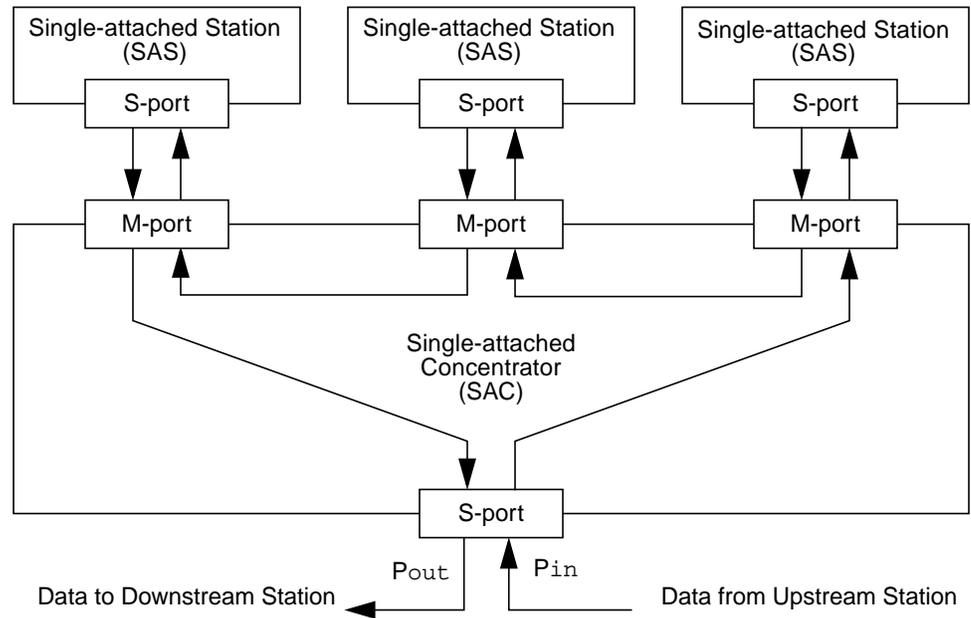
Figure 7-6 Single-Attached Concentrator (SAC)

Dual-Attached Concentrator (DAC)

A dual-attached concentrator (DAC) is attached to the FDDI network through two ports—the A-port and the B-port, which are identical to the ports on a dual-attached station. It has multiple M-ports, to which single-attached stations are connected, as shown in Figure 7-7.

Dual-attached concentrators and FDDI stations are often arranged in a very flexible network topology called the *ring of trees*. This configuration is discussed in more detail in Chapter 8, "FDDI Network Topologies."
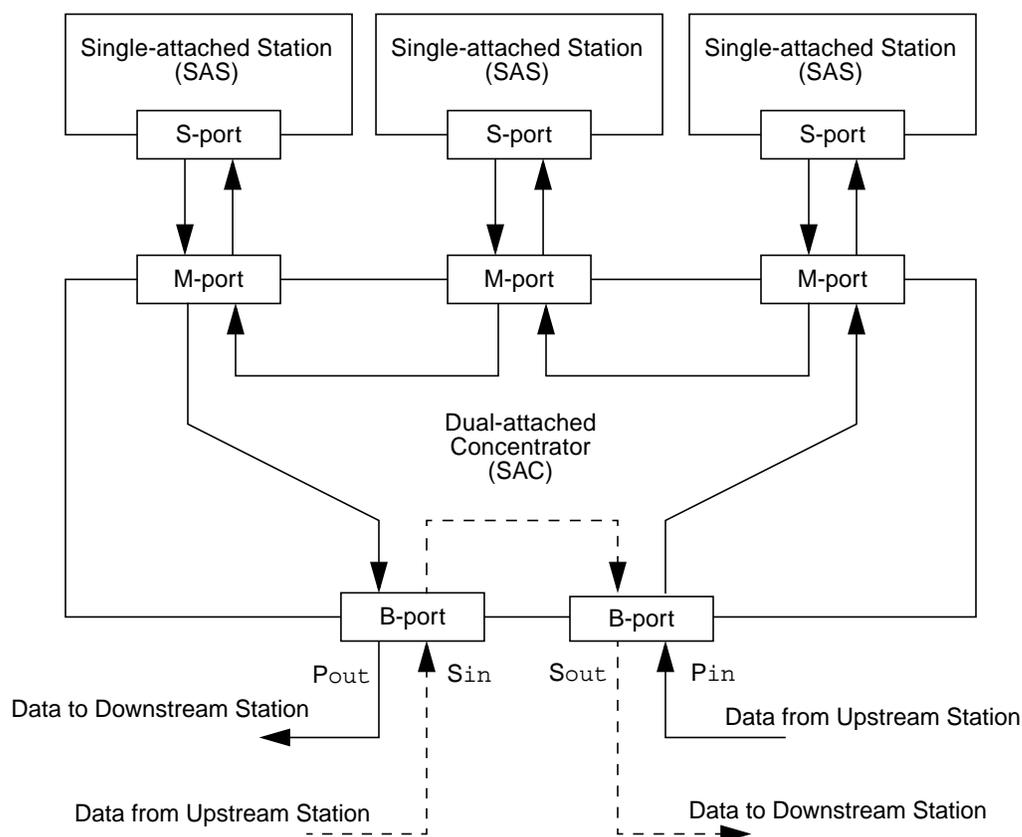
Figure 7-7   Dual-Attached Concentrator (DAC)

FDDI Failure Recovery

One of the primary advantages of FDDI is its ability to recover reliably from failures in stations and cables. The failure mechanism is implemented and controlled by the station management (SMT) entity described in the section "Station Management (SMT) Layer" on page 78.

A failure could be something as insignificant as someone switching off their workstation; the ability of FDDI to recover from such an event increases the reliability of the network significantly.

Station Wrapping

Figure 7-8 shows the effect of a single-station failure in an FDDI network that comprises four dual-attached stations connected in a basic ring configuration. When a failure occurs, the SMT entities on the stations on either side of the fault reconfigure the network dynamically to isolate it. In this condition, the primary and secondary rings are wrapped on to each other to form a single, one-way ring.
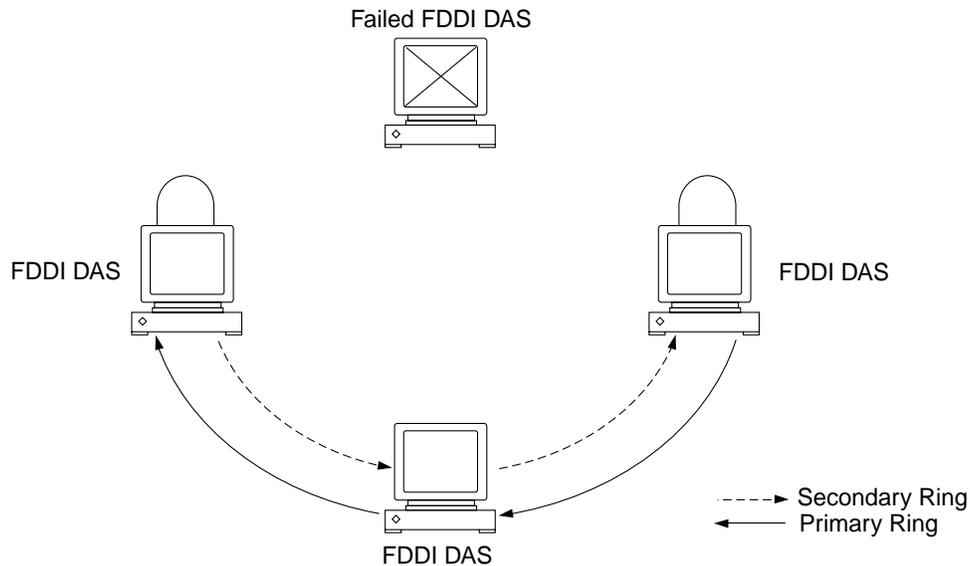
Figure 7-8   Isolating a Single Station Failure

This failure recovery mechanism is only supported by dual-attached stations and concentrators; a single-attached station connected directly to the ring cannot wrap around the fault because it is only connected to one ring at a time.

The effect of a cable or link failure on a basic FDDI network is very similar to a station failure, as shown in Figure 7-9. The SMT entities on the stations on either side of the failed connection reconfigure the network dynamically so that the primary ring is wrapped onto the secondary ring.
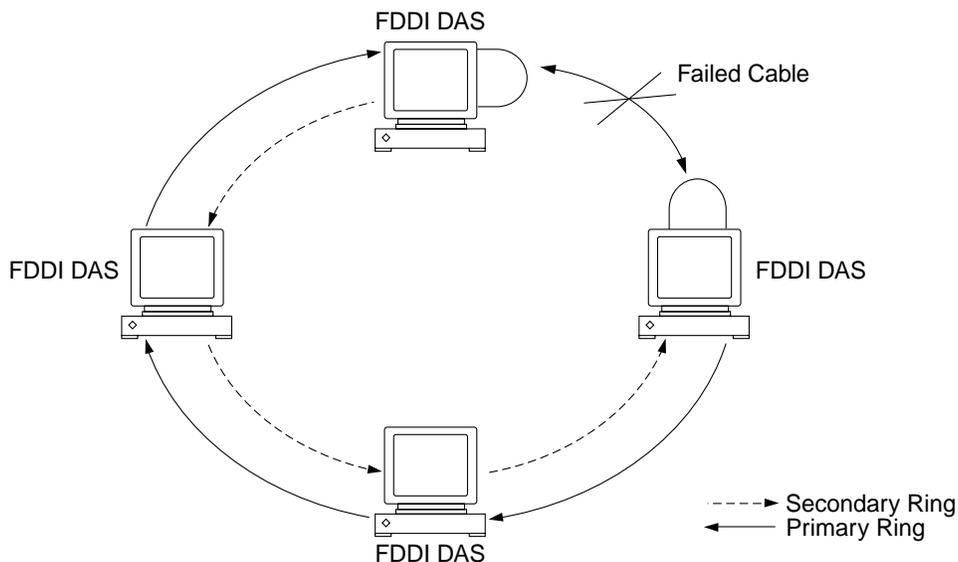
Figure 7-9   Isolating a Single Cable Failure

Wrapping occurs within the MAC layer. Figure 7-10 on page 88 illustrates how the dual-attached station is reconfigured to wrap the primary ring onto the secondary ring. Dual-attached concentrators wrap in a similar way. In this case, all of the single-attached stations connected to the concentrator are also wrapped onto the secondary ring.

Dual-attached Station
(DAS)

MAC

Primary and Secondary Rings
Wrapped within the MAC Layer

PHY B

PHY A

port B

port A

Pout

Sin

Data to Downstream Station

Data from Upstream Station
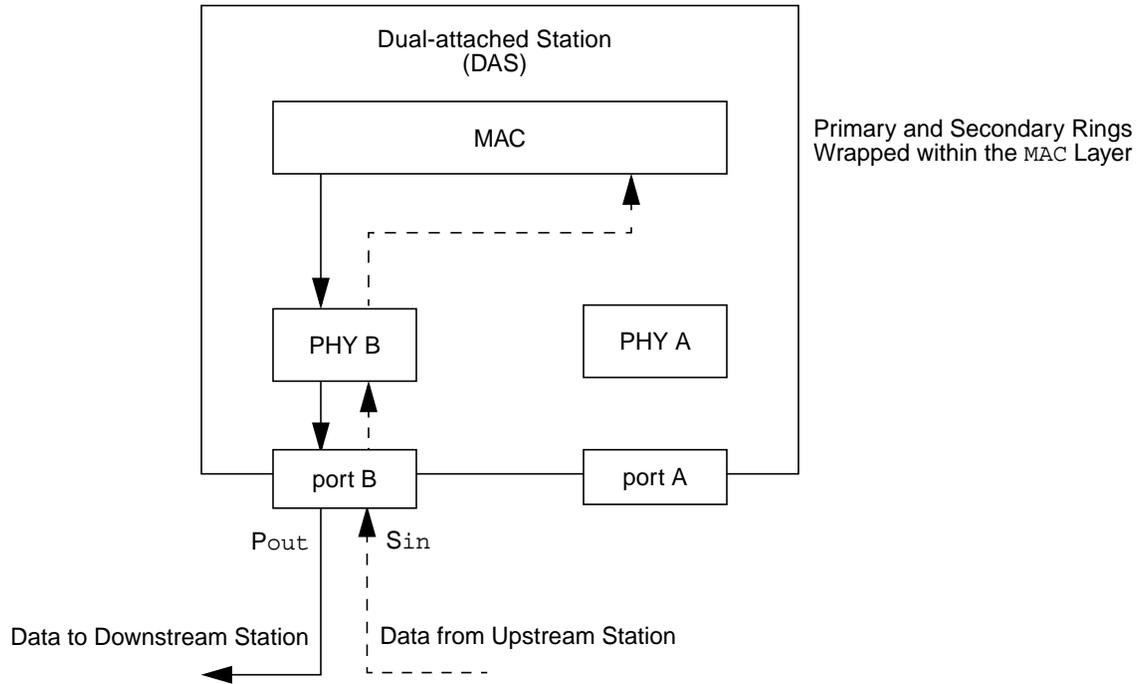
Figure 7-10 Dual-Attached Station in Wrap Mode

Optical Bypass Switches

Station wrapping provides effective network recovery in the event of a single station or cable failure. However, two or more failures in the ring will isolate portions of the network, as shown in Figure 7-11.

This limitation can be overcome to a certain extent by fitting *optical bypass switches* to the dual-attached stations. As their name suggests, these switches provide an optical connection that bypasses the station in the event of failure, or if the station needs to be removed from the ring. Optical bypass switches can only be used to overcome station faults; they have no affect in the event of a cable fault.

The attenuation in an optical bypass switch is far greater than in a normal FDDI connection; therefore, the number of optical bypass switches that can be connected in series in a single ring is limited.
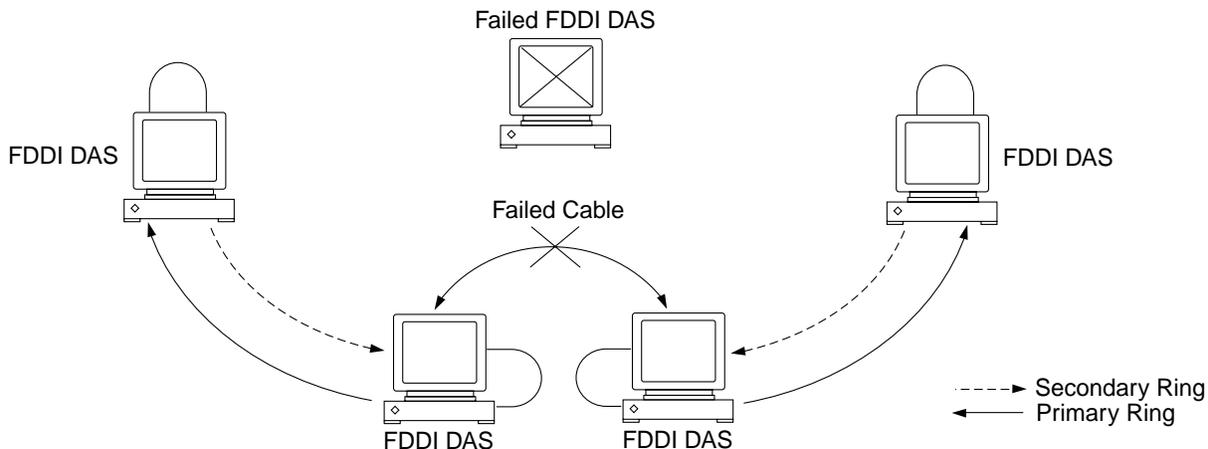
Figure 7-11 Dual Ring Network Divided by Two Faults

The maximum number of active optical bypass switches that can be connected in a series is four. This assumes that the maximum distance between stations in the ring is no more than 400m; otherwise, the aggregate attenuation in the ring exceeds the total optical power budget.

Figure 7-12 shows how connecting an optical bypass switch to a dual-attached station is used to conserve the connection when the station is switched off, or removed from the ring.
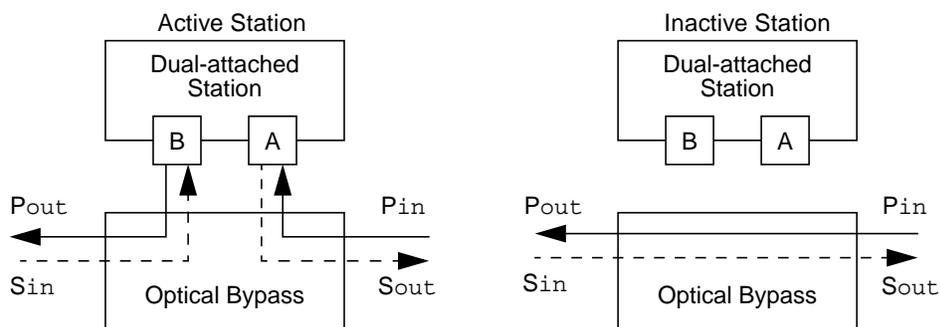
Figure 7-12 Optical Bypass Switch

Figure 7-13 shows the occurrence of two different fault conditions in an FDDI network that includes optical bypass switches. The station fault is bypassed effectively to conserve the majority of the network intact; however, the cable fault still causes station wrapping.
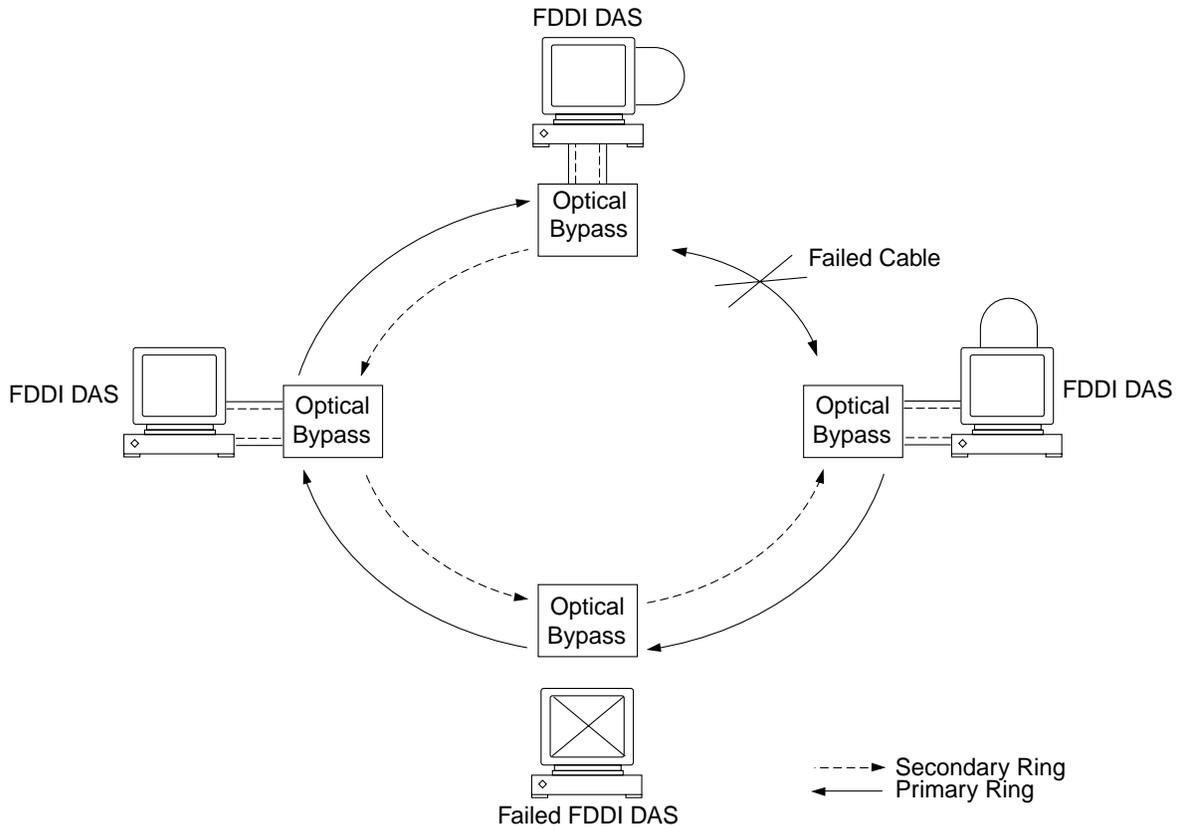
Figure 7-13   Optical Bypass Switches used in a Network

FDDI Ring Operation

Two types of traffic are allocated bandwidth on an FDDI ring:

- **Asynchronous traffic** (unrestricted and restricted)
  The unrestricted asynchronous service allocates bandwidth dynamically based on the *timed token rotation protocol*, and according to the priority assigned to the traffic. It does not guarantee bandwidth; therefore, unrestricted asynchronous traffic may be delayed on a heavily loaded network.

  The restricted asynchronous service allocates bandwidth for extended transmission based on *restricted token exchange* between communicating stations. During restricted transmission, the timed token rotation protocol is suspended until the restricted token is released. The use of the restricted asynchronous service is relatively rare.

- **Synchronous traffic**
  For synchronous operation, a specific portion of the bandwidth is reserved for synchronous traffic and a boundary is placed on the access delay (the time that a station must wait before being able to transmit). The synchronous service is assigned the highest priority on the ring and is normally used to transfer voice, video, and real-time process control data, where the access delay can have significant impact.

Note – The network user (the application) requests the synchronous service and manages synchronous operation. Synchronous transmission is not supported for SunFDDI running in a Solaris 1.*x* environment.

Unrestricted Asynchronous Transmission

The FDDI network is most commonly used for unrestricted asynchronous traffic. This is the default mode for SunFDDI and is the only mode supported in a Solaris 1.*x* environment. Access to the FDDI ring is controlled by a *permission token*, or right to transmit. The permission token is a special frame that is passed between stations. Unlike the tokens used in a token ring network, the FDDI permission token contains no additional information.

The station holding the permission token completes transmission either when it has no more packets to transmit, or when its right to hold the token expires. It then releases the token, which is now available for use by the next station on the ring. In this way, the permission token rotates around the ring, at a rate determined by the overall *target token rotation time* (TTRT).

The TTRT is established when the ring is initialized, based on a bidding procedure called the *claim process.* During the claim process, each station puts in a request (T_req) for the TTRT. The lowest bid or, the fastest rotation time wins the claim process and is stored in 2's complement form as the operating value of TTRT (T_opr). The maximum value of TTRT (T_max) allowed by the MAC layer is implementation dependent, but is always in the ~165 ms range. SunFDDI has T_max=167.874 ms.

The maximum time for which a station can hold the permission token, and therefore the time for which a station can transmit on the ring, is determined by these two station timers that work together to maintain the target token rotation time (TTRT), irrespective of the load on the network:

- **Token Rotation Timer (TRT)**
  The TRT measures the time between successive arrivals of the token or, the time taken for the token to rotate once around the ring. The TRT is reset to the operating value of TTRT (T_opr) each time the token is received. If the TRT expires before the token arrives, the token is considered late and the flag Late_Ct is set. Only synchronous (high priority) traffic can be transmitted when Late_Ct is set.

- **Token Hold Timer (THT)**
  The THT determines how long a station can hold a token or, how long a station can transmit asynchronous traffic. It is loaded with the value remaining in the token rotation timer (TRT) each time the token arrives. Thus, the faster the token rotates, the more transmission time is allocated to each station. If the token is delayed because the network is heavily loaded, the amount of transmission time is reduced. If THT expires during transmission, the current transmission is completed before the token is released.

The interaction of these two timers causes token rotation to increase (reduced transmission time per station) in a heavily loaded network, and to decrease (increased transmission time per station) in a lightly loaded network.

FDDI Performance

Although the FDDI standards define a medium that supports data transfer rates of up to 100 Mbps, this does not translate directly to an increase in overall system performance. If fact, it frequently transfers the bottleneck elsewhere, so that the FDDI connection is not used to full capacity.

An FDDI ring has the *potential* to carry more information rather than to be a faster connection. If the applications running over the network do not use the available bandwidth efficiently, you will not see much improvement in the performance of your network above that of traditional Ethernet connections.

Among the factors that affect FDDI network performance are:

- Network topology
- Network load
- Network efficiency
- Processor speed (on the FDDI stations)
- Bus architecture

There are a number of ways of improving the overall performance of the network, some of which are discussed in Chapter 5, "Improving Network Performance." However, the majority of these suggestions should only be undertaken by experienced system administrators. Any improvement made by modifying the actions of the FDDI network is negligible compared to what is gained by making more efficient use of the available bandwidth.

≡ *7*

FDDI Network Topologies 8

FDDI networks can be arranged in a variety of ways, depending on the placement of stations (SAS and DAS) and the use of concentrators (SAC and DAC). The optimum arrangement for a particular installation is dependent on several factors, including:

- Cost
- Network size
- Required Bandwidth
- Type of network traffic
- Fault resistance and network reliability

This chapter describes some of the common FDDI network topologies and discusses the primary advantages and disadvantages of each.

Basic Dual-Ring Network

The dual ring (or dual, counter-rotating ring) is one of the simplest FDDI network topologies. It clearly illustrates the distinctive ring architecture most commonly associated with the FDDI standards, as shown in Figure 8-1. Each station is critical to the operation of the network; therefore, the basic dual-ring topology is best adapted to small, stable networks that are not subject to frequent reconfiguration.

In a dual-ring network, dual-attached stations are connected directly to the primary and secondary rings. Data, and the token that controls the flow of data, are transmitted in one direction on the primary ring. Data flows in the other direction on the secondary ring, which is used during ring initialization and as a backup in case of a ring failure.
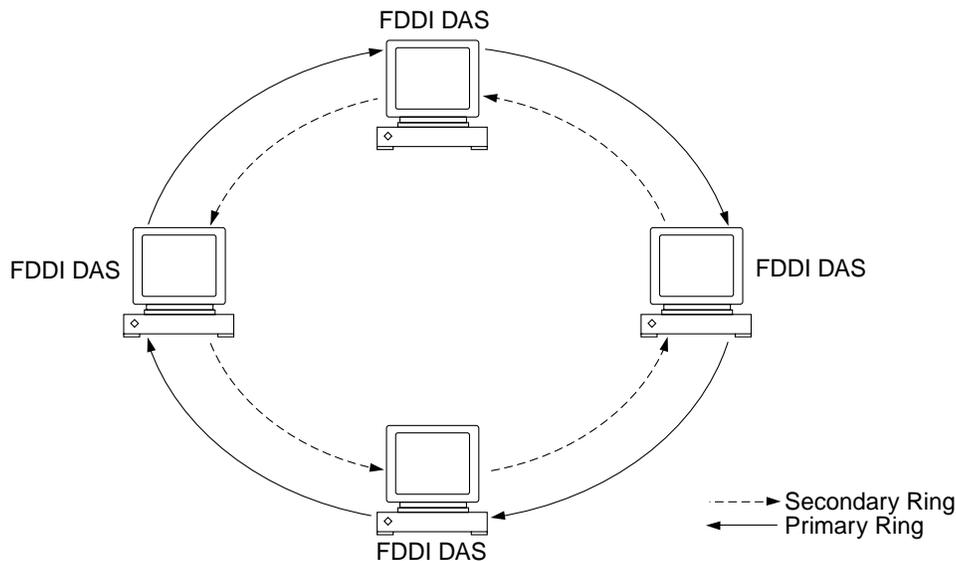
Figure 8-1   Basic Dual-Ring Network

Advantages

The primary advantages of the dual-ring network topology are its simplicity and its ability to recover from simple station and line faults. The secondary ring provides an effective backup in the event of a *single* failure in the ring.

When a ring fails, the primary ring is wrapped automatically on either side of the fault so that the primary and secondary rings are combined to form a single, one-way ring. This mechanism is described in more detail in "FDDI Failure Recovery" on page 86. A dual-ring network does not require a concentrator.

Disadvantages

Although the dual-ring topology is resistant to single failures in the ring, two or more failures break the network into parts. Small fragments of the network can still function, but they are isolated from the other stations. Figure 8-2 shows how two faults in a network, with five dual-attached stations, isolates two parts of the network. Cable and connection costs can be high in large installations because there are two cables between each station.
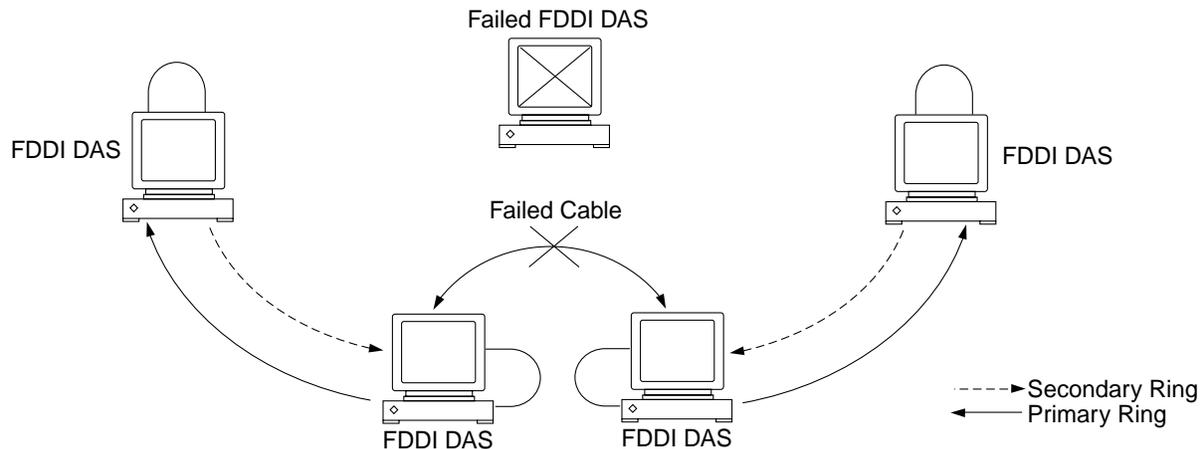
Figure 8-2   Basic Dual Ring Network with Two Faults

Standalone Concentrator

Figure 8-3 shows multiple single-attached stations connected to a single, dual-attached concentrator through its M-ports. The concentrator can also be connected to an external dual ring through its A- and B-ports.

A standalone concentrator provides a stable, low-cost alternative for small work groups that do not require the fault recovery facility provided by the dual-ring configuration.
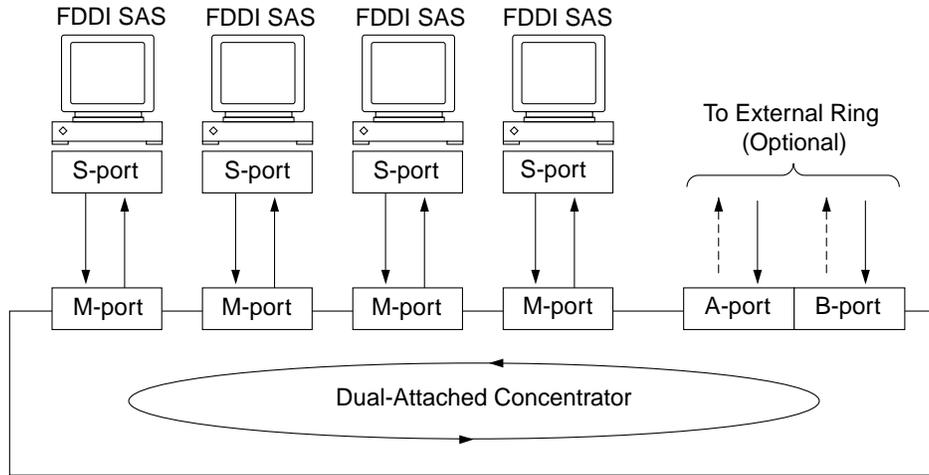
Figure 8-3   Standalone Concentrator

The typical ring architecture of the FDDI network is less obvious in this topology because it exists within the concentrator itself. For this reason, this arrangement of stations is usually described as a *tree*, with the concentrator as the root.

Advantages

In the standalone concentrator configuration, individual stations have less influence on the operation of the network, which is controlled by the concentrator. Concentrators are inherently more stable than FDDI stations. They do not have monitors, or disk drives, are subject to more predictable usage, and are less likely to be switched off. As a result, a standalone concentrator provides a more reliable network than the basic dual-ring configuration described on page 96.

Concentrators are equipped with built-in electrical bypass facilities that isolate single-station faults. Unlike the station optical bypass facility described on page 88, there is no limit to the number of stations that can be bypassed using the electrical switches in concentrators.

The majority of stations attached to the concentrator are single-attached stations; therefore, only one cable is required for each station.

More flexibility is allowed in the physical location and wiring configuration. Since the stations do not have to be attached in any fixed order and all cables return to a central concentrator, this configuration is useful at sites where FDDI cable has already been installed.

The A- and B-ports on a concentrator can be used to connect it to an external dual-ring configuration. This is a common configuration called the *ring of trees*, which is discussed on page 104.

Disadvantages

The number of stations that can be attached to a single concentrator is limited by the number of M-ports. This is typically in the 2 to 32 range.

The cost of a concentrator is significantly higher than that of a single-attached or dual-attached station; however, there are some low cost concentrators available that do not support all SMT management functions.

Although concentrators are more stable than FDDI stations, when the concentrator goes down, the entire network goes down with it.

Concentrators with Dual-Homing

Figure 8-3 shows two dual-attached stations connected to two dual-attached concentrators in a dual-homing configuration. In this case, each dual-attached station is connected to both DACs. This topology is typically used for connecting critical systems such as file and name servers.
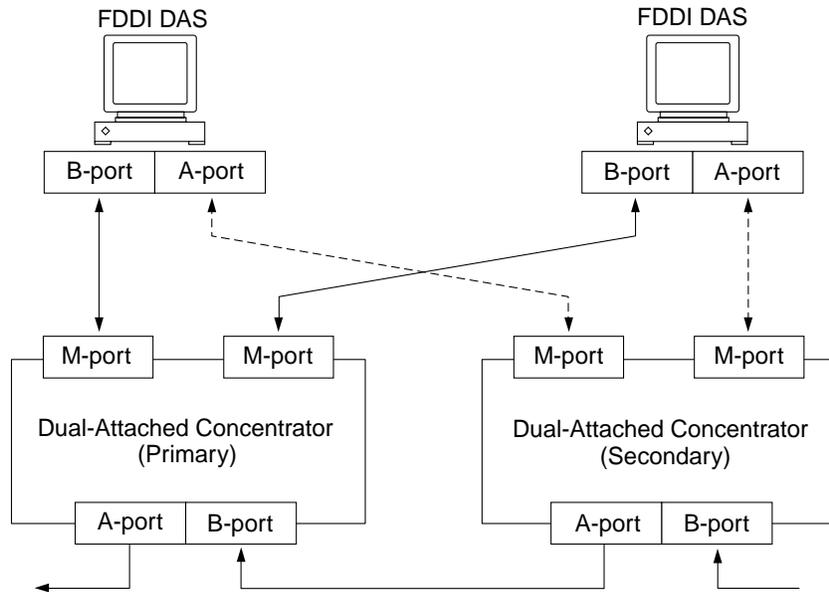
Figure 8-4   Standalone Concentrator

Dual-homing provides two independent data paths for each dual-attached station. Under normal conditions, the station communicates on its primary path through the B-port. In the event of a cable or concentrator failure, the station switches to the secondary path connected through the A-port.

Dual-homing is equivalent to the *redundant single-attached station* (RSAS) configuration, which was supported by SunFDDI 2.0. In the RSAS configuration, two single-attached interfaces are used to emulate a dual-attached interface connected in a dual-homing configuration. RSAS is not supported by SunFDDI 5.0.

Advantages

Dual-homing offers the same advantages as a standalone concentrator, described on page 97. It also offers improved resistance against cable faults and concentrator failure.

Disadvantages

The number of stations that can be attached to the concentrator is limited by the number of M-ports, which is typically between 2 and 32. Since each dual-homed station requires two M-ports, only a limited number of stations can be connected in this way.

The dual-homing configuration requires a minimum of two concentrators, which are more expensive than single-attached or dual-attached stations.

Tree of Concentrators

The *tree of concentrators* is an incremental expansion of the standalone
concentrator topology, described on page 97. Two or more concentrators are
connected in a hierarchical topology, with one concentrator designated as the
root of the tree, as shown in Figure 8-5. This topology is typically used to
connect a large number of stations within a single building or office.
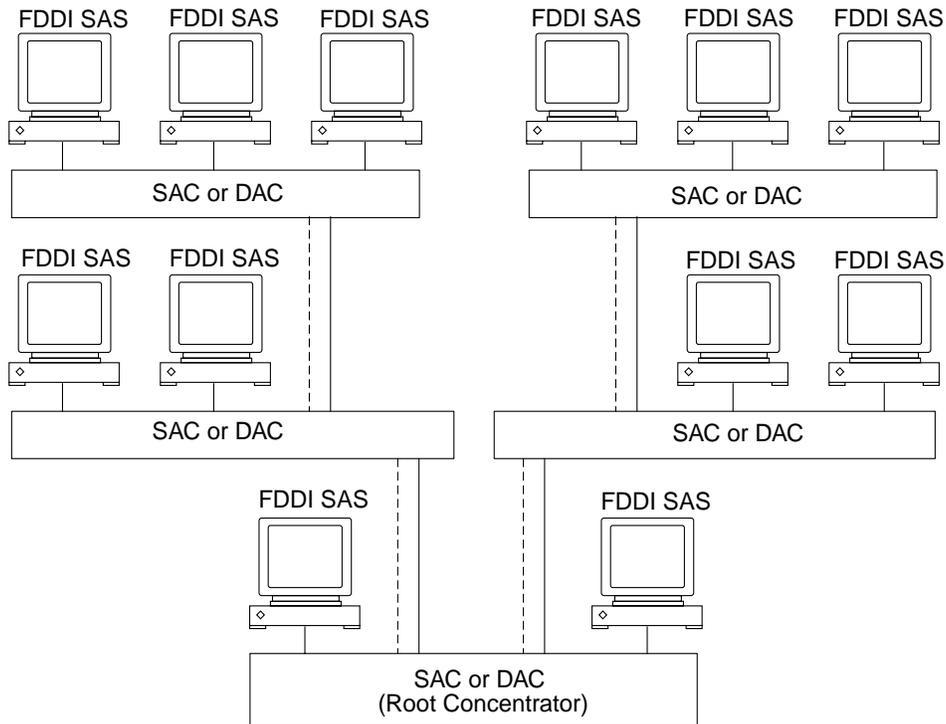
Figure 8-5   Hierarchical Tree of Concentrators

Advantages

The tree of concentrators configuration offers many of the advantages offered
by the standalone concentrator, described on page 97; however, it allows a
much larger number of stations to be connected.

The cumulative length of the network is increased significantly because the limitation on distance occurs between the end-station and the nearest concentrator, and not between the end-station and the root concentrator.

Disadvantages

This configuration suffers from the same disadvantages as the standalone concentrator, described on page 97. The number of stations connected to each concentrator is limited by the number of M-ports, and the loss of a concentrator takes down all of the stations to which it is attached.

Ring of Trees

The *ring of trees* is a derivative of the tree of concentrators topology described on page 102. The root concentrator is replaced by a dual-ring to which dual-attached concentrators are connected, as shown in Figure 8-6. This topology is ideal for connecting a large number of stations in a campus or multiple-building network.
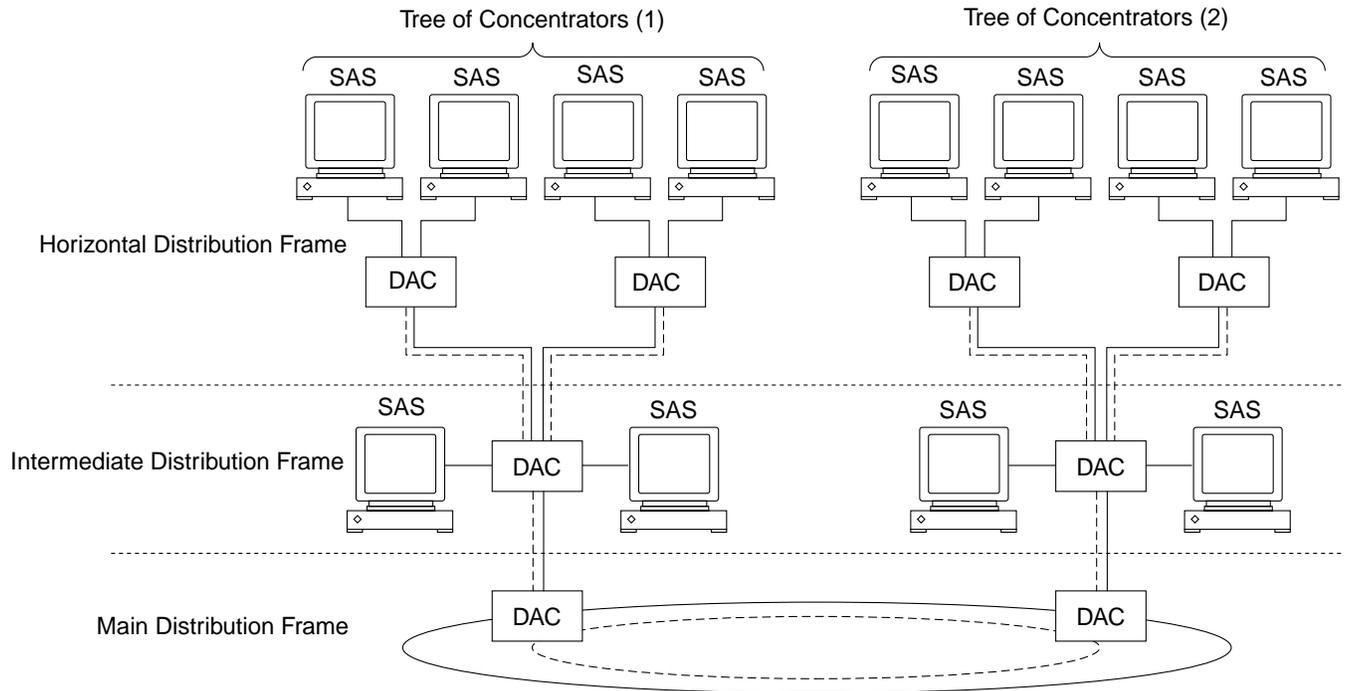
Figure 8-6   Ring of Trees

The dual-ring, and the concentrators attached to it directly, are called the *main distribution frame*. Only one main distribution frame per network is allowed.

The second level in the hierarchy is called the *intermediate distribution frame* and consists of the root concentrators in each building, and any stations connected to them directly.

The third level in the hierarchy is called the *horizontal distribution frame* and consists of concentrators connected to distributed stations, usually through wall boxes.

Advantages

The ring of trees configuration combines many of the advantages of the basic dual-ring and the concentrator-based topologies. It offers good fault tolerance and is a very flexible configuration.

Since only a few stations (the root concentrators) are attached to the dual ring, optical bypass switches can be used to increase the reliability of the main distribution frame, as described in "Optical Bypass Switches" on page 88. The small number of stations and the increased reliability of the concentrators means that the chances of having more than three adjacent stations bypassed at one time are minimal.

The cumulative length of the network can be increased because the limitation on distance occurs between the end-station and the nearest concentrator, and not between the end-station and the root concentrator.

Disadvantages

Only the concentrators attached directly to the dual-ring benefit from its inherent redundancy.

Mixed FDDI/Ethernet Networks

This section concentrates on the implementation of mixed FDDI/Ethernet networks. The same principles could be applied to mixed FDDI/802.3 and mixed FDDI/802.5 (Token Ring) networks. FDDI and Token Ring networks are based on a ring topology; Ethernet and 802.3 networks are based on a bus topology.

FDDI/Ethernet IP Router

The machines at the boundaries between the FDDI and Ethernet networks in the following examples are assumed to be Ultra systems that have at least one SunFDDI interface and one Ethernet interface installed. These machines operate as IP routers.

Special software is not required. The combination of the SunFDDI driver and the SunOS IP routing implementation handles the routing and relaying of packets between FDDI and Ethernet stations. Note that the packets on the FDDI ring can be as long a 4500 octets, whereas Ethernet packets are restricted to a maximum of 1500 octets.

Figure 8-7 illustrates the basic topology of a mixed FDDI/Ethernet network. In this example, there are two FDDI/Ethernet IP routers that represent the boundaries between the three networks.
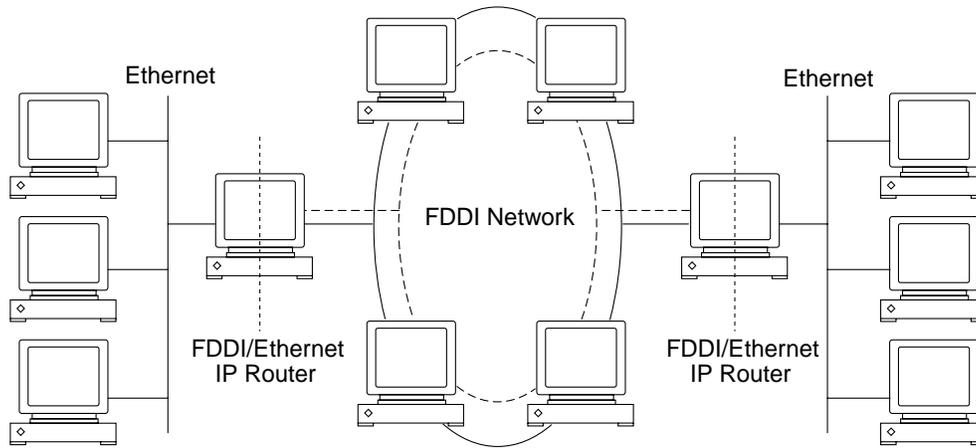
Figure 8-7   Basic Mixed FDDI/Ethernet Topology

A Tree of Concentrators in an Ethernet Network

A tree of concentrators of the type described on page 104 is attached to an Ethernet network by connecting your system to the root concentrator as shown in Figure 8-8
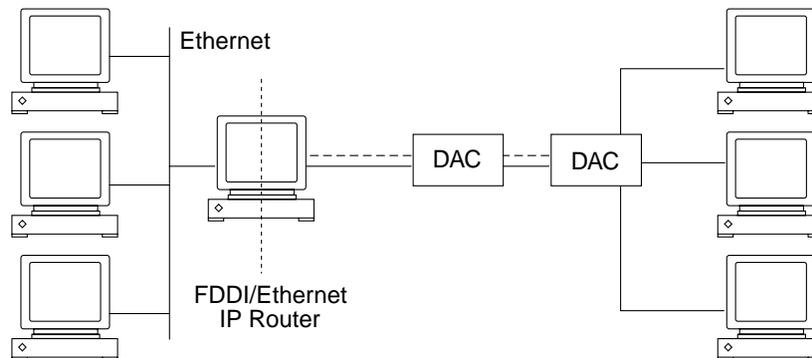
Figure 8-8   Tree of Concentrators Attached to an Ethernet Network

A Ring of Trees in an Ethernet Network

A ring of trees of the type described on page 104 is attached to an Ethernet network by connecting your system to one of the concentrators in the main distribution frame, as shown in Figure 8-9.
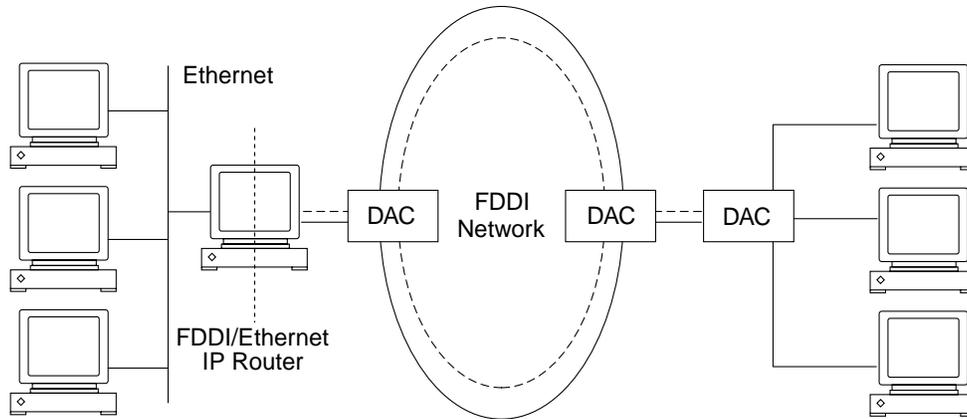
Figure 8-9   Ring of Trees Attached to an Ethernet Network

Routes in Mixed FDDI/Ethernet Networks

The FDDI/Ethernet IP router is recognized by the IP address and host name of its FDDI interface by the FDDI network; it is recognized by the IP address and host name of its Ethernet interface by the Ethernet network. Note that an FDDI/Ethernet router usually has the same MAC address on both networks.

In general, stations in an internetwork can be reached via multiple routes. The standard routing daemon (routed) included in the SunOS determines the optimum route between stations based on relative hop count, and not on the speed of the interface. Therefore, in cases where the hop count is equal, the use of the faster FDDI network is not guaranteed.
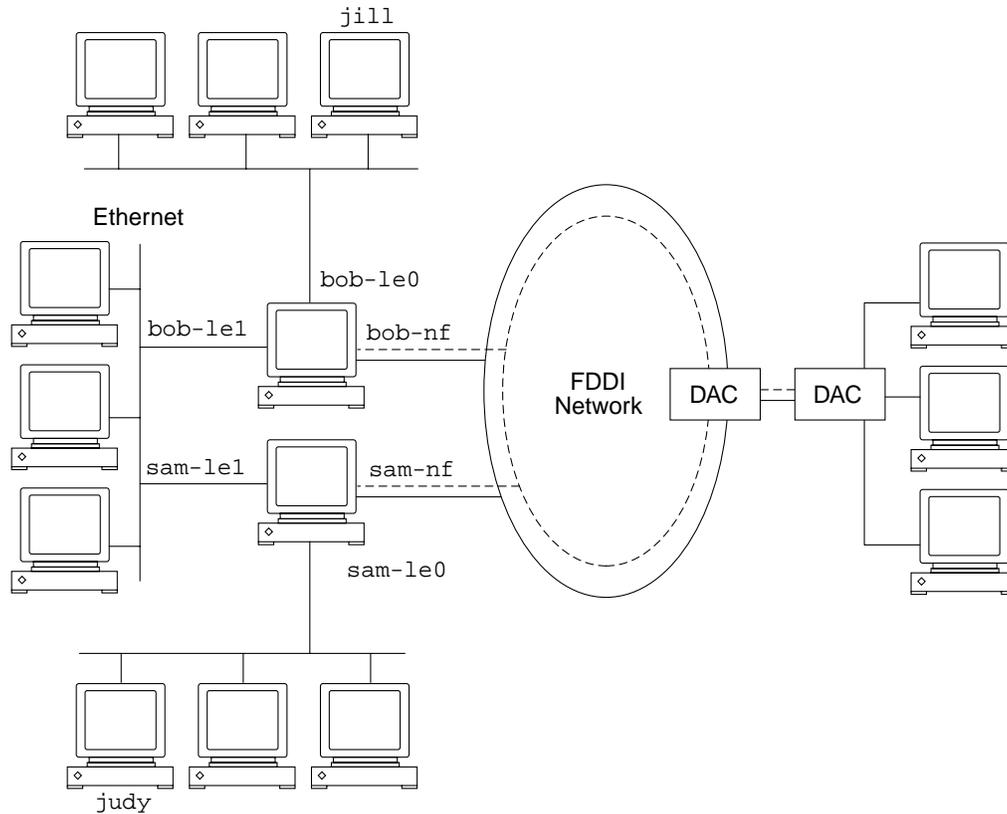
Figure 8-10   Routing in an FDDI/Ethernet Internetwork

For example, in the FDDI/Ethernet internetwork shown in Figure 8-10, there are parallel routes between `jill` and `judy` across relatively faster and slower networks:

• **Route 1** (across FDDI link):
  `jill➔bob-le0➔bob-nf➔sam-nf➔sam-le0➔judy`
• **Route 2** (across Ethernet link):
  `jill➔bob-le0➔bob-le1➔sam-le1➔sam-le0➔judy`

An identical number of hops between the two stations exists regardless of the route taken; therefore, `routed` may send all of the traffic across the slower Ethernet link in preference to the faster FDDI link.

To avoid this situation, set up a static route that forces the traffic between `jill` and `judy` to pass over the FDDI link; however, static routing is not recommended in a network beyond a few nodes in size. For efficient IP routing in large FDDI/Ethernet networks, use a routing package that provides routing metrics *in addition* to hop count (for example, `gated`, public domain software available by the Center for Theory and Simulation in Science at Cornell University).

To improve the end-to-end system performance across the boundary between the networks, you may need to alter the default MTU for the FDDI network (see "Performance Across Bridges" on page 41) or to disable the MTU path discovery feature introduced by the Solaris 2.*x* environment (see "Improving Performance in a Solaris 2.x Environment" on page 43).

Using the SunFDDI Network Utilities

9

Changing the Default MAC Address (`nf_macid`)	*page 112*
Displaying SunFDDI Statistics (`nf_stat`)	*page 113*
Monitoring SMT Frames (`nf_smtmon`)	*page 124*
Configuring Synchronous FDDI (`nf_sync`)	*page 128*

This chapter describes the network utilities delivered with SunFDDI. The behavior of the SunFDDI network utilities is identical for both Solaris 2.*x* and Solaris 1.*x* environments.

Throughout this chapter, it is assumed that you have installed the SunFDDI software under the default base directory *<basedir>* for your operating system:

For a Solaris 2.*x* environment, the default base directory *<basedir>* is:

```
/opt/SUNWconn/bin
```

For a Solaris 1.*x* environment, the default base directory *<basedir>* is:

```
/usr/sunlink/nf/usr/etc
```

≡ *9*

Changing the Default MAC Address (`nf_macid`*)*

Each attachment to an FDDI network is identified by a unique 48-bit MAC address. By default, the first SunFDDI SBus card takes the host-resident MAC address, which is stored in nonvolatile memory (NVRAM) on the motherboard of the machine. Each subsequent SunFDDI SBus card adopts the card-resident MAC address stored in its own IDPROM.

In general, this convention is sufficient to ensure that each SunFDDI SBus card installed in the machine has a unique MAC address. However, there may be a conflict with other LAN interfaces that also take the host-resident MAC address—for example, an Ethernet (`le`) interface, or a SunFDDI 2.0 (`bf`) interface. In this event, change the default MAC address assigned to the first SunFDDI SBus card installed in the system.

▼ To Change the Default MAC Address with `nf_macid`

Use the `nf_macid(1M)` utility to recover the card-resident MAC address, and then modify the system files to override the default MAC address:

1. **Log in as** `root` **or become** `superuser`**.**

2. **Use the** `nf_macid(1M)` **utility to recover the MAC address from the IDPROM on the SunFDDI interface identified by the instance number** *<inst>***.**

```
#  <basedir>/nf_macid nf<inst>
<mac_address>
```

3. **Modify the start-up file on your machine so that the MAC address is assigned correctly when the system is rebooted.**

- For a Solaris 2.*x* environment, edit the `/etc/rcS.d/S30rootusr.sh` file to add the following `if` statement immediately *after* the `ifconfig` command that initializes the interface nf*<inst>*. If you are changing the MAC address of more than one interface, add one `if` statement for each interface.

`if` statement to initialize
interface nf*<inst>* -------->

```
ifconfig $1 plumb
    if [ $1 = "nf<inst>" ]; then
        ifconfig nf<inst> ether <mac_address>
    fi
```

On most systems, the `/etc/rcS.d/S30rootusr.sh` file is a hard link to the `/etc/rootusr` file.

- For a Solaris 1.*x* environment, edit the file `/etc/rc.local` to add the following line immediately *before* the `ifconfig` command that initializes the interface nf*<inst>*. If you are changing the MAC address of more than one interface, you should add one declaration for each interface.

Command to initialize
interface nf*<inst>* ---->

```
ifconfig nf<inst> ether <mac_address>
ifconfig nf<inst> <hostname> netmask + -trailers up
```

4. **Reboot your machine to assign the new MAC address to the SunFDDI interface.**

When a SunFDDI SBus card takes the host-resident MAC address, it can be swapped to another system without affecting the existing network. However, once a station starts sending packets on the network, the Address Resolution Protocol (ARP) updates the ARP tables on other stations to include the MAC address of its interface. The ES-IS protocol performs the same function for SunFDDI OSI running over FDDI. If you swap SunFDDI SBus cards that use the card-resident MAC address, you must wait until the ARP entries time-out, or remove the ARP entries from every active station manually before packets can be routed correctly.

Displaying SunFDDI Statistics (`nf_stat`*)*

The `nf_stat(1M)` utility interrogates a specified SunFDDI interface and displays the accumulated statistics. This command must be executed as `root` (or `superuser`) and has the general form:

```
# <basedir>/nf_stat [-m] nf<inst> [<interval>] [<count>]
```

nf*<inst>* specifies the SunFDDI interface

<interval> is the elapsed time (in seconds) between interrogations

<count> the total number of interrogations

The `nf_stat` utility displays information using column headings that conform to SMT revision 7.3, which differ from SMT revision 5.1 and 4.2 headings in the following cases:

- The ECM heading corresponds to the 5.1 MIM heading.

- The RMT heading does not have an analog in SMT revision 4.2. If you run SunFDDI at revision level 4.2, ignore any data displayed under the RMT heading of `nf_stat`.

Displaying Local Interface Statistics

When you enter the `nf_stat` command *without* the `-m` option, it displays statistics recovered from the local interface nf*<inst>*.

For example, to display the accumulated statistics for the interface `nf0`, type:

```
# <basedir>/nf_stat nf0
Ring    ECM        RMT        PCMS    Ring_OP      XmitP      RecvP
UP       IN    RING_OP     ACTIVE          c      16fde      1862d
```

You can also monitor the interface dynamically (active monitor), by specifying the *interval* (the elapsed time between interrogations) and *count* (the total number of interrogations). This displays the incremental difference between the current state and the previous state. The minimum interval is one second and the accumulated statistics are displayed after every tenth interrogation.

For example, to monitor the interface `nf0` once every 60 seconds for 3 minutes (a total of 3 interrogations), type:

```
# <basedir>/nf_stat nf0 60 3
Ring   ECM      RMT      PCMS    Ring_OP     XmitP      RecvP
UP     IN    RING_OP   ACTIVE        c      131a0      131aa
UP     IN    RING_OP   ACTIVE        0          0          0
UP     IN    RING_OP   ACTIVE        0          1          1
```

Interpreting Local Statistics

Running the `nf_stat` utility *without* the –m option displays information about the various `SMT` state machines and the network to which the local station is attached:

`Ring` *(Ring Status)*

The `Ring` status shows the current state of the physical connection to the FDDI network. The following states may be returned by `nf_stat` under the `Ring` heading:

UP	SunFDDI interface physically connected to the active network
DOWN	SunFDDI interface disconnected, or connected to the inactive network

`ECM` *(Entity Coordination Management)*

`ECM` shows the current state of the Entity Coordination Management state machine, which controls the following features and facilities:

- Media availability
- Trace
- Path Test
- Optical Bypass (optional)
- Hold Policy (optional)

The following states may be returned by `nf_stat` under the `ECM` heading:

State	Meaning
OUT	`ECM` is inactive and is waiting for a connect request (initial state).
IN	`ECM` is active; normal state after successful connection request.
TRACE	`ECM` is propagating a trace request to the appropriate entity.
LEAVE	`ECM` is closing all connections prior to the station leaving the ring.
PATH_TEST	`ECM` is entering a path test state following trace completion.
INSERT	`ECM` is sending a request to the optical bypass switch to indicate that the station is entering the ring. This disengages the switch.
CHECK	`ECM` is verifying that symbols are being received from the network.
DEINSERT	`ECM` is sending a request to the optical bypass switch, to indicate that the station is leaving the ring. This engages the switch.

`RMT` *(Ring Management)*

`RMT` shows the current state of the Ring Management state machine, which controls the following features and facilities:

- `MAC` availability
- Detection and resolution of duplicate addresses
- Identification of stuck beacon and initiation of trace

The following states may be returned by `nf_stat` under the `RMT` heading:

State	Meaning
ISOLATED	`RMT` is inactive (initial state).
NON_OP	`RMT` is waiting for an operational ring.
RING_OP	`RMT` is operating normally.
DETECT	`RMT` is checking for duplicate addresses (transient state during initialization).
NON_OP_DUP	`RMT` has detected that its address is duplicated and is initiating recovery. The ring is not operational in this state.

RING_OP_DUP	RMT has detected that the MAC address is duplicated and flagged the error. The ring is operational in this state.
DIRECTED	RMT has been beaconing for an extended period of time and is transmitting a stream of directed beacons prior to initiating recovery.
TRACE	RMT has initiated a trace to recover a stuck beacon.

PCM *(Physical Connection Management)*

PCM shows the current state of the Physical Connection Management state machine that controls the following features and facilities:

* Connection initialization
* Maintenance support

This heading is modified to indicate the type of port that is being managed:

* PCMS: single-attached station, S-port
* PCMA: dual-attached station, A-port
* PCMB: dual-attached station, B-port

The following states may be returned by nf_stat under the PCM heading:

State	Meaning
OFF	PCM is inactive (initial state).
BREAK	PCM is starting the connection synchronization phase.
CONNECT	PCM is synchronizing the connection end-points prior to the signaling sequence.
NEXT	PCM is transmitting PDUs prior to entering SIGNAL state.
SIGNAL	PCM is transmitting and receiving signal bits (information) following a NEXT state.
JOIN	First state in the sequence leading to a synchronized connection.
VERIFY	Second state in the sequence leading to a synchronized connection.
ACTIVE	Final state indicating that the port is successfully incorporated in the token path.
TRACE	PCM is localizing a stuck beacon condition.

The normal sequence of `PCM` states leading to a fully synchronized connection and incorporation of the port into the token path is shown in Figure 9-1. Note that the minimum interval between interrogations is one second and that this is not always fast enough to recover and display the complete sequence of `PCM` states.
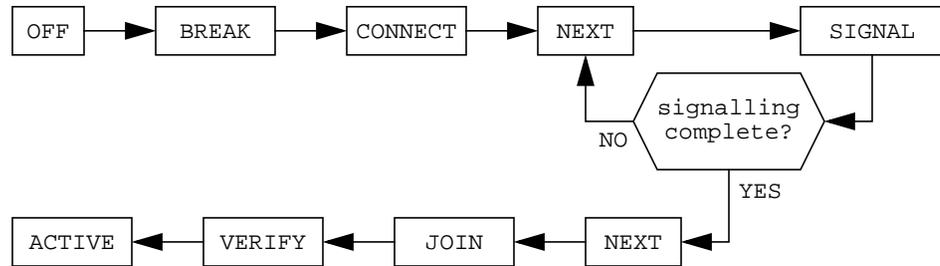
Figure 9-1   Normal Sequence of `PCM` States

`Ring_OP` *(Ring Operational)*

`Ring_OP` shows the number of `Ring_OP` (Ring Operational) signals received. This signal is generated when the station is incorporated into an operational network.

`XmitP` *(Transmit Packets)*

Running `nf_stat` *without* an interval and count, displays the total number of packets transmitted since the interface was activated. Running `nf_stat` *with* an interval and count, displays the number of packets transmitted since the last interrogation.

`RecvP` *(Receive Packets)*

Running `nf_stat` *without* an interval and count displays the total number of packets received since the interface was activated. Running `nf_stat` *with* an interval and count displays the number of packets received since the last interrogation.

Example Local Statistics

The following output was recovered from a single-attached station using the command shown. A temporary fault condition was simulated by disconnecting the FDDI cable from the SunFDDI SBus card and then reconnecting it.

Start nf_stat ---------->

Accumulated statistics ->

Path disconnected ------>

Path re-established ---->
(Ring_OP received)

Accumulated statistics ->

Finish nf_stat --------->

```
# <basedir>/nf_stat nf0 1 20
Ring    ECM       RMT       PCMS   Ring_OP    XmitP    RecvP
UP      IN    RING_OP    ACTIVE        2       26       1d
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        0        0
DOWN    IN   ISOLATED   CONNECT        0        1        1
DOWN    IN   ISOLATED   CONNECT        0        0        0
DOWN    IN   ISOLATED      NEXT        0        0        0
UP      IN    RING_OP    ACTIVE        1        0        0
UP      IN    RING_OP    ACTIVE        0        1        1

Ring    ECM       RMT       PCMS   Ring_OP    XmitP    RecvP
UP      IN    RING_OP    ACTIVE        3       29       20
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        1        1
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        0        0
UP      IN    RING_OP    ACTIVE        0        1        1
```

Note the following observations regarding this example:

- Accumulated statistics are displayed automatically after every tenth interrogation.
- The combination of Ring=DOWN and RMT=ISOLATED indicate that the station is disconnected from the network.
- The minimum interval of one second is not fast enough to recover and display the complete sequence of PCM states during the path re-establishment phase.
- A Ring_OP signal is received when the path is re-established indicating that the ring is operational.

The link status indicator mounted on the SunFDDI SBus card displays the following sequence of events:

GREEN (connected) ---> **AMBER** (disconnected) ---> **GREEN** (connected)

Displaying Statistics from Neighboring Stations

When you use the nf_stat command *with* the −m option, it displays information about the neighboring stations attached to the local interface nf*<inst>* and the frames received from the network.

For example, to display information about the neighboring stations attached to the interface nf0, type:

```
# <basedir>/nf_stat −m nf0
PhyS   Frame  Error   Lost        SA         UNA         DNA
   M  b43eb2      0      3    <mac_addr1>  <mac_addr2>  <mac_addr3>
```

You can also monitor the neighboring stations dynamically (active monitor), by specifying the *interval* (the elapsed time in seconds between interrogations) and *count* (the total number of interrogations). The minimum interval is one second and the accumulated statistics are displayed after every tenth interrogation.

For example, to monitor the stations attached to nf0 once every 10 seconds for 1 minute (a total of 6 interrogations), type:

```
# <basedir>/nf_stat −m nf0 10 6
PhyS    Frame  Error   Lost        SA         UNA         DNA
   M c460a6d      0      3    <mac_addr1>  <mac_addr2>  <mac_addr3>
   M   27224      0      0    <mac_addr1>  <mac_addr2>  <mac_addr3>
   M   27227      0      0    <mac_addr1>  <mac_addr2>  <mac_addr3>
   M   27220      0      0    <mac_addr1>  <mac_addr2>  <mac_addr3>
   M   2722e      0      0    <mac_addr1>  <mac_addr2>  <mac_addr3>
   M   27223      0      0    <mac_addr1>  <mac_addr2>  <mac_addr3>
```

Interpreting Statistics from Neighboring Stations

Running the nf_stat utility *with* the −m option displays information about the neighboring stations attached to the local interface nf*<inst>*.

`Phy` *(Physical Connection)*

PHY shows the type of physical connection to the FDDI network. This heading is modified to indicate the type of port being managed:

- `PhyS`: single-attached station, port S
- `PhyA`: dual-attached station, port A
- `PhyB`: dual-attached station, port B

The following states may be returned by `nf_stat` under the `Phy` heading:

State	Meaning
NONE	Port disconnected
M	Port connected to Port M on a concentrator
S	Port connected to Port S on a single-attached station
A	Port connected to Port A on a dual-attached station
B	Port connected to Port B on a dual-attached station

`Frame` *(Frames Received)*

Running `nf_stat` *without* an interval and count displays the total number of SMT frames received since the interface was activated. Running `nf_stat` *with* an interval and count displays the number of SMT frames received since the last interrogation.

More detailed information about the SMT frames can be recovered using the `nf_smtmon(1M)` utility described in "Monitoring SMT Frames (`nf_smtmon`)" on page 124.

`Error` *(Error Frames)*

Running `nf_stat` *without* an interval and count displays the total number of `error` frames received since the interface was activated. Running `nf_stat` *with* an interval and count displays the number of `error` frames received since the last interrogation. An `error` frame is defined as an SMT frame whose E (error) bit is set, and whose E bit is first detected by the local station. It does not indicate the location of the cause of the error. Frequent `error` frames can indicate a noise problem on the network, either dirt (optical fiber) or electrical interference (UTP).

`Lost` *(Lost Frames)*

Running `nf_stat` *without* an interval and count displays the total number of `lost` frames since the interface was activated. Running `nf_stat` *with* an interval and count displays the number of `lost` frames since the last interrogation. A `lost` frame is defined as an `SMT` frame whose reception is aborted by the `local` station. It does not indicate the location of the cause of the error. A large number of `lost` frames can indicate a noise problem on the network, either dirt (optical fiber) or electrical interference (UTP).

`SA` *(Station Address)*

Displays the `MAC` address for the local station.

`UNA` *(Upstream Neighbor Address)*

Displays the `MAC` address for the neighboring station, connected upstream on the ring from the local station.

`DNA` *(Downstream Neighbor Address)*

Displays the `MAC` address for the neighboring station, connected downstream on the ring from the local station.

Example Neighbor Statistics

The following output was recovered from a single-attached station using the command shown. A temporary fault condition was simulated by disconnecting the FDDI cable from the SunFDDI SBus card and then reconnecting it.

Start `nf_stat` ---------->
Accumulated statistics **->**

Path disconnected **------>**

Path re-established **---->**
(Lost frames indicated)

Accumulated statistics **->**

Finish `nf_stat` --------->

```
# <basedir>/nf_stat —m nf0 1 20
PhyS    Frame  Error   Lost      SA          UNA         DNA
   M  c45d5463     1     1b  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27437     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27427     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27435     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
NONE     182f1     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
NONE         0     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
NONE         0     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M      d432     0      7  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     2707e     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>

PhyS    Frame  Error   Lost      SA          UNA         DNA
   M  c46e5ce7     1     22  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27228     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27230     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27227     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     2722e     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     2722c     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27228     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27231     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     2722b     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
   M     27227     0      0  <mac_addr1>  <mac_addr2>  <mac_addr3>
```

Note the following observations regarding this example:

- Accumulated statistics are displayed automatically after every tenth interrogation.
- The combination of `PhyS=NONE` and the loss of frame activity indicates that the station is disconnected from the network.

Monitoring SMT Frames (`nf_smtmon`*)*

The `nf_smtmon(1M)` utility is an active monitor that displays the SMT frames received by the local station. It is particularly useful for diagnosing communication problems with the SunNet Manager proxy agent.

This command must be executed as `root` (or `superuser`) and has the general form:

```
# <basedir>/nf_smtmon [-i nf<inst>] [-x] [-h] [<frameclass>]
```

−i nf*<inst>* specifies the SunFDDI interface

−x displays the received SMT frames in hexadecimal

−h displays help information, including a list of valid frame classes

<frameclass> specifies one or more SMT frame classes (used to filter output)

If you do not specify an interface, `nf_smtmon` returns the SMT frames received by `nf0`. If you do not specify a frame type, `nf_smtmon` displays all the SMT frames that it receives. Use `Ctrl-C` to stop `nf_smtmon`.

To display the encoded SMT frames received by interface `nf1`, type:

```
# <basedir>/nf_smtmon -i nf1
nf1: nif_request  v=0x1 t=0xfc03e781 s=10-0-4-48-6f-a5 i=0x28
nf1: nif_response v=0x1 t=0xfc03e781 s=10-0-4-8-24-5c  i=0x28
nf1: nif_request  v=0x1 t=0xfc00dec6 s=10-0-4-b8-6e-ab i=0x28
nf1: nif_request  v=0x1 t=0xfc03e787 s=10-0-4-48-6f-a5 i=0x28
nf1: nif_response v=0x1 t=0xfc03e787 s=10-0-4-8-24-5c  i=0x28
```

The elements of the SMT frames are defined as follows:

class_type	Identifies the SMT frame class and type (see page 126)
v	Version ID; identifies the structure of the SMT information field
t	Transaction ID; used to pair SMT response and request frames
s	Station ID; uniquely identifies the station transmitting the frame
i	InfoField Length; defines the length of the SMT information field

♦ **To display the** SMT **frames received by interface** nf1 **in hexadecimal format, type:**

```
# <basedir>/nf_smtmon -i nf1 —x
nf1: nif_request v=0x1 t=0x170 s=10-0-4-8-24-5c i=0x28
004DC000 0000004F FFFFFFFF FFFF1000 0408245C 01020001 00000170 00001000
0408245C 00000028 00010008 00001000 04B86EAB 00020004 00010100 00030004
00002100 200B0008 00000001 00000001 76C467A0

nf1: nif_request v=0x1 t=0x5e0f s=10-0-d4-78-42-4d i=0x28
004D0000 0000004F FFFFFFFF FFFF1000 D478424D 01020001 00005E0F 00001000
D478424D 00000028 00010008 00001000 0408245C 00020004 01010208 00030004
00001200 200B0008 0000000B 00000002 A522BBA1

nf1: nif_response v=0x1 t=0xfc00d94a s=10-0-4-8-24-5c i=0x28
004D0000 00000041 100004B8 6EAB1000 0408245C 01030001 FC00D94A 00001000
0408245C 00000028 00010008 00001000 04B86EAB 00020004 00010100 00030004
00002100 200B0008 00000001 00000001 865549E2 0049C020 F0154E4F FFFFFFFF
FFFF1000 04B86EAB 01020001 FC00D94A 00001000 04B86EAB 00000028 00010008
00001000 D478424D 00020004 00010100 00030004 00002000 200B0008 00000001
00000001

nf1: nif_request v=0x1 t=0x5e13 s=10-0-d4-78-42-4d i=0x28
004D0000 0000004F FFFFFFFF FFFF1000 D478424D 01020001 00005E13 00001000
D478424D 00000028 00010008 00001000 0408245C 00020004 01010208 00030004
00001200 200B0008 0000000B 00000002 4AD75A79

nf1: nif_request v=0x1 t=0x5e17 s=10-0-d4-78-42-4d i=0x28
004D0000 0000004F FFFFFFFF FFFF1000 D478424D 01020001 00005E17 00001000
D478424D 00000028 00010008 00001000 0408245C 00020004 01010208 00030004
00001200 200B0008 0000000B 00000002 DCEBADA2

nf1: nif_request v=0x1 t=0x171 s=10-0-4-8-24-5c i=0x28
004DC000 0000004F FFFFFFFF FFFF1000 0408245C 01020001 00000171 00001000
0408245C 00000028 00010008 00001000 04B86EAB 00020004 00010100 00030004
00002100 200B0008 00000001 00000001 127B1D3B

nf1: nif_request v=0x1 t=0x5e1b s=10-0-d4-78-42-4d i=0x28
004D0000 0000004F FFFFFFFF FFFF1000 D478424D 01020001 00005E1B 00001000
D478424D 00000028 00010008 00001000 0408245C 00020004 01010208 00030004
00001200 200B0008 0000000B 00000002 626FA878
```

SMT Frame Classes and Types

SMT frames are used for peer-to-peer (station-to-station) management. They are divided into *classes*, which define the function of the frame. Each class is then divided into up to three *types*, which define whether the frame is an announcement (information only), a request for service, or a response to a request. Refer to the *ANSI/FDDI Station Management (SMT) X3.299 R7.3 Specification* for a detailed description of SMT frames and their functions.

The nf_smtmon utility is used to monitor the following SMT frame classes:

NIF *(Neighbor Information Frames)*

These are the most common frames displayed when you run nf_smtmon. As the name suggests, they carry information about a neighboring station (for example, address, description, state, MAC capabilities) and are used as keep-alive notifications that a station is still attached to the ring and functioning. An NIF frame can be an announcement, a request, or a response.

SIF *(Status Information Frames)*

These frames carry more detailed information about a station. SIF configuration frames describe the station configuration (for example, number of ports, number of MAC entities, connection policy); SIF operation frames describe the current status of the station. A SIF frame can be either a request or a response.

ECF *(Echo Frames)*

These frames are equivalent to ICMP ping packets and are used to test connectivity between stations. An ECF frame can be either a request or a response.

RDF *(Request Denied Frame)*

These frames are used to indicate that the request is rejected. If an SMT agent (such as the SunNet Manager proxy agent delivered with SunFDDI) receives an unsupported or unrecognized request, it issues an RDF frame to indicate that the request is rejected. An RDF frame is always a response.

ESF *(Extended Service Frame)*

These frames are implementation dependent. An ESF frame can be an announcement, a request, or a response.

PMF *(Parameter Management Frame)*

These frames are used to access remote station attributes. The Parameter Management Protocol supports both get (display) and set (modify) functions; however, the nf_smtmon utility can display only PMF_get frames. A PMF_get frame can be either a request or a response.

Filtering Output from nf_smtmon

By default, nf_smtmon displays all of the SMT frames received by the local station. You can filter the output generated by nf_smtmon by specifying one or more frame classes on the command-line: nif, sif_config, sif_operat, ecf, rdf, esf, pmf_get.

For example:

♦ **To display only the** RDF **frames received by interface** nf0, **type:**

```
# <basedir>/nf_smtmon rdf
```

♦ **To display the** SIF **configuration and** SIF **operation frames received by interface** nf1, **type:**

```
# <basedir>/nf_smtmon −i nf1 sif_config sif_operat
```

♦ **To display the** NIF **frames,** SIF **configuration frames, and** SIF **operation frames received by interface** nf1, **type:**

```
# <basedir>/nf_smtmon −i nf1 nif sif_config sif_operat
```

Configuring Synchronous FDDI (`nf_sync`*)*

Note – Synchronous FDDI, configured using the `nf_sync(1M)` utility, is only supported when operating in a Solaris 2.*x* environment.

Two types of traffic are allocated bandwidth on an FDDI ring:

* Asynchronous traffic (unrestricted and restricted)
* Synchronous traffic

The unrestricted asynchronous service allocates bandwidth dynamically based on the *timed token rotation protocol* and according to the *priority* assigned to the traffic. It does not guarantee bandwidth.

The restricted asynchronous service allocates bandwidth for extended transmission based on *restricted token exchange* between communicating stations.

For synchronous operation, a specific portion of the bandwidth is reserved for synchronous traffic and a boundary is placed on the access delay (the time that a station must wait before it can transmit). The synchronous service is normally used to transfer voice, video, and real-time control data, where the access delay can have significant impact. Note that it is the network user (the application) that requests the synchronous service and manages synchronous operation. The FDDI protocols do not provide any mechanism for controlling synchronous allocation directly.

By default, the SunFDDI interface is configured to carry asynchronous traffic only. The `nf_sync(1M)` utility is used to configure SunFDDI interfaces operating in a Solaris 2.*x* environment to operate in synchronous mode. This command must be executed as `root` (or `superuser`) and has the general form:

```
# nf<inst> [ <tsync> <sap> ]
```

nf*<inst>* specifies the SunFDDI interface

<tsync> synchronous timer (in nanoseconds, 400000 nanoseconds minimum)

<sap> service access point (SAP) for synchronous operation

Running `nf_sync` without specifying values for *<tsync>* and *<sap>* returns the current configuration of the interface.

♦ **To display the current configuration of the interface** `nf0`**, type:**

```
# <basedir>/nf_sync nf0
Current Bound Saps
 Async: sap=0x800 : sap=0x806
 Sync:
```

This example shows that the interface `nf0` has two SAPs (`800` and `806`) configured for asynchronous operation and no SAPs configured for synchronous operation. This is the default condition.

♦ **To display the current configuration of the interface** `nf1`**, type:**

```
# <basedir>/nf_sync nf1
Current Bound Saps
 Async: sap=0x806
 Sync: sap=800 (tsync=10001040ns)
```

This example shows that the interface `nf1` has one SAP (`800`) configured for synchronous operation with a clock rate of 1000000 nanoseconds (1 ms), and one SAP (`806`) configured for asynchronous operation.

♦ **To configure the interface** `nf1` **in this way, type:**

```
# <basedir>/nf_sync nf1 1000000 800
```

The required value of *<tsync>* is application dependent; however, you must enter a value that is large enough to transmit at least one full FDDI packet (~0.333 ms). Note that *<tsync>* defaults to the maximum `TTRT` (`T_MAX`) for SunFDDI (167.874 ms) if you enter a value less than 400000 nanoseconds (0.4ms)

♦ **To reconfigure a SAP for asynchronous operation, specify** *<tsync>*=0:

```
# <basedir>/nf_sync nf1 0 800
# <basedir>/nf_sync nf1 0 806
```

☰ *9*

_____

Managing FDDI Stations
Using SunNet Manager

Installing the SunNet Manager Agents	page 132
Managing FDDI Networks with SunNet Manager	page 139

This chapter describes how to configure and start the SunNet Manager agents for SunFDDI on your machine. It also tells you how to use a SunNet Manager Console to recover FDDI statistics from the agents.

To use the SunNet Manager agents for SunFDDI you must run SunNet Manager 2.2 or later, and you must install a SunNet Manager Console (or management application) on your network. For instructions on how to install and configure SunNet Manager on the management station, refer to the *SunNet Manager Installation Guide*.

All Sun-4d, Sun-4m and Sun-4u systems (and most other machines that connect to TCP/IP networks) support protocols that enable a SunNet Manager Console to recover basic information and network statistics. In addition, SunNet Manager provides a number of agents that are used to recover more detailed information from the network:

- SunNet Manager SNMP (simple network management protocol) agents are used to gather information and to set parameters on remote machines.

- SunNet Manager proxy agents are installed locally and used to recover information as if they were located on a remote machine.

▤ *10*

Installing the SunNet Manager Agents

You must install a SunNet Manager Console (or management application) on your network before installing and configuring the SunNet Manager agents for SunFDDI.

Two SunNet Manager agents are provided for SunFDDI.

- The **SunFDDI local agent** (`nf_fddi`) responds to SNM requests from the Console and returns statistics gathered on the client machine. These statistics are similar to those returned by `nf_stat(1M)`. See "Displaying SunFDDI Statistics (`nf_stat`)" on page 114 for detailed information.

- The **SunFDDI proxy agent** (`nf_fddismt7_2`) responds to SNM requests from the Console and issues SMT Request Frames (SRF) to a specified target station. It then returns the information received from the target station in response to the request. This information is similar to that returned by `nf_smtmon(1M)`. See "Monitoring SMT Frames (`nf_smtmon`)" on page 124 for detailed information.

▼ To Install the Standard SNM Agents on a Client

Although you can recover basic information from your FDDI station without installing the standard agents supplied with SunNet Manager, you can manage your network more completely if you install these agents on each machine. You must install the standard agents *before* installing the agents for SunFDDI.

There are two ways to install the SunNet Manager agents on the client machine:

- ♦ **Install the package that contains the SunNet Manager agents and libraries on the client machine.**
  This method ensures that the system information is updated correctly on Solaris 2.*x* machines and makes it easier to remove the agents and libraries, if required.

- ♦ **Mount the SunNet Manager directory tree from the Console machine and run** `getagents(8)` **to install the agents in a local directory.**
  This is a quick way to distribute the standard agents across a large network and is the only way to install SNM agents on Solaris 1.*x* clients.

Both methods are described in the *SunNet Manager Installation Guide.*

▼ To Install the FDDI Schemas on a Client (Solaris 2.*x*)

Use `nf_install_agents` to install the SunNet Manager agents for SunFDDI. This script copies the FDDI schema files to the directory in which the standard agents are installed and updates the configuration files for SunNet Manager. Run `nf_install_agents` on each SunNet Manager Console and each SunFDDI station you want to manage using SunNet Manager.

1. **Log in as** `root` **or become** `superuser`.

2. **Start** `nf_install_agents` **by typing:**

```
# /opt/SUNWconn/bin/nf_install_agents
```

3. **Enter the host names of Console machine(s) that you want to receive SRF trap reports from the local station.**
   You can send SRF trap reports to more than one Console. Each Console must have the schemas for SunFDDI installed.

```
Enter hostnames (if any) to send SRF Trap reports to:
(enter blank to terminate)
        enter hostname:opus
        enter hostname:rigel
        enter hostname:
```

Host name of Console --------->
Host name of Console --------->
Press Return to terminate ---->

4. **Specify the destination directory for the SunFDDI schemas.**
   SunNet Manager expects to find the schemas in the directory *<basedir>*`/SUNWconn/snm/agents`. The variable *<basedir>* is set to `/opt` by default. If you installed the SunNet Manager agents and libraries on the FDDI station, or if you used `getagents` to recover the agents from the Console, then this directory exists.

   a. Enter the base directory under which you installed the SunNet Manager agents as the destination directory for the SunFDDI schemas:

```
What is the name of the root directory under which the SunNet
Manager is installed? [/opt]: [?]
```

5. Specify the current location of the SunFDDI schemas.
This is the base directory under which you installed the SunFDDI software
package (`SUNWnf`) using `pkgadd(1M)`.

```
What is the name of the root directory under which the nf FDDI
software is installed? [/opt]: [?]
```

The SunFDDI schemas are copied to the destination directory and the local
configuration files are modified for SunNet Manager.

6. Start the SunNet Manager agents for SunFDDI by typing:

```
# /sbin/nf_snmd
```

Note – If you typed `n` in response to the question, "Do you want to start the
SunNet Manager daemons for SunFDDI at boot time?", when you installed the
SunFDDI software, you must start the SunNet manager agents for SunFDDI
manually each time you reboot your machine. Alternatively, you can install the
start-up script into the `/etc/rc2.d` directory:

```
# cd /opt/SUNWconn/SUNWnf/utilities
# cp nf_fddidaemon /etc/rc2.d/S98nf_fddidaemon
```

▼ To Install the FDDI Schemas on a Client (Solaris 1.*x*)

Use `nf_install_agents` to install the SunNet Manager agents for SunFDDI.
This script copies the FDDI schema files to the directory in which the standard
agents are installed and updates the configuration files for SunNet Manager.
Run `nf_install_agents` on each SunNet Manager Console and each
SunFDDI station you want to manage using SunNet Manager.

1. Log in as `root` or become `superuser`.

2. Start `nf_install_agents` by typing:

```
# /usr/sunlink/nf/install/nf_install_agents
```

3. **Enter the host names of Console machines that will receive SRF trap reports when an event occurs at the local station.**
   You can send SRF trap reports to more than one Console. Each Console must have the schemas for SunFDDI installed on it.

```
Enter hostnames (if any) to send SRF Trap reports to:
(enter blank to terminate)
          enter hostname:opus
          enter hostname:rigel
          enter hostname:
```

Host name of Console -------->
Host name of Console -------->
Press Return to terminate ---->

4. **Specify the destination directory for the SunFDDI schemas.**
   SunNet Manager expects to find the schemas in the directory *<basedir>*/sunlink/nf/sys/snm/agents. The variable *<basedir>* is set to /usr by default. If you installed the SunNet Manager agents and libraries on the client, or if you used getagents to recover the agents from the Console, then this directory exists on the client machine.

   Enter the base directory under which you installed the SunNet Manager agents as the destination directory for the SunFDDI schemas.

```
What is the name of the root directory under which the SunNet
Manager is installed? [/usr]: [?]
```

5. **Specify the current location of the SunFDDI schemas.**
   This is the base directory under which you installed the SunFDDI software using cdm.

```
What is the name of the root directory under which the nf FDDI
software is installed? [/usr]: [?]
```

   The SunFDDI schemas are copied to the destination directory and the local configuration files are modified for SunNet Manager.

6. **Start the SunNet Manager agents for SunFDDI by typing:**

```
# /usr/etc/nf_snmd
```

Note – If you typed n in response to the question, "Do you want to start the SunNet Manager daemons for SunFDDI at boot time?", when you installed the SunFDDI software, you must start the SunNet manager agents for SunFDDI manually each time you reboot your machine. Alternatively, you can add the following lines to the file /etc/rc.local:

```
if [ -f /usr/etc/nf_snmd ]; then
        /usr/etc/nf_snmd
fi
```

▼ To Install the FDDI Schemas on the Console

If you install the SunFDDI software on the Console machine, you can run nf_install_agents locally. See "To Install the FDDI Schemas on a Client (Solaris 2.x)" on page 133 for detailed instructions.

If you do not have a SunFDDI SBus card installed in the Console machine, you must either copy the script and SunFDDI schemas onto the Console machine or NFS mount the nf_install_agents script from a client machine and execute it remotely.

▼ To share nf_install_agents from a Solaris 2.*x* client:

1. **On the client machine, ensure that the NFS and mount daemons are running:**

```
# /usr/lib/nfs/nfs 8
# /usr/lib/nfs/mountd
```

2. **On the client machine, share the** /opt **directory under which you installed SunFDDI.**

```
# share -F nfs -o ro /opt
```

3. On the Console machine, NFS mount the `/opt` **directory:**

```
# mount <client>:/opt /<mnt_dr>
```

4. On the Console machine, run `nf_install_agents`**:**

```
# /mnt/SUNWconn/bin/nf_install_agents
```

Follow the directions in "To Install the FDDI Schemas on a Client (Solaris 2.x)" on page 133, and specify *<mnt_dir>* as the current location of the SunFDDI schemas.

- On a Solaris 2.*x* client, the SunFDDI schemas are located in the directory `/opt/SUNWconn/SUNWnf/agents` and the `nf_install_agents` script is located in the directory `/opt/SUNWconn/SUNWnf/utilities`.

- On a Solaris 1.*x* client, the SunFDDI schemas are located in the directory `/usr/sunlink/nf/snm/agents` and the `nf_install_agents` script is located in the directory `/usr/sunlink/nf/install`.

- On the Console machine, the schemas must be installed in the directory
  /opt/SUNWconn/snm/agents. The nf_install_agents script modifies
  the file /opt/SUNWconn/snm/struct/elements.schema to add the
  definition for the host name or MAC address of an FDDI station. If you do
  not run nf_install_agents on the Console machine, you must modify
  this file to add the following line to each component (workstation, server, or
  router) definition:

```
string[40]      Name/Mac-Address
```

For example, if you want to specify the host name or MAC address of an FDDI
interface installed in an Ultra system, add the following line to
elements.schema:

```
#
# Sun workstations
#

record component.ultra1 (                    # ultrasparcstation1
        string[64]     Name
        string[40]     IP_Address
        string[40]     User
        string[40]     Location
        string[80]     Description
        string[40]     SNMP_RdCommunity
        string[40]     SNMP_WrCommunity
        string[64]     SNMP_Vendor_Proxy
        int            SNMP_Timeout
        string[40]     Name/Mac-Address
```

Add this line -------->

Managing FDDI Networks with SunNet Manager

For detailed information on how to use SunNet Manager to manage the various components of your network, refer to the *SunNet Manager User's Guide.*

The SunNet Manager agents for SunFDDI are used to recover FDDI statistics and SMT frame information. For detailed instructions on how to interpret this data, refer to the ANSI/FDDI standards listed in "FDDI Conformance" on page 2.

▼ To Set Up the SunNet Manager Console

1. **Log in as** `root` **or become** `superuser`.

2. **Start the SunNet Manager Console, and initialize the runtime management database (MDB).**

```
# /opt/SUNWconn/bin/snm —i
```

3. **Use HeadStart to begin discovering the nodes in your network, or use BasicStart to build your own database with the Console's graphical editor.**
   Note that the Discover facility will only fully recognize the Ethernet (`le`) interfaces in your network.

4. **Modify the component properties of each SunFDDI station in your network.**
   Use the scroll bar to show all of the properties associated with the station. You must enter either the host name or the MAC address associated with the SunFDDI interface in the last field in the list of properties. If you enter a host name, it must be registered in the NIS/NIS+ map or in the `/etc/hosts` file on each machine in your network. If you enter a MAC address, it must be entered in ring format.

5. **Click SELECT on the check box associated with each SunNet Manager agent that you want to communicate with the station.**
   If you enable the SunFDDI proxy agent (`nf_fddismt7_2`), you must also specify the host name (or MAC address) of the target FDDI station from which the proxy agent will request information.

▼ **Use the SunFDDI Local Agent (`nf_fddi`)**

The SunFDDI local agent (`nf_fddi`) responds to SNM requests from the
Console and returns statistics gathered on the client machine. These statistics
are similar to those returned by `nf_stat(1M)`. See "Displaying SunFDDI
Statistics (`nf_stat`)" on page 114 for detailed information.

▼ **To Gather Local Station Statistics**

Use the SunFDDI local agent to recover a snapshot of the current state of the
local FDDI station:

1. **Click SELECT on the icon for the FDDI station that you want to
   interrogate.**

2. **Press MENU on the Requests menu button and release MENU on the
   Quick Dump item.**

3. **Choose `nf_fddi` as the Agent Schema and `if` (interface) as the Group.**

4. **Click SELECT on Apply to display local station statistics similar to those
   shown in Figure 10-1.**

```
┌────────────────────────────────────────────────────────────────────┐
│         SunNet Manager Console: Quick Dump Report – arbois           │
├────────────────────────────────────────────────────────────────────┤
│ Mon Apr 18 21:51:35 1994 [ arbois ] : Quick Dump: fddi.if            │
│                                                                      │
│   ifname=nf0                                                         │
│   Ring_State=1                                                       │
│   Ring=Up                                                            │
│   ECM_State=1                                                        │
│   ECM=In                                                             │
│   ECM_Status=Connected                                               │
│   OBP_State=0                                                        │
│   OBP=None                                                           │
│   RMT_State=2                                                        │
│   RMT=Ring_Op                                                        │
│   CFM_State=12                                                       │
│   CFM=Thru                                                           │
│   PCM_A_State=8                                                      │
│   PCM_A=Active                                                       │
│   PCM_B_State=8                                                      │
│   PCM_B=Active                                                       │
│   Recovery=35                                                        │
│   XmitP=3267652                                                      │
│   XmitB=2595625496                                                   │
│   RecvP=3057228                                                      │
│   RecvB=946139565                                                    │
│   RFull=1                                                            │
│   RErr=1                                                             │
│   XAbort=0                                                           │
└────────────────────────────────────────────────────────────────────┘
```
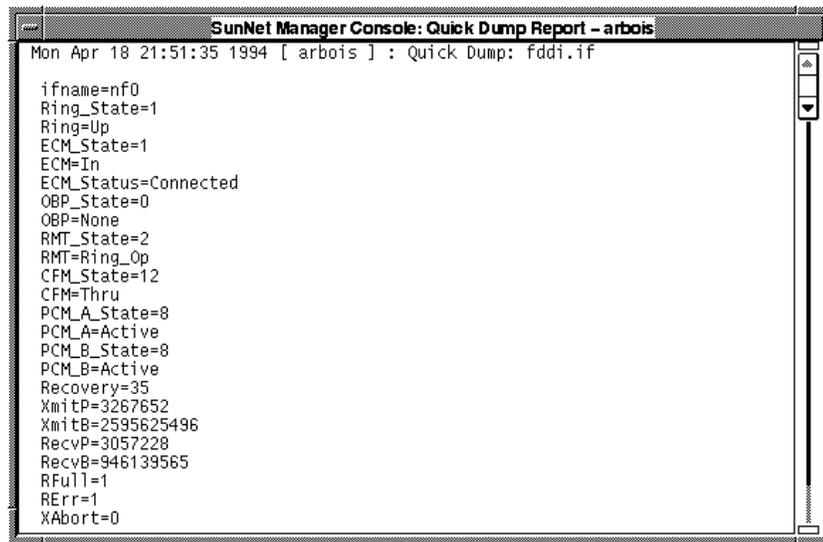
Figure 10-1 Quick Dump (Local Statistics)

▼ To Gather Neighbor Station Statistics

Use the **SunFDDI local agent** to recover a snapshot of the current state of the neighboring FDDI stations:

1. **Click SELECT on the icon for the FDDI station that you want to interrogate.**

2. **Press MENU on the Requests menu button and release MENU on the Quick Dump item.**

3. **Choose** `nf_fddi` **as the Agent Schema and** `route` **as the Group.**

4. **Click SELECT on Apply to display neighbor station statistics similar to those shown in Figure 10-2.**
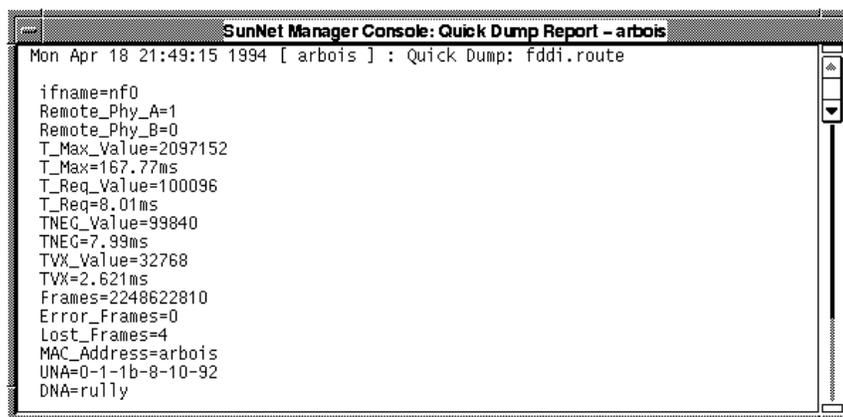
```
SunNet Manager Console: Quick Dump Report – arbois
Mon Apr 18 21:49:15 1994 [ arbois ] : Quick Dump: fddi.route

 ifname=nf0
 Remote_Phy_A=1
 Remote_Phy_B=0
 T_Max_Value=2097152
 T_Max=167.77ms
 T_Req_Value=100096
 T_Req=8.01ms
 TNEG_Value=99840
 TNEG=7.99ms
 TVX_Value=32768
 TVX=2.621ms
 Frames=2248622810
 Error_Frames=0
 Lost_Frames=4
 MAC_Address=arbois
 UNA=0-1-1b-8-10-92
 DNA=rully
```

Figure 10-2   Quick Dump (Neighbor Statistics)

▼ To Request Information from the Local Agent

♦ **Use the Console to create a request for data that is sent periodically to the SunFDDI local agent.**
This is similar to running the `nf_stat(1M)` utility and specifying an *<interval>* and *<count>* as described in "Displaying Local Interface Statistics" on page 114.

The local agent responds with the requested information, which can be saved to file or displayed using the SunNet Manager Grapher.

▼ **To Use the SunFDDI Proxy Agent (**`nf_fddismt7_2`**)**

The **SunFDDI proxy agent** (`nf_fddismt7_2`) responds to SNM requests from the Console and issues SMT Request Frames (SRF) to a specified target station. It then returns the information received from the target station in response to the request. The basic operation of the SunFDDI proxy agent is shown in Figure 10-3.
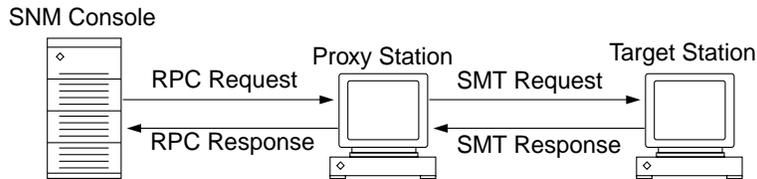

Figure 10-3   SunFDDI Proxy Agent

▼ **To Specify the Target Station**

1. **Specify the target machine when you enable the SunFDDI proxy agent in the properties window for the proxy station.**

2. **Use the scroll bar to display the two SunFDDI agents as shown in Figure 10-4 and enter the host name of the target station on the input line.**
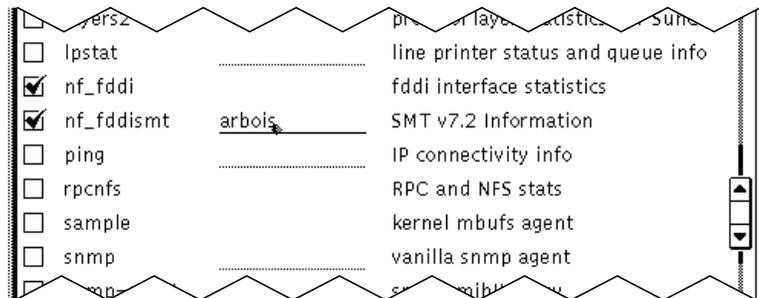

Figure 10-4   Specifying the Target Station

▼ To Gather Information from the Proxy Agent

Use the SunFDDI proxy agent to recover SMT information from one of the neighboring FDDI stations:

1. Click SELECT on the icon for the FDDI proxy station that you want to interrogate.

2. Press MENU on the Requests menu button and release MENU on the Quick Dump item.

3. Choose `nf_fddismt7_2` **as the Agent Schema and one of the SMT frame types or FDDI MIBs as the Group.**

4. Click SELECT on Apply to display SMT information similar to that shown in Figure 10-5.
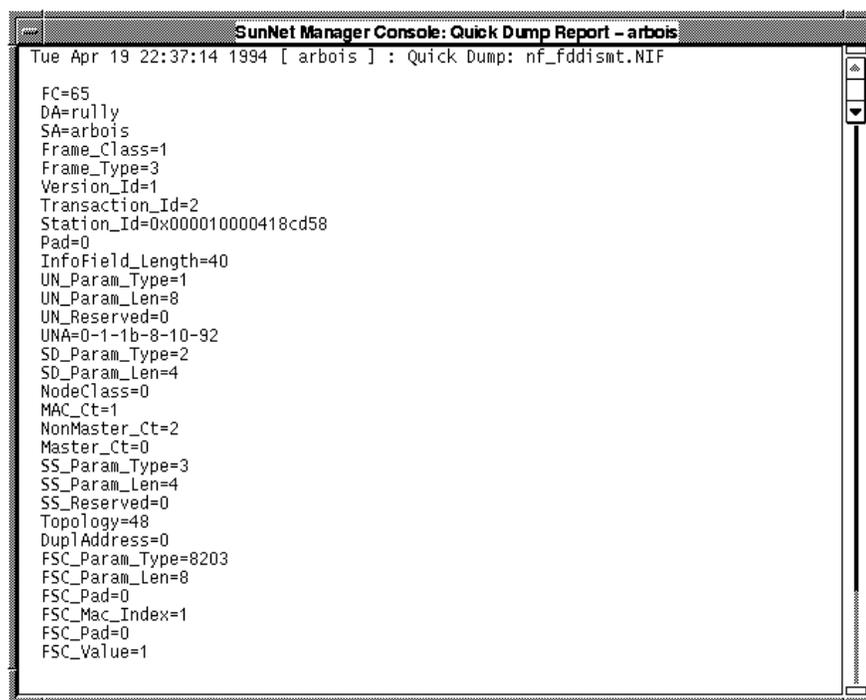
```
SunNet Manager Console: Quick Dump Report – arbois
Tue Apr 19 22:37:14 1994 [ arbois ] : Quick Dump: nf_fddismt.NIF

FC=65
DA=rully
SA=arbois
Frame_Class=1
Frame_Type=3
Version_Id=1
Transaction_Id=2
Station_Id=0x000010000418cd58
Pad=0
InfoField_Length=40
UN_Param_Type=1
UN_Param_Len=8
UN_Reserved=0
UNA=0-1-1b-8-10-92
SD_Param_Type=2
SD_Param_Len=4
NodeClass=0
MAC_Ct=1
NonMaster_Ct=2
Master_Ct=0
SS_Param_Type=3
SS_Param_Len=4
SS_Reserved=0
Topology=48
DuplAddress=0
FSC_Param_Type=8203
FSC_Param_Len=8
FSC_Pad=0
FSC_Mac_Index=1
FSC_Pad=0
FSC_Value=1
```

Figure 10-5 Quick Dump (SMT Frames)

Interpreting Traps and Errors

One of the most important functions of the SunNet Manager agent for
SunFDDI is to signal the occurrence of faults or errors on the network.
Traps are sent to the Consoles that you specified when you ran the
`nf_install_agents` script to install the agents on the client machine. See "To
Install the FDDI Schemas on a Client (Solaris 2.x)" on page 133 for detailed
instructions.

When a cable or station fault occurs, it is signaled by the neighboring station
and not by the station implicated by the fault; however, the information
provided in the trap report sent to the Console allows you to identify the
probable source of the problem. Faults often generate a series of trap events
that are caused when the ring reconfigures itself automatically.

The reason for repeated errors and lost frames is more difficult to identify
because the fault is signaled by the first system to receive the error frame,
which is not necessarily the station closest to the problem.

The trap report in Figure 10-6 was generated by the upstream neighbor
after the cable between a single-attached station and a concentrator was
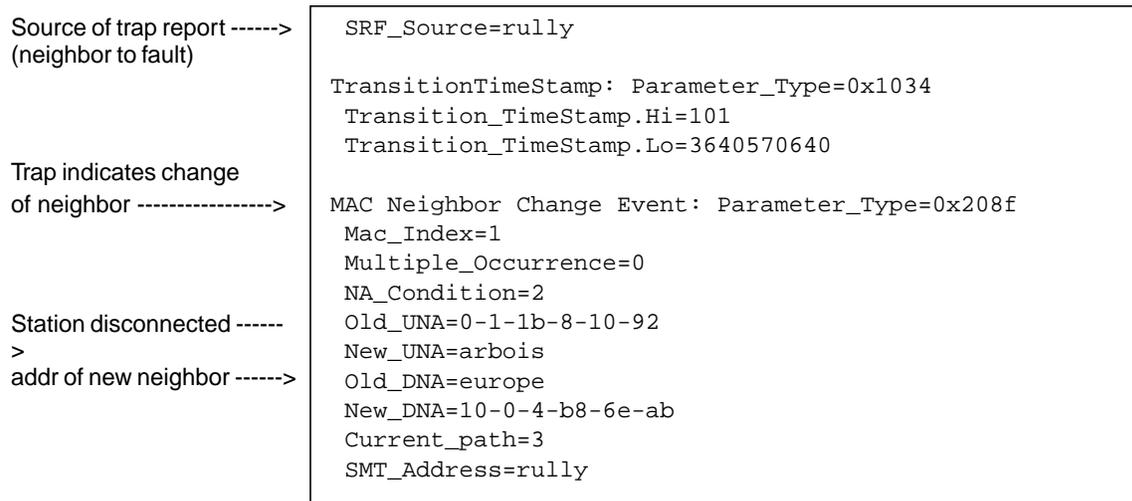disconnected. All MAC addresses are displayed in *ring* format.

```
Source of trap report ------>    SRF_Source=rully
(neighbor to fault)

                               TransitionTimeStamp: Parameter_Type=0x1034
                                 Transition_TimeStamp.Hi=101
                                 Transition_TimeStamp.Lo=3640570640

Trap indicates change
of neighbor ---------------->   MAC Neighbor Change Event: Parameter_Type=0x208f
                                 Mac_Index=1
                                 Multiple_Occurrence=0
                                 NA_Condition=2
Station disconnected ------      Old_UNA=0-1-1b-8-10-92
>                                New_UNA=arbois
addr of new neighbor ------>     Old_DNA=europe
                                 New_DNA=10-0-4-b8-6e-ab
                                 Current_path=3
                                 SMT_Address=rully
```

Figure 10-6   Single-Attached Station Disconnected

The trap reports in Figure 10-7 were generated when a single-attached station was reconnected to the ring.

Trap report generated by the upstream station

Source of trap report ------>
(neighbor to fault)

Trap indicates change
of neighbor ---------------->

Current neighbor --------->
Old neighbor returns ------>

```
 SRF_Source=rully


TransitionTimeStamp: Parameter_Type=0x1034
 Transition_TimeStamp.Hi=102
 Transition_TimeStamp.Lo=845593304


MAC Neighbor Change Event: Parameter_Type=0x208f
 Mac_Index=1
 Multiple_Occurrence=0
 NA_Condition=2
 Old_UNA=0-1-1b-8-10-92
 New_UNA=arbois
 Old_DNA=10-0-4-b8-6e-ab
 New_DNA=europe
 Current_path=3
 SMT_Address=rully
```

Trap report generated by the reconnected station

Source of trap report ------>
(reconnected station)

Trap indicates error
condition ------------------>

Frames lost during
fault condition ------------->

```
 SRF_Source=europe

 TransitionTimeStamp: Parameter_Type=0x1034
 Transition_TimeStamp.Hi=1
 Transition_TimeStamp.Lo=2482471160

 MAC Frame Error Condition: Parameter_Type=0x208d
 Mac_Index=1
 Condition_State=0
 Frame_Ct=101367280
 Error_Ct=0
 Lost_Ct=8
 Frame_Error_Ratio=0
```

Figure 10-7 Single-Attached Station Reconnected

The trap report in Figure 10-8 was generated by the upstream neighbor when a dual-attached station connected directly to the FDDI ring was halted. The neighbor wraps itself around the fault automatically.
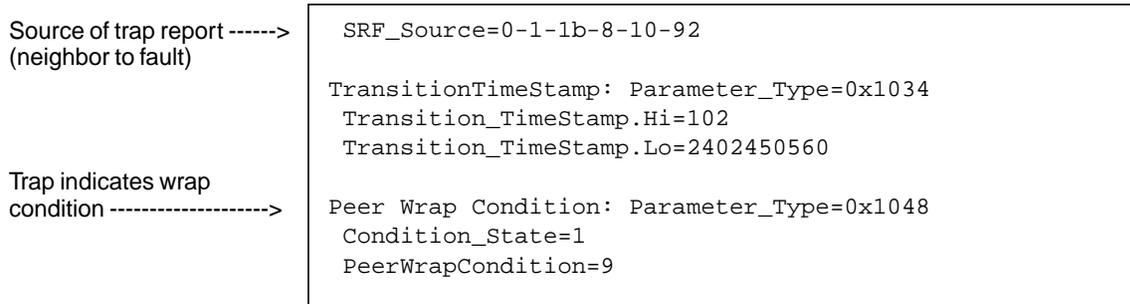
```
Source of trap report ------>     SRF_Source=0-1-1b-8-10-92
(neighbor to fault)

                                TransitionTimeStamp: Parameter_Type=0x1034
                                 Transition_TimeStamp.Hi=102
                                 Transition_TimeStamp.Lo=2402450560

Trap indicates wrap
condition ------------------->  Peer Wrap Condition: Parameter_Type=0x1048
                                 Condition_State=1
                                 PeerWrapCondition=9
```

Figure 10-8   Dual-Attached Station Halted

The trap report in Figure 10-9 was generated by the upstream neighbor when a dual-attached station connected directly to the FDDI ring was restarted. It shows a change of neighbor event that occurs as the wrapped station rediscovers the neighbors to which it was connected before the fault.

```
Source of trap report ------>     SRF_Source=rully
(reconnected station)

                                TransitionTimeStamp: Parameter_Type=0x1034
                                 Transition_TimeStamp.Hi=102
                                 Transition_TimeStamp.Lo=3714602392

Trap indicates change
of neighbor ---------------->   MAC Neighbor Change Event: Parameter_Type=0x208f
                                 Mac_Index=1
                                 Multiple_Occurrence=0
                                 NA_Condition=1
                                 Old_UNA=0-1-1b-8-10-92
Rediscover neighbor ------>      New_UNA=arbois
                                 Old_DNA=10-0-4-b8-6e-ab
Rediscover neighbor ------>      New_DNA=europe
                                 Current_path=3
                                 SMT_Address=rully
```

Figure 10-9   Dual-Attached Station Restarted

Mac Address Formats

The MAC address can be represented in one of two formats:

- Canonical format
- Ring format

The MAC address is always displayed in canonical format by the SunFDDI network utilities and hardware self-test program. It is also displayed in canonical format when you reboot your machine.

The SMT information displayed by SunNet Manager always shows the MAC address in ring format. If you specify a target station using a MAC address (instead of a host name), you must enter it in ring format.

■ 10

Developing Applications over SunFDDI

11

This chapter describes how to develop your own applications running over SunFDDI networks. It assumes that you are an experienced programmer who is already familiar with the DLPI Version 2 interface (Solaris 2.*x*) or the NIT monitor (Solaris 1.*x*).

This chapter provides a brief overview of each of the programming environments and concentrates on network addressing and other issues that are specific to the SunFDDI implementation of these interfaces. It does not provide a detailed specification of the interfaces.

DLPI is used to access the services of the SunFDDI link layer of the SunFDDI device driver in a Solaris 2.*x* environment without the need for specialized knowledge of the underlying protocol. You can obtain a copy of the *STREAMS DLPI Specification* from Sun Microsystems, Inc. (part number 800-6915-10).

NIT is used to monitor and recover information from the `ifnet` interface to the link layer of the SunFDDI device driver in a Solaris 1.*x* environment.

Developing Applications with DLPI (Solaris 2.x)

The Data Link Provider Interface (DLPI) Version 2 enables any data link user to access the services of a data link provider without any special knowledge about the underlying protocols. In addition to FDDI services, DLPI supports access to X.25, LAPB, LAPD, ISDN, Ethernet, Token Ring, Token Bus, SDLC, BISYNC, and other data link protocols.

The DLPI header file provided in the SunOS 5.*x* operating system is `<sys/dlpi.h>`. A detailed description of the functions contained in the header is available in the *STREAMS DLPI Specification.*

DLPI defines a STREAMS message set, state table, and conventions that support interaction between a data link service user (for example, IP or ARP) and a data link service provider (for example, the `nf` FDDI driver). Figure 11-1 illustrates the relationship between DLPI and the layers of the FDDI protocol.

OSI	TCP/IP	Data Link Service Users
Data Link Provider Interface (DLPI) V2		
LLC1	LLC/SNAP	Data Link Service Provider
Media Access Control (MAC) Layer		
Physical (PHY) Layer		
Physical Media Dependent (PMD) Layer		

Station Management (SMT)
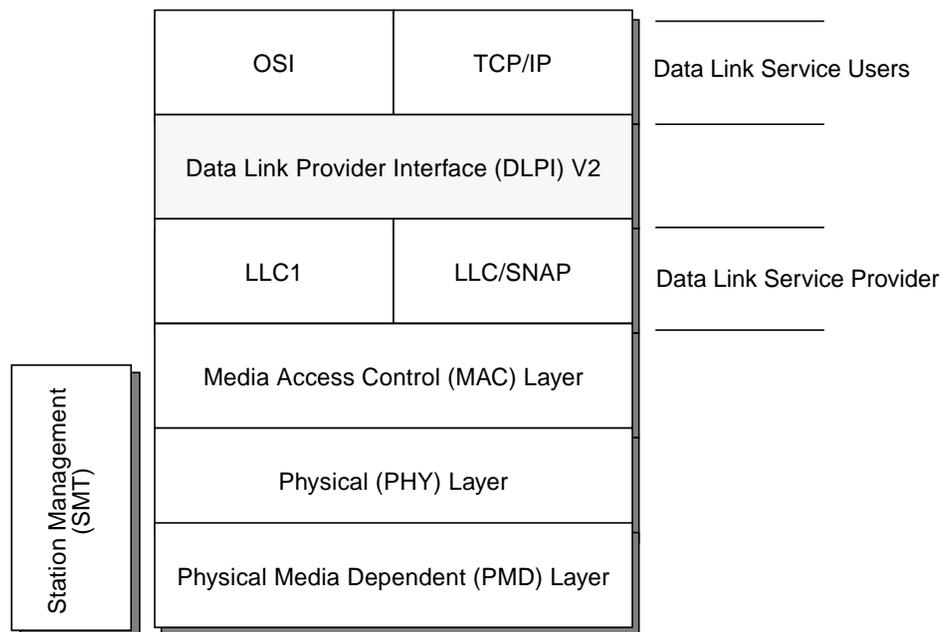
Figure 11-1   DLPI Access to FDDI

DLPI primitives are defined in terms of STREAMS messages. All DLPI messages are either type `M_PROTO` or type `M_PCPROTO` and contain the appropriate DLPI message structure. DPLI primitives are exchanged by *confirmed* negotiation—request primitives are issued by the user, which then waits to receive an acknowledge (or error) response primitive from the provider. This type of negotiation is illustrated in Figure 11-2.

Request Primitive   Response Primitive

Data Link Service User

Data Link Service Provider
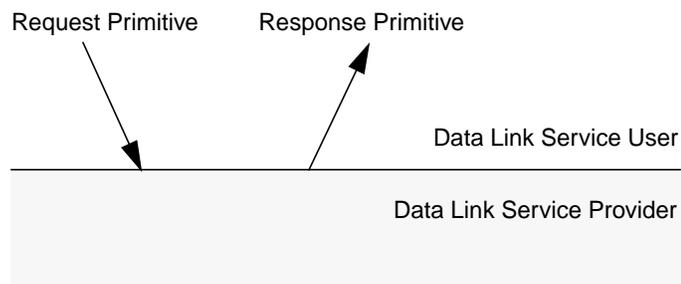
Figure 11-2   DLPI Request/Response Negotiation

DLPI Connectionless Service Primitives

Applications developed to run over FDDI use the DLPI connectionless service primitives, which support the following set of operations:

- Attaching a STREAM to a given addressable device
- Getting and setting the physical address of the attached device
- Binding a data link service access point (SAP) to the STREAM
- Transmitting and receiving data link frames
- Transmitting and receiving LLC XID/TEST messages
- Transmitting and receiving multicast addresses
- Enabling physical and data link promiscuous mode

DLPI Addressing

DLPI addressing is implemented as a DLSAP (data link service access point) address. Each DLSAP consists of a physical address plus a service access point (SAP). The format of the DLSAP is provider-dependent. Figure 11-3 shows the general format of the SunFDDI DLSAP.
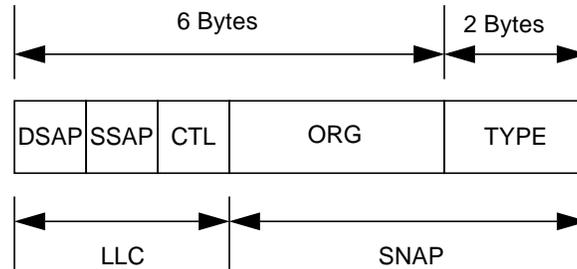
Figure 11-3   SunFDDI DLSAP Format

If an application is bound with a SAP that is greater than 1500 decimal (`0x5dc`), the SunFDDI device driver encodes and decodes the DLSAP using LLC and SNAP. However, if an application is bound with a SAP that is less than or equal to 1500 decimal (`0x5dc`), the SunFDDI device driver expects that the application will handle the LLC and SNAP encoding and decoding.

This behavior is similar to that of the SunFDDI 2.0 device driver, with one important difference—the critical SAP is 255 and not 1500. Thus if an application is bound with a SAP that is greater than 255 decimal (`0xff`), the SunFDDI 2.0 device driver encodes and decodes the DLSAP using LLC and SNAP. However if an application is bound with a SAP that is less than, or equal to, 255 decimal (`0xff`), the SunFDDI 2.0 device driver expects that the application will handle the LLC and SNAP encoding and decoding.

Information returned by the DL_INFO_ACK primitive (in response to a
DL_INFO_REQ) is used to specify the encoding and decoding rules to insert
and extract the DLSAP for a given data link provider. The information
returned by the SunFDDI device driver is shown in Table 11-1.

Table 11-1 DL_INFO_REQ Frame

Element	Value	Description
dl_max_sdu	4352 bytes	Maximum MTU
dl_min_sdu	0 bytes	Minimum MTU
dl_addr_length	8 bytes	Length of DLSAP
dl_mac_type	DL_FDDI	FDDI MAC layer
dl_reserved	0	Unused
dl_current_state	—	Current state of data link provider
dl_sap_length	-2 bytes	Length of DLSAP
dl_service_mode	DL_CLDLS	Connectionless service
dl_qos_length	0	Not supported
dl_qos_offset	0	Not supported
dl_range_length	0	Not supported
dl_range_offset	0	Not supported
dl_provider_style	DL_STYLE2	Can be cloned
dl_addr_offset	76	Offset to address location
dl_version	2	DLPI Version 2
dl_brdcst_addr_length	6	Broadcast address length
dl_brdcst_addr_offset	84	Offset to broadcast address location

Note that the value dl_sap_length is set to minus 2, which indicates that the
SAP is 2 bytes beyond the physical address.

≡ 11

Monitoring the Link Layer with NIT (Solaris 1.x)

The Network Interface Tap (NIT) is used to monitor and recover information from the `ifnet` interface to the link layer of the SunFDDI device driver in a Solaris 1.*x* environment. It is composed of several STREAMS modules and drivers that collectively provide link-level network access. Examples of NIT clients include `rarpd(8C)` and `etherfind(8C)`.

NIT consists of the following components:

- `nit_if(4M)` is a STREAMS module that interacts directly with the SunFDDI device driver. After being bound to a specific instance of the SunFDDI device driver, the `nit_if` device transcribes packets that arrive on the interface and delivers messages to the raw packet output code for transmission.

- `nit_pf(4M)` is a STREAMS module that provides packet filtering services to NIT. It discards all unessential incoming packets and automatically forwards all outgoing packets.

- `nit_buf(4M)` is a STREAMS module that buffers incoming messages and reduces the reliance on repeated `read(2V)` system calls.

A detailed description of these components, including generic examples on using the NIT can be found in the associated man pages for these modules.

Setting Up Servers and Diskless Clients

12

Setting Up a Solaris 2.x Server and Diskless Clients	*page 156*
Setting Up a Solaris 1.x Server and Diskless Clients	*page 165*
Booting a Diskless Client Over SunFDDI	*page 167*
To Boot by Default	*page 168*

This chapter describes how to set up a server so that you can boot and run diskless clients across an FDDI network. It assumes that the server and clients are attached to the same FDDI ring.

Setting up a server for a diskless client running over FDDI involves these steps:

1. Setting up the server so that it can export an operating system for a diskless client and installing the SunFDDI device driver and utilities.

2. Introducing the diskless client to the network.

3. Installing the SunFDDI device driver in the `/export/root/`*<client>* file system for the diskless client.

4. Booting the diskless client over the network.

Setting Up a Solaris 2.x Server and Diskless Clients

When you are setting up a Solaris 2.*x* server so that it can export its operating system, you need to know the number of diskless clients that it will serve. This tells you the amount of disk space you need to reserve in the `/export` file system.

♦ **Use** `df(1M)` **to check the amount of space available. For example, type:**

```
% /usr/bin/df -k /export
Filesystem         kbytes  used   avail  capacity  Mounted on
/dev/dsk/c0t3d0s0  24143   17222  4511   79%       /
```

The figure in the `avail` column tells you how much space is available in the file system.

- For each Solaris 2.*x* client, you will require:
  - 20 Mbytes of disk space under `/export/root`
  - 24 Mbytes (by default) of disk space under `/export/swap`

- For each Solaris 1.*x* client, you will require:
  - 16 Mbytes of disk space under `/export/root`
  - 16 Mbytes (by default) of disk space under `/export/swap`

In addition, you require 10 Mbytes under `/export/exec` for the Solaris 2.*x* template files (Sun-4d, Sun-4m and Sun-4u) plus an additional 15 Mbytes under `/export/exec` for each kernel architecture supported (Sun-4d, Sun-4m and Sun-4u).

For example,

If you have five Sun-4m diskless clients running Solaris 2.*x*, you will require:

- (5 x 20) = 100 Mbytes of disk space under `/export/root`
- (5 x 24) = 120 Mbytes of disk space under `/export/swap`
- (10 + 15) = 25 Mbytes of disk space under `/export/exec`

▼ To Set Up a Solaris 2.*x* Server

Follow these steps to set up a Solaris 2.*x* server to boot diskless clients across an FDDI connection:

1. **Install and configure the relevant SunOS operating system on the server so that it can be exported.**
   See the *Solaris System Configuration and Installation Guide* for detailed instructions on how to set up an operating system for diskless clients.

2. **Check that you have sufficient space in the** /export/root, /export/swap, **and** /export/exec **file systems.**

3. **Install and configure the SunFDDI driver and utilities on the server.**

4. **Ensure that you have an entry for the** sysadmin **group (with a GID of 14) in the group table** /etc/group.
   You need this entry to use admintool to administer remote machines. See *Solaris 2.5 Setting up User Accounts, Printers, and Mail.*

Note – If you are running NIS on your server, you can use admintool to update the local /etc/hosts and /etc/ethers files. You must merge these files manually into the NIS maps afterwards.

▼ To Introduce Diskless Clients to a Solaris 2.*x* Server

Before you can introduce a diskless client to the network you need to know its host name, IP address and Ethernet address. The six-byte Ethernet address is displayed when the machine is switched on. (Note that the Ethernet address of the client is the MAC address stored in NVRAM on its motherboard.)

1. **Log in as a member of the** sysadmin **group.**
   You must log in as a member of the sysadmin group to administer remote machines. Do *not* log in as root or become superuser.

2. **Start** admintool **by typing:**

```
% /usr/bin/admintool &
```

3. **Click SELECT on the Host Manager icon to display the Select Naming Service window.**

```
┌──────────────────────────────────────────────────┐
│ ₒ─▯�ᵰ     Host Manager: Select Naming Service       │
├──────────────────────────────────────────────────┤
│ Naming Service:                                    │
│ ┌─────┐                                            │
│ │NIS+ │  Domain Name: group.Div.Comp.COM           │
│ ├─────┤                                            │
│ │ NIS │  Domain Name: sunknc.France.Sun.COM        │
│ ├─────┤                                            │
│ │None │  Use /etc files on host: papyrus           │
│ └─────┘                                            │
│                                                    │
│                                                    │
│ Show: ▽  All Hosts                                 │
│                                                    │
│             ( Apply )  ( Reset )                   │
│                                                    │
│                        Naming Service: None        │
└──────────────────────────────────────────────────┘
```
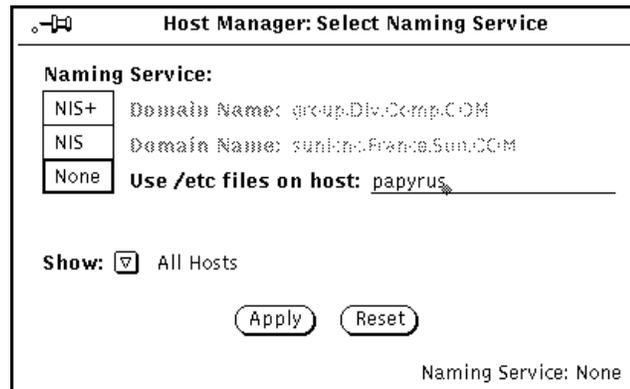
4. **Click SELECT on the button corresponding to the naming service used by the server to display list of available hosts.**
    - If the server is running NIS+, click SELECT on NIS+ and type in your domain name.
    - If the server is running NIS or is not using a naming service, click SELECT on None and type the name of the host on which the files `/etc/hosts` and `/etc/ethers` are located.

5. Click SELECT on Apply to display the Host Manager window.
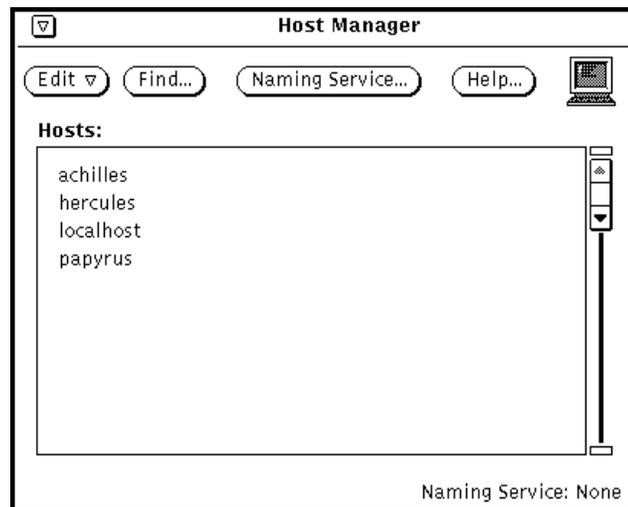
6. Press MENU on the Edit menu button and choose the Add Host... item to activate the Add Host window.
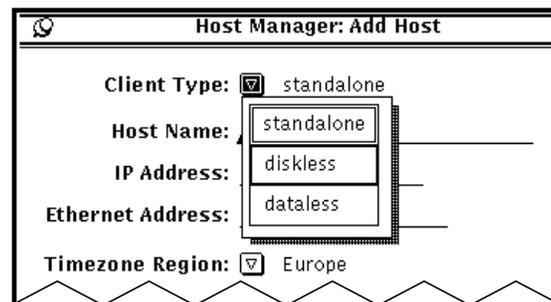
7. **Press MENU on the Client Type menu button. Drag and release MENU to display defaults for diskless clients.**

```
┌─────────────────────────────────────────────┐
│  Ⓠ        Host Manager: Add Host            │
├─────────────────────────────────────────────┤
│          Client Type: ▽  diskless            │
│           Host Name: ◇_____   │
│           IP Address: _____        │
│      Ethernet Address: _____     │
│      Timezone Region: ▽  Europe              │
│            Timezone: ▽  Middle Europe         │
│           File Server: ▽  papyrus             │
│          OS Release: ▽                        │
│            Root Path: /export/root _____   │
│            Swap Path: /export/swap _____   │
│            Swap Size: 24  ▲▼    megabytes     │
│         Terminal Type: sun _____   │
│                                               │
│          ( Add )  ( Reset )  ( Help... )      │
│                          Naming Service: None │
└─────────────────────────────────────────────┘
```

8. **Enter the host name, IP address, and Ethernet address of the diskless client.**
   This information uniquely identifies the diskless client on the network. The six-byte Ethernet (MAC) address has the form X:X:X:X:X:X. It is stored in NVRAM on the motherboard and is displayed when the system is booted.

9. **Use MENU to set the Timezone Region and Timezone that are relevant for your location.**

10. **Press MENU on the File Server menu button to display a list of available servers. Drag and release MENU to specify a file server.**
    The file server is the machine from which the diskless client will receive its operating system. If you are running `admintool` on the server, it will be shown as the default.

11. **Press MENU on the OS Release menu button to display a list of the operating systems available on the server. Drag and release MENU to specify the operating system for the diskless client.**
    Ensure that the operating system you choose is compatible with the architecture of the diskless client.

12. **Specify the** `root` **and** `swap` **paths.**
    By default, these paths are set to `/export/root` and `/export/swap`. If you change these paths, you must ensure that you have sufficient disk space in the file systems you specify.

13. **Specify the size of the swap space reserved for the diskless client.**
    This is normally set to 24 Mbytes. If you increase the amount of swap space reserved for the diskless client, you must ensure that you have sufficient space under `/export/swap` on the server.

14. **If you specified a SunOS 4.x operating system release in Step 11, specify the terminal type used by the diskless client.**
    Any type defined in the `terminfo` database is valid. Common terminal types include `sun`, `tvi910+`, `wyse90`, `vt100`, `vt52`, `adm31`, `h19`, `ansi`. The default is a `sun` terminal.

15. **Click SELECT on the Add button to update the list of recognized hosts.**
    This creates the export file systems for the diskless client on the server and updates the relevant files for the naming service you are using. The operation will take a few minutes. The message "operation completed" appears when the list has been updated and the name of the diskless client is added to the scrolling-list on the Host Manager window.

▼ To Update the NIS Server

If you are running NIS, you must update the NIS map on the NIS server manually. If you are running NIS+ or you are not using a naming service, proceed to "Installing the SunFDDI Driver for a Diskless Client" on page 163.

1. On the diskless client's server (on which you just ran `admintool`), note the new entries created in the following files:

- `/etc/hosts`
- `/etc/ethers`
- `/etc/timezone`
- `/etc/bootparams`

2. **On the NIS server, update the four files with the information taken from the diskless client's server.**

3. **On the NIS server, remake the NIS map by typing:**

```
# cd /var/yp
# make
```

Adding a Solaris 1.x Client Only

If you are updating the NIS map to add a diskless client running Solaris 1.x, then you need to perform the following additional steps:

4. **Change directories to the diskless client's** root **file system.**

```
# cd /export/root/<client>
```

5. **Update the** yp **map for the diskless client:**

```
# mv var/yp- var/yp
```

6. **Replace the** noname **entry in the file** /etc/defaultdomain **with the your local NIS domain name.**

Installing the SunFDDI Driver for a Diskless Client

After you have created the export file systems under /export/root/*<client>* and /export/swap/*<client>* on the server, you must install the SunFDDI device driver for the diskless client.

▼ To Install a Solaris 2.x Client

1. **Use** pkgadd **to install the Solaris 2.x version of SunFDDI so that it is exported to the diskless client.**
By default, the SunFDDI software is installed under /opt.

2. To redirect the installation so that the package is installed in the root partition for the diskless client, type:

```
# pkgadd -R /export/root/<client> —d \
/cdrom/sun_fddi_5_0_sparc/sol_2.5
```

Note – If you are installing SunFDDI on an environment other than Solaris 2.5, replace "`sol_2.5`" with your specific operating environment, as in `sol_2.3` or `sol_2.4`.

Follow the installation instructions contained in the section "Installing and Configuring SunFDDI" on page 13.

▼ To Install a Solaris 1.x Client

SunFDDI for Solaris 1.*x* includes a pre-built kernel to support diskless clients. This kernel is based on the GENERIC kernel configuration and supports all standard devices, plus up to two SunFDDI interfaces.

1. To use the pre-built kernel for a diskless client, install the Solaris 1.*x* version of SunFDDI according to the installation instructions contained in the section Chapter 4, "Installing and Configuring SunFDDI in a Solaris 1.x Environment."
Use /export/root/<*client*>/usr as the destination directory. Since you are trying to install the 1.x version of the driver on a 2.x system, you may get some error messages. Ignore them and continue with the installation.

2. Copy the pre-built kernel to the root partition for the diskless client:.

```
# cd /export/root/<client>/usr/sunlink/nf/sys/<arch>/OBJ
# cp ./vmunix /export/root/<client>
```

Setting Up a Solaris 1.x Server and Diskless Clients

When you are setting up a Solaris 1.*x* server so that it can export its operating system, you need to know the number of diskless clients that it will serve. This tells you the amount of disk space you need to reserve in the `/export` file system.

♦ **Use** `df(1V)` **to check the amount of space available. For example, type:**

```
% /usr/5bin/df -k /export
Filesystem         kbytes   used    avail   capacity   Mounted on
/dev/dsk/c0t3d0s0   24143   17222   4511    79%         /
```

The figure in the `avail` column tells you how much space is available in the file system.

- For each Solaris 2.*x* client, you will require:
  - 20 Mbytes of disk space under `/export/root`
  - 24 Mbytes (by default) of disk space under `/export/swap`

- For each Solaris 1.*x* client, you will require:
  - 16 Mbytes of disk space under `/export/root`
  - 16 Mbytes (by default) of disk space under `/export/swap`

In addition, you require 10 Mbytes under `/export/exec` for the Solaris 2.*x* template files (Sun4, Sun4c, and Sun4m) plus an additional 15 Mbytes under `/export/exec` for each kernel architecture supported (`Sun4`, `Sun4c`, and `Sun4m`).

For example,

If you have five `Sun4c` diskless clients running Solaris 1.*x*, you will require:

- (5 x 16) = 80 Mbytes of disk space under `/export/root`
- (5 x 16) = 80 Mbytes of disk space under `/export/swap`
- (0+15) = 15 Mbytes of disk space under `/export/exec`

▼ To Set Up a Solaris 1.*x* Server

Follow these steps to set up a Solaris 1.*x* server to boot diskless clients across an FDDI connection:

1. **Install and configure the relevant SunOS operating system on the server so that it can be exported.**

2. **Check that you have sufficient space in the** `/export/root`, `/export/swap`**, and** `/export/exec` **file systems.**

3. **Install and configure the SunFDDI driver and utilities on the server.**

▼ To Update the Name Service

Before introducing clients to the server, you must update the name service to register an IP address, host name, and Ethernet (MAC) address for each client.

1. **On the NIS server (or on the diskless client's server if you are not running NIS), modify the following files to enter the relevant information:**
   - `/etc/hosts`
   - `/etc/ethers`
   - `/etc/bootparams`

2. **On the NIS server, remake the NIS map by typing:**

```
# cd /var/yp
# make
```

3. **On the diskless client's server, change directories to the diskless client's** `root` **file system.**

```
# cd /export/root/<client>
```

▼ To Introduce Diskless Clients to a Solaris 1.*x* Server

Before you can introduce a diskless client to the network you need to know its host name, IP address, and Ethernet address. The six-byte Ethernet address is displayed when the machine is switched on. (Note that the Ethernet address of the client is the MAC address stored in NVRAM on its motherboard.)

1. **Log in as** `root` **or become** `superuser`.

2. **Start** `add_client` **in interactive mode by typing:**

```
% /usr/etc/install/add_client -i
```

See the *Solaris 1.x System and Network Administration Manual* for detailed instructions on how to use `add_client` to introduce a diskless client.

3. **Modify the file** `/etc/rc.local` **to enable** `rarpd(8C)` **for the SunFDDI interface automatically when the machine is rebooted.**

```
if [ -d /tftpboot ]; then
#        rarpd -a;                    echo -n ' rarpd' <----- comment out
         rarpd nf<inst>               echo -n ' rarpd' <----- add line
         rpc.bootparamd;              echo -n ' bootparamd'
fi
```

4. **Copy the new kernel to the root partition for the diskless client.**

```
# cp /usr/sunlink/sys/<arch>/OBJ/vmunix /export/root/<client>
```

5. **Reboot the server to invoke the changes to the system.**

Booting a Diskless Client Over SunFDDI

You need to specify the SunFDDI interface when you boot your diskless client over the FDDI network.

▼ To Boot Manually

1. **Check that the server is running and that its FDDI interface is up.**

2. **Switch on the diskless client and press** `L1`**(Stop)-**`a` **to halt the boot procedure.**

3. **At the boot prompt, enter the appropriate** `boot` **command for your machine architecture and boot PROM revision.**

- For Sun-4u clients, type:

```
> b -r /sbus/nf@<slot_number>
```

   or

```
ok boot -r /sbus/nf@<slot_number>
```

- For Sun-4m clients, type:

```
> n
ok boot -r /iommu/sbus/nf@<slot_number>
```

If you do not specify a slot number, the client boots over the first FDDI interface it detects.

For example, for a Sun-4u client, type:

```
ok boot -r /sbus/nf
```

Warning – Once a diskless client has been booted over the SunFDDI interface, do not move the SunFDDI card to a different slot.

▼ **To Boot by Default**

   1. **Check that the server is running and that its FDDI interface is up.**

   2. **Switch on the diskless client and press** `L1`**(Stop)**-`a` **to halt the boot procedure.**

   3. **At the boot prompt, enter the appropriate commands to set the default boot device and diagnostic device.**

- For Sun-4u clients, type:

```
ok setenv boot-device /sbus/nf
ok setenv diag-device /sbus/nf
```

- For Sun-4m clients, type:

```
> n
ok setenv boot-device /iommu/sbus/nf
ok setenv diag-device /iommu/sbus/nf
```

4. **At the boot prompt, enter the appropriate** boot **command for your machine architecture and boot PROM revision.**

- For Sun-4u clients, type:

```
> b
```

  or

```
ok boot
```

- For Sun-4m clients, type:

```
> n
ok boot
```

☰ *12*
_____

Frequently Asked Questions About SunFDDI A≡

This appendix contains the responses to some of the most frequently asked questions about SunFDDI. It also contains references to more detailed information.

1. **The network performance is much lower than I expected. Why can't I transfer data at 100 Mbps over my FDDI connection?**
   If the applications running over FDDI don't use the available bandwidth efficiently you will see much lower performance than expected. See Chapter 5, "Improving Network Performance" for advice on how to get the best performance from your SunFDDI station.

2. **Can I connect a dual-attached FDDI station to two different subnetworks?**
   No. A dual-attached station must be either connected to the primary and secondary rings in a dual-ring configuration, or connected in a dual-homing configuration. See "Dual-Attached Station (DAS)" on page 83 for more information.

3. **Can I use a dual-attached station in single-attached mode?**
   Yes. Simply connect one port to the network. No overhead is incurred when you operate a dual-attached station in this way.

4. **Can I use SunFDDI 5.0 in a redundant single-attached station (RSAS) configuration?**
   No. This configuration, which was supported by SunFDDI 2.0, provides a redundant connection that is used in the event of a link failure. The equivalent configuration for SunFDDI 5.0 is provided by a dual-attached station connected in dual-homing mode. See "Concentrators with Dual-Homing" on page 100 for more information.

5. **Can I use the SunFDDI 5.0 software with the hardware for earlier versions of the SunFDDI products?**
   Yes. The SunFDDI 5.0 software supports SunFDDI 3.0 and 4.0 hardware.

6. **Can I use the SunFDDI 5.0 SBus card with the software for earlier SunFDDI products?**
   No. The SunFDDI 5.0 hardware requires 5.0 software.

7. **Can I use the SunFDDI 5.0 software with a hardware interface from a third-party vendor?**
   No. The device driver does not support any third-party hardware products.

8. **Can I mix SunFDDI 5.0 with earlier SunFDDI products (1.0 and 2.0) in the same machine?**
   No.

9. **Can I connect two single-attached stations "back-to-back", without connecting them to a concentrator?**
   Yes. This configuration is supported by the FDDI specifications. You can also connect two single-attached stations and one dual-attached station directly in a ring, but this configuration is not supported by the FDDI specifications and is not recommended.

10. **Can SunFDDI 5.0 coexist with a SunFDDI 4.0 or 3.0 card in a system?**
    Yes. But the 5.0 software must be installed.

11. **Can I install SunFDDI 5.0 in an SBus expansion box?**
    No. SunFDDI 5.0 is not supported in the SBus expansion box. An SBus expansion box has serious restrictions in both throughput and latency, combined with a significant delay for control between the CPU and SBus boards in general. These restrictions become critical in conjunction with FDDI, which stresses the SBus bandwidth.

12. **What is dual-homing?**
Dual-homing is a special configuration that connects a single dual-attached station to two concentrators so that there is a redundant path, which is used in the event of a link failure. See "Concentrators with Dual-Homing" on page 100 for more information.

13. **What is the MTBF for the various SunFDDI options?**
The mean time before fail (MTBF) for each of the SunFDDI interface cards is:

   - SunFDDI (SAS)   750525 hours
   - SunFDDI Dual (DAS)   446748 hours

   These characteristics were obtained using the standard method for calculating MTBF.

14. **What is the difference between fiber and UTP?**
The FDDI specifications support communication over optical fiber and unshielded twisted-copper pair (UTP). Both physical media support data transfer rates of up to 100 Mbps, but UTP provides a much cheaper solution, particularly where high-quality UTP has been deployed already. However, UTP does not support such large distances between stations and is not suitable in areas that are sensitive to electromagnetic interference.

15. **Why can't I connect port A to port A, or port B to port B?**
This type of configuration is forbidden by the FDDI specifications. It is called "twisting" or "crossing" the ring. If you connect your FDDI station in this configuration, the link status indicator will remain blinking green to indicate an invalid connection. See Chapter 2, "Hardware Installation," for a description of the different diagnostic LEDs and Chapter 8, "FDDI Network Topologies," for a description of the valid network configurations.

16. **Why do I get the error message "`duplicate address`" or "`duplicate macid`"?**
By default, the first SunFDDI interface installed in the machine adopts the host-resident `MAC` address stored on the motherboard. Each subsequent SunFDDI interface takes the card-resident `MAC` address stored in its IDPROM.

   A potential conflict exists between the first SunFDDI 5.0 interface and a SunFDDI 2.0 interface because both of these products take the host-resident MAC address. You must change the `MAC` address assigned to one of these interfaces. See "Changing the Default MAC Address (`nf_macid`)" on page 112 for detailed instructions.

17. Why is performance poor when I use NFS over FDDI?

For a single file transfer, the maximum data transfer rate is limited to 0.5 Mbytes/second when writing to an NFS mounted file system and 1.5 Mbytes/second when reading from an NFS mounted file system. This performance limitation, which affects Ethernet, FDDI, and any other fast network, is due to the NFS version 2 protocol. It will be removed by NFS version 3.

Note that when several NFS operations are performed in parallel (several reads and writes in parallel), the aggregate throughput is much higher with FDDI than it is with Ethernet.

18. How do I disable the SunFDDI interface?

Use `ifconfig(1M)` to disable the SunFDDI interface temporarily. It will be reconfigured when you reboot your machine.

```
# ifconfig nf<inst> down
```

To disable the SunFDDI interface so that it is not reconfigured when you reboot the machine, change the name of the `/etc/hostname.nf`*<inst>* file.

```
# mv /etc/hostname.nf<inst> /etc/oldhostname.nf<inst>
```

19. How do I disable the Ethernet interface?

Use `ifconfig(1M)` to disable the Ethernet interface temporarily. It will be reconfigured when you reboot your machine.

```
# ifconfig le<inst> down
```

To disable the Ethernet interface so that it is not reconfigured when you reboot the machine, change the name of the `/etc/hostname.le`*<inst>* file.

```
# mv /etc/hostname.le<inst> /etc/oldhostname.le<inst>
```

20. **How do I use the host name assigned to the SunFDDI interface as the primary host name for my machine?**
In a Solaris 2.*x* environment, the primary host name is defined in the `/etc/nodename` file. Normally, this file contains the host name assigned to the Ethernet interface built in to your machine. Edit `/etc/nodename` to enter the host name assigned to the SunFDDI interface.

21. **Can I associate several IP addresses with a single SunFDDI interface?**
Yes. To associate multiple IP addresses with a single interface, type:

```
# ifconfig nf0    <ip_address_1> netmask <mask> up
# ifconfig nf0:1 <ip_address_2> netmask <mask> up
# ifconfig nf0:2 <ip_address_3> netmask <mask> up
.
.
.
```

22. **How many SunFDDI 5.0 SBus cards can I install in one machine?**
You can install up to eight SunFDDI 5.0 SBus cards in one machine; however, you cannot use an SBus extender.

23. **What kind of connector can be used to connect SunFDDI 5.0 to optical fiber networks?**
The connector type has been changed for SunFDDI 5.0 to an SC-type connector. See Chapter 2, "Hardware Installation" for a description of FDDI cables and connectors.

24. **What kind of optical fiber can be used to link SunFDDI 5.0?**
SunFDDI uses a 62.5/125 μm standard twin optical fiber. A compliant link design will have no more than 11 db of end-to-end loss. This includes loss incurred in the fiber (typically 1.2 db/km), the connectors (typically 0.5 db/connector), and any bypass switches (typically 2.5 db/switch). It also allows loss margins for source aging (typically 1 to 3 db), safety (typically 1 to 2 db), and temperature variation (typically 1 db).

25. **Does SunFDDI Dual incorporate an optical bypass?**
No. You must order an optical bypass switch from a third-party vendor.

26. Why does my optical bypass switch not work when I plug it in?
Optical bypass switches require a power cycle to be initialized. Plug in the optical bypass switch, switch off the FDDI station, and then reboot to initialize the station. See "Optical Bypass Switches" on page 88 for a more detailed description.

27. How can I increase the distance between two SunFDDI stations?
SunFDDI supports multimode fiber connections only. This limits the maximum distance between stations to two kilometers (1.24 miles). To increase the distance between stations, either:

- Use converters from multi- to single-mode fiber, and a single-mode connection between converters.

- Add more nodes between the stations to act as repeaters.

28. Does SunFDDI take advantage of the 25 MHz SBus?
The SunFDDI SBus board operates at the speed of the host SBus; therefore, if it is installed in a machine with a 25 MHz SBus, it will take advantage of the speed. If there is a performance problem, it is more likely to be related to insufficient CPU cycles to handle the data processing, or congestion on the network, than to the speed of the SBus.

Glossary

This appendix contains a glossary of FDDI-related acronyms, abbreviations, and terms.

ARP

(Address Resolution Protocol). A protocol used within local networks that support broadcast (such as Ethernet or FDDI) that binds IP addresses to MAC addresses. View the current contents of a Sun machine's ARP table with the SunOS command, `arp -a`; specified in RFC 826.

baud

A measure of the rate of data transfer, based on the number of times per second the signal can change states on the transmission line. The transmission line ordinarily uses just two signal states to indicate data transfer, making the baud rate equal the number of bits per second that can be transferred. See also **bps**, **Bps**, **kbps**, and **Mbps**.

bps

(bits per second). A measure of the rate of data transmission. See also, **baud**, **Bps**, **kbps**, and **mbps**.

Bps

(bytes per second). A measure of the rate of data transmission. See also, **baud**, **bps**, **kbps**, and **mbps**.

CFM

(Configuration Management). The portion of CMT that manages the internal configuration of the station's MAC and PHY entities.

CMT

(connection management). The portion of the FDDI SMT that manages the operation of physical layer insertion and removal, and the logical connection of PHY entities to the MAC entities.

DAS

(dual attached station). An FDDI station that connects to both primary and secondary rings; a Class A station, capable of dynamic reconfiguration to accommodate certain levels of failure. Two rings are used to interconnect the stations so that the failure of any station or link causes the remaining stations to reconfigure the network to maintain connectivity: wrapping.

dotted decimal notation

A way of representing the 32-bit Internet address: four decimal integers separated by decimal points, where each integer is the decimal value of one octet of the Internet address. For example, the 32-bit Internet address (binary) `11000000 00001001 11001001 00000111` is represented as `192.9.201.7` (decimal).

ECM

(Entity Coordination Management). The portion of CMT that manages the media interface to the FDDI network, which includes coordinating the activity of all the PHYs associated with that physical attachment and controlling the optional optical bypass function within the station.

FDDI

(*Fiber Distributed Data Interface*). A standard specified by the American National Standards Institute for fiber optic links of up to 100 Mbits/s.

gateway

A dedicated computer that connects to two or more networks and routes packets from one network "community" to the other. Gateways route packets to other gateways until the packets can be delivered directly to their specified destination across one physical network. In general usage, the term identifies any computer that transfers information between networks.

kbps

(kilobits per second). A measure of the rate of data transmission. See also, **baud**, **bps**, **Bps**, and **Mbps**.

LLC

(Logical Link Control). Local area network protocol functions corresponding to the OSI data link layer (layer 2), but providing only service access points (SAP).

MAC

(Media Access Control). Local area network protocol functions corresponding to the OSI data link layer (layer 2), but providing the following: 1) on transmission, assembles data into a frame with address and error-detection fields; 2) on reception, disassembles frame, performs address recognition and error detection; 3) links communications management.

Mbps

(millions of bits per second). A measure of the rate of data transmission. See also, **baud**, **bps**, **Bps**, and **kbps**.

NRZI-4B/5B

(nonreturn to zero inverted). A differential encoding technique that encodes four-bit data groups as five bits (data plus a signal element); satisfies the minimum signal transition requirements to transmit synchronization (clock) within the data stream.

OSI

(Open System Interconnection). The OSI model is a standard for computer-communications protocols and the implementation of these protocols. The model is a product of the *International Standards Organization* (ISO), and specifies a seven-layer architecture with one or more protocol standards at each layer.

PCM

(Physical Connection Management). The portion of CMT that manages a physical connection between the PHY being managed and another PHY, likely in an adjacent station on the FDDI network.

PHY

(Physical Layer Protocol). Performs the clock recovery and serial-to-parallel (on receive) or parallel-to-serial (on transmit) of data between the transmission medium and the MAC entity.

PMD

(physical medium dependent). Specifies the optical-to-electrical conversion mechanism to conform to FDDI.

RARP

(Reverse Address Resolution Protocol). A protocol used by diskless machines and their servers within local networks such as Ethernet or FDDI. In a RARP exchange, a diskless machine broadcasts its MAC address. A boot server responds to that client with the client's IP address, enabling the client to begin the boot process.

RMT

(Ring Management). The portion of CMT that monitors the MAC. It provides a trace mechanism to handle the stuck beaconing condition, and also detects duplicate addresses on the ring.

RSAS

(redundant single-attach station). An FDDI station in which single-attach interfaces are paired, with one active and the other a hot standby. In an RSAS with connections to two different dual-attach concentrators, you enjoy the network availability afforded by concentrators with single-attach connections, plus, for the RSAS itself, the station availability of a dual-attach station.

SAP

(service access point). A logical interface between two adjacent protocol layers (such as LLC and IP).

SAS

(single attached station). An FDDI station that attaches to a single FDDI ring, as distinguished from a dual-attached station, which attaches to both rings in a dual ring.

SMT

(Station Management). The complete standard of PHY layer and frame-based protocols to be supported by conforming FDDI stations.

THT

(Token-Holding Timer). A timer in the FDDI MAC protocol that controls the length of time a station can hold a token for asynchronous transmission.

timing jitter

The deviation of clock recovery that can occur when the receiver attempts to recover both clocking and data from the received signal. The PHY specification describes a phase lock loop (PLL) and elasticity buffer to eliminate jitter.

TRT

(Token Rotation Timer). A timer in the FDDI MAC protocol that measures the time between receipt of tokens. It expires when it reaches a value that exceeds the Target Token Rotation Time (TTRT).

TTRT

(Target Token Rotation Time). A parameter in the FDDI MAC protocol that is negotiated by all of the stations on a ring at the time a ring is brought up. FDDI uses the TTRT in conjunction with the TRT and THT to allocate bandwidth to individual stations on the ring. The FDDI MAC protocol guarantees that the worst-case delay a token will experience while circulating the ring will never exceed twice the negotiated TTRT.

Index

Symbols

`/cdrom/sun_fddi_5_0_sparc`
      directory, 13, 45
`/dev/nf.AUTOLOAD` file, 36
`/etc/hosts` file, 16, 22, 24, 36
`/etc/loadable` script, 36
`/etc/rc.local` file, 35, 37
`/etc/rc2.d` directory, 22
`/etc/rcS.d` directory, 46
`/etc/system` file, 47
`/sys/netinet/in_proto.c` file, 49
`/usr` directory, 29
`/usr/sunlink/nf` directory, 35
`/var/tmp/unbundled` directory, 29

Numerics

`5.0_FDDI_S.log` file, 28
802.3 networks, 106
802.5 networks, 106

A

access delay, 40, 42, 128
`adb` command, 50
`add_drv` command, 61
aggregate attenuation, 89

ANSI/FDDI standards, 76
A-ports, 83
`arch` command, 25
asynchronous
    service, 91
    traffic, 91
attenuation, 77, 88

B

backup, 96
bandwidth, 1, 40, 91, 93, 95, 128
base directory
    installing `SUNWnfr`, 15
    installing `SUNWnft`, 17
basic dual ring, 96
battery, lithium, 3
bidding, 92
bit stream, 79
boot test, 67
bridges, 41, 65
buffers, 41, 49
bulk data transfers, 40
bus topology, 106
bypass switch, 88

C

cable
  failure, 81, 87
  keys, 56
  specification, 2, 77
card-resident MAC address, 57
cdm command, 26
checking the installation, 19, 32
checklist, troubleshooting, 56
claim process, 42, 48, 52, 92
classes, 126
cleaning
  fibre optic cable, 8
code examples, 47, 50
commands
  adb, 50
  add_drv, 61
  arch, 25
  extract_unbundled, 28
  getsockopt, 46, 50
  ifconfig, 19, 32, 58
  mount, 25
  mount_hsfs, 13
  ndd, 43, 44
  netstat, 19, 32
  ping, 19, 32
  pkgadd, 13, 15, 17
  pkgrm, 22
  rm, 35
  setsockopt, 46, 50
  umount, 18, 31
common problems, 55, 61
concentrators, 84
configurations
  dual-ring, 86
  ring of trees, 85
connectionless service primitives, 151
connectors, 77, 83
  fiber optic, 8, 56
console, 132, 136
copper twisted-pair, 76, 77
counter-rotating ring, 81, 96
CPU speed, 40

crossing the ring, 56
cumulative length, 103

D

DAC, dual-attached concentrator, 82, 84,
    85, 95
daemons
  in.routed, 59
  nf_snmd, 34
  vold, 18
DAS, dual-attached station, 82, 83, 87, 95
data link
  provider interface, 150
data queues, 43
data transfer, 76
  rate, 2, 41
declaring IP addresses, 12, 16
default
  buffer size, 41, 49
  IP addresses, 16
  MAC address, 57
  target token rotation time, 42
developing applications, 149
device
  drivers, 36, 56
  names, 4, 30
diagnostic LED, 9, 17, 31, 56
dialogue-based traffic, 78
directories
  /cdrom/sun_fddi_5_0_
      sparc, 13, 45
  /etc/rc2.d, 22
  /etc/rcS.d, 46
  /usr, 29
  /usr/sunlink/nf, 35
  /var/tmp/unbundled, 29
disabling MTU path discovery, 48
disk space, 12, 24
diskless clients, 155
distributed clocking scheme, 77
distribution frame, 104
DL_INFO_ACK, 153

self-test, 67
hierarchical topology, 102
high priority, 92
high water marks, 43
hop count, 108, 110
horizontal distribution frame, 105
host name, 12, 15, 24, 138
host-resident MAC address, 57
hubs, 84

I

IDPROM, 57
`ifconfig` command, 19, 32, 58
`ifnet` command, 154
`ifnet command`, 149
improving
    ring latency, 40
`in.routed` daemon, 59
increased transmission time, 92
independent data paths, 100
inherent redundancy, 105
`install_nf` script, 30
installation
    checking the, 19, 32
    requirements, 12, 24
    scripts, 12, 16, 17, 24, 28
installing
    multiple cards, 4
    SNM agents, 20 to 22, 33 to 35
    the hardware, 5
    the software, 13 to 18, 25 to 31
instance
    names, *See* device names
    numbers, 4, 30
interference, 77
intermediate distribution frame, 104
IP
    addresses, 12, 16, 24, 58
    routing, 106

J

jumpers, 5

K

kbps, 178
kernel space, 41, 49
keying, 83
keys, 56

L

LAN interfaces, 12, 24
`Late_Ct` parameter, 92
latency, 40
LED, diagnostic, 9, 17, 31, 56
link status indicator, 9, 17, 31, 56, 63
list of packages, 14
lithium battery, 3
LLC, logical link control, 152, 179
loadable drivers, 52
loading the CD-ROM
    Solaris 1.*x* installation, 25
local agent, 132, 140
lost frames, 122
low water mark, 43

M

MAC, 108
    address, 57
MAC, medium access control, 75, 76, 78
main distribution frame, 104
major numbers, 61
master SBus slot, 4
maximizing data throughput, 41
maximum
    bandwidth, 40
    length of ring, 40
    number of bypass switches, 89
    number of cards, 2, 4
    number of stations, 40
    throughput, 40

initialization, 81
management, 75, 76
upstream, 82
wrapping, 81, 88
station management (SMT), 2, 86
status information frames, 126
STREAMS
DLPI specification, xxi
modules, 43
subnetwork number, 58
SunFDDI features, 1
SunNet Manager, 131
console, 132, 136
daemons, 66
proxy agent, 131
schemas, 20, 33, 138
SunOS IP routing, 106
SUNWnf package, 21, 61
switches, 5
symbols, 77, 79
synchronization, 77
synchronous
FDDI, 128
service, 91
traffic, 91
sys/dlpi.h file, 150
system performance, 93

T

T_max parameter, 92
T_opr parameter, 42, 92
T_Req parameter, 48
T_req parameter, 42, 48, 92
target
station, 132, 142
token rotation time, 42, 48, 92
TCP
applications, 43
buffers, 49
high water marks, 43
low water mark, 44
window size, 49

TCP/IP networks, 131
tcp_recv_hiwat parameter, 44
tcp_xmit_hiwat parameter, 44
tcp_xmit_lowat parameter, 45
test, boot, 67
theoretical
maximum bandwidth, 40
maximum number of stations, 40
throughput, 40
THT, 92, 180
timed token
rotation protocol, 78, 91, 128
timing jitter, 180
token, 91
ring, 91, 106
rotation timer, 92
total ring length, 40
traffic, 78
transceiver unit, 6
transfer rate, 93
transmission time, 92
transmit packets, 118
transmitters, 77
trap reports, 21, 33, 135, 144
tree of concentrators, 102, 104, 107
trees, 98
troubleshooting checklist, 56
TRT, 92, 181
TTRT, 42, 48, 92, 181
tuning
high performance networks, 40
the high water marks, 43
the low water marks, 44
the maximum transfer unit, 47, 51
the socket options, 46, 50
the target token rotation time, 48, 52
the water marks at boot, 45
twisting the ring, 56
types, 126
types of traffic, 42

U

UDP
    applications, 43
    high water marks, 43
    low water mark, 44
`udp_recv_hiwat` parameter, 44
`udp_xmit_hiwat` parameter, 44
`udp_xmit_lowat` parameter, 45
`umount` command, 18, 31
UNA, 122
unrestricted traffic, 91
unshielded twisted-pair, 76, 77
upstream station, 82
user space, 41, 49
user-configurable hardware, 5
utilities
    `nf_macid`, 112
    `nf_smtmon`, 60
    `nf_stat`, 140
    `nf_sync`, 128
UTP, 76
UTP, unshielded twisted-pair, 77

V

video applications, 40
vmunix, 49
voice applications, 40
volume manager (`vold`), 13, 18, 45

W

wall boxes, 105
water marks, 43
work groups, 98
wrapping, 81, 88

X

X.3.139-1987, 2, 75
X3.148-1988, 2, 75
X3.166-1990, 2, 75
X3.299 R7.3, 2, 75

`XmitP`, 118

SunFDDI SBus Adapter 5.0 User's Guide—September 1996

Reader Comment Card

Your comments and suggestions are important to us. Please let us know what you think about the *SunFDDI SBus Adapter 5.0 User's Guide,* part number 802-6674-10.

1. Were the procedures well documented?   Yes ❑   No ❑

   Please explain: _____

   _____

2. Were the tasks easy to follow?   Yes ❑   No ❑

   Please explain: _____

   _____

3. Were the illustrations clear?   Yes ❑   No ❑

   Please explain: _____

   _____

4. Was the information complete and easy to find?   Yes ❑   No ❑

   Please explain: _____

   _____

5. Do you have additional comments about the SunFDDI SBus Adapter 5.0 User's Guide?
   You can send detailed comments via email to `smcc-docs@sun.com`, or send a fax to *SMCC Doc Feedback* at (415) 786-6443.

   _____

   _____

   _____

   _____

   _____

   _____

   _____

   _____

   Your Name: _____

   Title: _____

   Company Name: _____

   Address: _____

   City: _____ State/Province: _____

   Country: _____ Zip/Postal Code: _____

   Email Address: _____

   Telephone: _____

   Thank you.

AIR MAIL
PAR AVION

IBRS/CCRI No. 808

Sun

INTERNATIONAL BUSINESS REPLY MAIL/REPONSE PAYEE

PERMIT NO. 808   MOUNTAIN VIEW CA

POSTAGE WILL BE PAID BY ADDRESSEE

INFORMATION PRODUCTS
M/S MPK14-108
SUN MICROSYSTEMS INC
2550 GARCIA AVE
MOUNTAIN VIEW CA 94043-9551
UNITED STATES OF AMERICA