

# SunTRI/S 4.0 Installation Guide

---



THE NETWORK IS THE COMPUTER™

**Sun Microsystems, Inc.**  
901 San Antonio Road  
Palo Alto, CA 94303-4900 USA  
650 960-1300  
Fax 650 969-9131

Part No.: 805-6906-10  
October 1998, Revision 05

1998 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A.

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® system, licensed from Novell, Inc., and from the Berkeley 4.3 BSD system, licensed from the University of California. UNIX is a registered trademark in the United States and in other countries and is exclusively licensed by X/Open Company Ltd. Third-party software, including font technology in this product, is protected by copyright and licensed from Sun's suppliers. RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

Sun, Sun Microsystems, the Sun logo, SunTRI, OpenBoot, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and in other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox Corporation in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a nonexclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

Copyright 1998 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303-4900 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie et la décompilation. Aucune partie de ce produit ou de sa documentation associée ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Des parties de ce produit pourront être dérivées du système UNIX® licencié par Novell, Inc. et du système Berkeley 4.3 BSD licencié par l'Université de Californie. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays, et licenciée exclusivement par X/Open Company Ltd. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Sun, Sun Microsystems, le logo Sun, SunTRI, OpenBoot, et Solaris sont des marques déposées ou enregistrées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC, utilisées sous licence, sont des marques déposées ou enregistrées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Les utilisateurs d'interfaces graphiques OPEN LOOK® et Sun™ ont été développés de Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox Corporation pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique, cette licence couvrant aussi les licenciés de Sun qui mettent en place les utilisateurs d'interfaces graphiques OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, NI EXPRESSE NI IMPLICITE, Y COMPRIS, ET SANS QUE CETTE LISTE NE SOIT LIMITATIVE, DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DES PRODUITS A REpondre A UNE UTILISATION PARTICULIERE OU LE FAIT QU'ILS NE SOIENT PAS CONTREFAISANTS DE PRODUITS DE TIERS.



# Contents

---

- 1. SunTRI/S Adapter Introduction 1**
  - Token Ring Networks 1
  - SunTRI/S Adapter Overview 3
    - Hardware Overview 3
    - Software Overview 4
  - Preparing for Installation 4
    - Requirements 4
    - Adapter Handling Precautions 5
    - Recommended Types of STP cabling links 5
    - Platform Compatibility 5
  - Identifying the SunTRI/S Adapter 6
    - Card Slot Selection 8
      - Slot Selection Notes 8
  - Installing the Adapter 8
    - To Install the SunTRI/S Adapter 9
- 2. Installing the SunTRI/S 4.0 Software Packages 11**
  - Software Installation 11
    - Installation and Configuration Overview 11
    - Summary of Installation Steps 12

- ▼ To Remove Previous Versions of the SunTRI/S Software 12
- ▼ To Mount the CD-ROM 13
- ▼ To Install the Driver 13
  
- 3. Software Configuration 15**
  - Configuring the SunTRI/S Software for TCP/IP 15
  - Editing `tr.conf` Parameters 16
    - Setting the Maximum Transmission Unit (MTU) 16
    - Disabling Source Routing 17
    - Disabling ARI/FCI Soft Error Reporting 17
    - Configuring the Operating Mode 18
  - Configuring the Locally Administered Address (LAA) 18
  - Configuring the SunTRI/S Adapter as the Sole Network Interface 19
  - Configuring Internetwork Routing 20
  - Resource Configuration Parameter Tuning 20
  
- 4. Troubleshooting and Diagnostics 23**
  - Troubleshooting Tools 23
    - Ifconfig 23
    - Netstat 24
    - Ping 24
    - Snoop 25
    - Spray 25
    - Trsr 25
  - Maintenance Hints 26
  - System Console Messages 26
    - Error Messages 27
    - Incompatibility with Local Source Routing Bridges 40

<b>A. Advanced Topics for Programmers</b>	<b>41</b>
LLC Packet Encapsulation	41
MAC Frame Headers	42
Source Route Support	43
OPEN Stages	44
Multicasting	45
The <code>trkstat</code> Data Structure	46
<b>B. Token Ring Architecture Overview</b>	<b>49</b>
The Data Link Control Layer	50
DLC.LAN.MGR	50
The Logical Link (LLC) Sub-layer	51
The Medium Access Control (MAC) Sub-layer	51
The Physical Layer	52
Basic Operating Concepts	52
The Ring	52
Multiple Ring Connections	53
MAC Sub-layer Operating Modes	53
The Active Monitor	54
Maintaining the Master Clock	54
Ensuring Proper Ring Delay	55
Initiating Neighbor Notification	55
Monitoring Neighbor Notification	55
Monitoring Token and Frame Transmission	56
Detecting Lost Tokens and Frames	56
Purging the Ring	57
Attaching the Ring	57
Phase 0: Lobe Test and Physical Attachment	57

Phase 1: Monitor Check	58
Phase 2: Duplicate Address Check	58
Phase 3: Participate in Neighbor Notification	58
Phase 4: Request Parameters	58
Soft Error Detection and Reporting	59
Hard Error Detection and Reporting	59
Addressing and Routing Concepts	60
Source Routing	60
Addresses	61
Individual and Group Addresses	61
Universal and Local Address Administration	62
Null Addresses	62
All-Stations Broadcast Addresses	62
Functional Addresses	62
Broadcast Terminology	62
<b>C. Network Configuration Planning</b>	<b>65</b>
Token Ring Network Topologies	65
General Guidelines	66
Performance	66
Reliability	67
Affinity Groups	67
Growth	69
Backbone Configuration Strengths	72
Backbone Configuration Weaknesses	73
Impact on Performance	73
Token Ring Bridge Concept	74
Source Routing	74

Setting Up Routers 75

Selecting the MAC Address 75





# Tables

---

TABLE 1-1	Platform Compatibility	5
TABLE 3-1	MTU Sizes	16
TABLE 3-2	Source Routing Values	17
TABLE 3-3	ARI/FCI Soft Error Reporting Values	17
TABLE 3-4	Operating Mode Values	18
TABLE 4-1	Flag Descriptions	26
TABLE 4-2	Error Descriptions	27
TABLE 4-3	Adapter Initialization Error Codes	29
TABLE 4-4	Possible Errors During Open Phases	31
TABLE 4-5	Command Reject Error Codes	35
TABLE 4-6	Command Codes	35
TABLE 4-7	Bad Token Ring Status Bits	37
TABLE 4-8	Ring Status Bit Function	37
TABLE 4-9	Error Causes	39
TABLE A-1	OPEN stages	45
TABLE A-2	<code>trkstat</code> Statistical Counters	47
TABLE B-1	Seven Layers of Open Systems Interconnection (OSI)	49
TABLE C-1	Overhead Determining Factors	73



# Preface

---

This guide provides the information needed to install and configure the hardware and software for the SunTRI/S Adapter. These instructions are designed for an experienced system administrator with networking knowledge.

---

## Using UNIX Commands

This document may not contain information on basic UNIX commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- *Solaris Handbook for Sun Peripherals*
- AnswerBook™ online documentation for the Solaris software environment
- Other software documentation that you received with your system

---

# Typographic Conventions

TABLE P-1 Typographic Conventions

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

---

# Shell Prompts

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	<i>machine_name%</i>
C shell superuser	<i>machine_name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

---

# Sun Documentation on the Web

The `docs.sun.com` web site enables you to access Sun technical documentation on

the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

<http://docs.sun.com>

---

## Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`smcc-docs@sun.com`

Please include the part number of your document in the subject line of your email.



# SunTRI/S Adapter Introduction

---

The SunTRI/S adapter supports the Solaris 2.5.1, Solaris 2.6, and Solaris 7 operating environments. The function of the SunTRI/S adapter is to provide a means for connecting Sun SPARCstations to a token ring network.

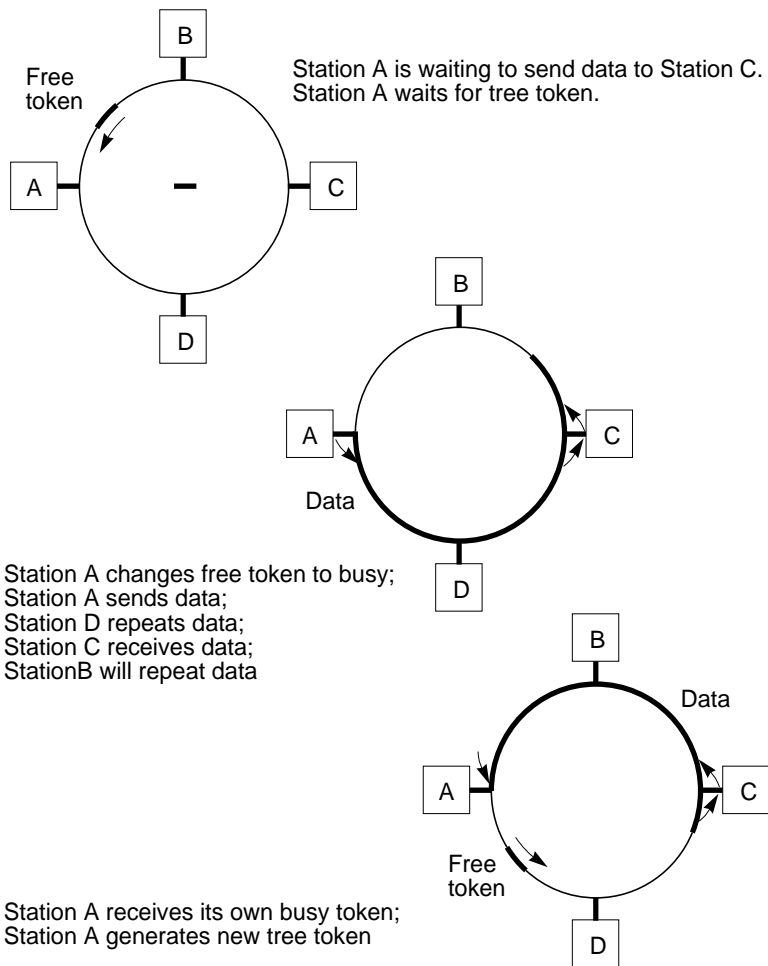
---

## Token Ring Networks

A token ring network is a physically star-wired local area network that interconnects various devices such as personal computers and workstations into a logical ring configuration. The cabling system consists of wiring concentrators, connectors, and end stations.

The Sun token ring protocol conforms to the IEEE 802.5-1988 standard. Token ring refers to the media access control (MAC) portion of the link layer (DLC) as well as the entire physical layer (PHY). Access to the ring is controlled by a bit pattern, called a token, that circulates from station to station around the ring. The circulating token bit pattern means that any station can use the ring. Capturing the token means that a station changes the token bit pattern so that it is no longer that of a token but of a data frame. The sending station then sends its data within the information field of the frame. The frame also includes the destination address of the destination station. The frame is passed from station to station until it arrives at the proper destination. At the destination station, the frame is altered to indicate that the address was recognized and that the data was copied. The frame is then passed back to the original sending station, where the sending station checks to see that the destination station copied the data. If there is no more data to be sent, the sending station alters the frame's bit configuration so that it now functions as a free token available to another station on the ring.

If a station fails, it's physically switched out of the ring, dynamically. The ring is then automatically re-configured. When the station has been repaired, the ring is automatically re-configured to include the added station.



For a detailed description of the token ring protocol, refer to the IEEE 802.5 standard.



---

# SunTRI/S Adapter Overview

The SBus-based token ring interface provides access to 4-Mbps or 16-Mbps token ring local area networks. The interface design is based on the Texas Instruments TMS380 token ring chip set. It contains EPROM and DRAM used for various purposes as described in this chapter.

## Hardware Overview

The token ring interface connects to the SBus via a custom DMA ASIC chip. The DMA provides a fast and efficient 32-bit transfer of data from the token ring logic to the host processor, across the bus. The DMA packs and unpacks 16-bit halfwords into 32-bit fullwords as well as providing a direct I/O path for accessing and programming the onboard registers. The token ring supports both slave and master cycles across the bus.

The token ring interface contains 512-Kbytes local memory that is accessible by both DMA and direct CPU access. A portion of the local memory is used for execution of the TMS380 firmware. The remainder is used for buffering data packets.

The design is based on the Texas Instrument's TMS380 and TMS38053 Token Ring chip sets, which are fully compatible with the IEEE 802.5 Token Ring specifications.

The electrical interface to the token ring is provided by a token ring module. The module provides all the line conditioning and signal recovery necessary to physically connect the token ring interface to the token ring network.

The SunTRI/S adapter can run at either 4-Mbps or 16-Mbps by installing or removing a single jumper on the adapter. The 16-Mbps selection is selected when the adapter is manufactured.

The token ring interface has a 9-pin D-connector socket that allows attachment to a token ring network using the IBM STP cable type 1, 2, 6, or 9. Both the 4-Mbps and 16-Mbps data rate networks can be attached via this connector.

---

**Note** – To use UTP (Unshielded Twisted Pair) cabling, you must also have a media filter.

---

# Software Overview

SunTRI/S software supports the IEEE 802.5 standards for token ring networks. The IEEE standard specifies the lower two layers of the OSI 7-layer model. The two layers are the Physical layer (layer 1) and the Data Link layer (layer 2). The Data Link layer is further divided into the Logical Link sub-layer (LLC) and the Media Access Control sub-layer (MAC).

The token ring driver is a multi-threaded, loadable, clonable, STREAMS hardware driver that supports the connectionless Data Link Provider Interface, `dspi(7p)`, over a token ring controller. Multiple token ring controllers installed within the system are supported by the driver.

SunTRI/S software can support different protocol architectures, concurrently, via the SNAP encapsulation technique of RFC1042. From this SNAP encapsulation, high-level applications can communicate through their different protocols over the same SunTRI/S interface. Support also exists for adding different protocol packages (not included with SunTRI/S). These protocol packages include OSI and other protocols available directly from Sun, or through 3rd party vendors. TCP/IP is implicit with the Solaris operating environment.

The software driver also provides source routing which enables the workstation to access multiple ring networks connected by source-route bridges. Locally-administered addressing is also supported and aids in management of certain user/vendor-specific network configurations.

Support for IBM LAN Manager is provided by the TMS380 MAC-level firmware which complies with the IEEE 802.5 standard.

---

## Preparing for Installation



---

**Caution** – Always turn off the power to your computer before installing or removing any adapters.

---

## Requirements

To install the SunTRI/S adapter, you must have a supported Sun SPARCstation with an available SBus slot.

# Adapter Handling Precautions

Do not remove the adapter from its protective antistatic packaging until you are ready to use it.

Before handling the adapter, discharge static electricity from your body by touching a protective grounding device or the metal chassis of your computer. As an added precaution, wear a grounding strap.

Always handle the adapter carefully by its edges. Avoid touching any of the components or connections, particularly the gold edge-connectors.

## Recommended Types of STP cabling links

---

**Note** – Refer to the information supplied by your MAU or switch vendor for maximum supported cable lengths.

---

The 9 pin 'D' type connector is compatible with IBM STP cable types 1, 2, 6, and 9.

## Platform Compatibility

The SunTRI/S adapter is compatible with Solaris 2.5.1, Solaris 2.6, and Solaris 7 operating environments. TABLE 1-1 describes the platforms and number of supported cards for each adapter.

TABLE 1-1 Platform Compatibility

Platform	Open Boot PROM Level	Number of Adapters Supported
SPARCstation 1	OBP Version 1.1 or later	1
SPARCstation 1+	OBP Version 1.3 or later	1
SPARCstation 2	OBP Version 2.0 or later	1
SPARCstation IPC	OBP Version 1.6 or later	1
SPARCstation IPX	OBP Version 2.0 or later	1
SPARCstation 5	OBP Version 2.14 or later	1
SPARCstation 10	OBP Version 2.7 or later	1
Ultra 1 Series	OBP Version 3.0 or later	1
Ultra 2 Series	OBP Version 3.5 or later	1

TABLE 1-1 Platform Compatibility

Platform	Open Boot PROM Level	Number of Adapters Supported
SPARCsystem 600 Series	OBP Version 2.0 or later	1
SPARCstation 4	OBP Version 2.14 or later	1
SPARCstation 20	OBP Version 2.7 or later	1
SPARCstation LX/Classic	OBP Version 2.9 or later	1
SPARCserver 1000/1000E	OBP Version 2.9 or later	3
SPARCcenter 2000/2000E	OBP Version 2.9 or later	4
Ultra Exx00 Series	OBP Version 3.2.3 or later	4

The PROM version appears when the machine is booted up. To find the PROM version without rebooting the machine:

1. Go to the PROM level of the operating system by typing:

```
# stop-A
```

2. At the `ok` prompt, type:

```
ok banner
```

3. Use the `go` command to return to the operating system.

```
ok go
```

---

**Note** – The SunTRI/S adapter is not supported in the SBus expansion (XBox) chassis on Solaris 2.5.1, Solaris 2.6, or Solaris 7 operating environments.

---

---

## Identifying the SunTRI/S Adapter

The SunTRI/S adapter comes packaged in an antistatic bag to protect it from harmful electrical charges.

Before you attempt to install a new adapter, or replace a defective one, be sure that you have the correct replacement. The following illustration shows an outline of the SunTRI/S adapter. The data rate selection jumper is displayed in a blowup on the lower, right, corner of the drawing.

The jumper on the SunTRI/S adapter is used to select between the 4-Mbps and the 16-Mbps data rate. With the jumper installed, the 16-Mbps rate is selected.

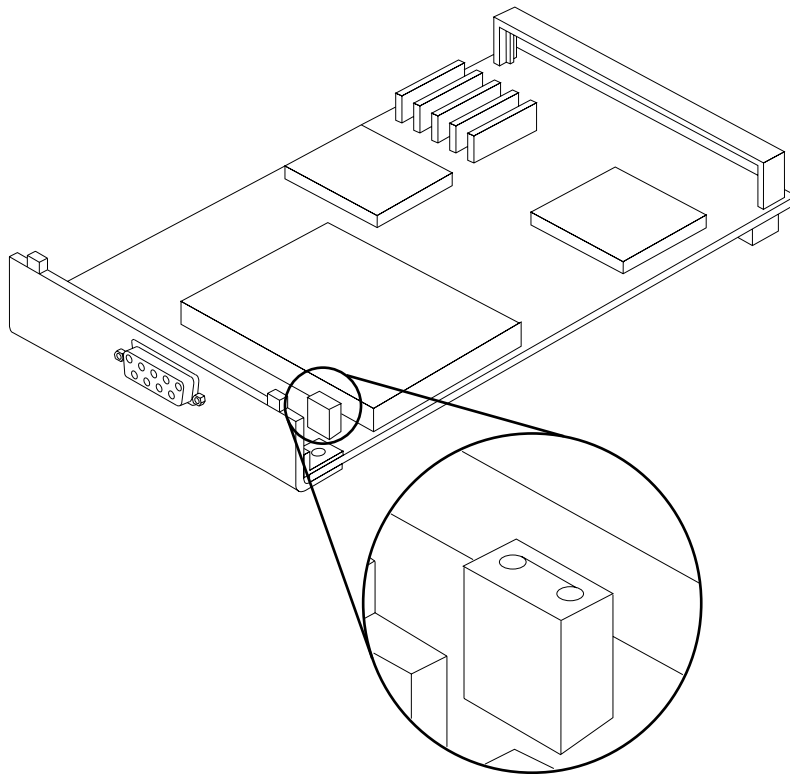
The jumper is installed when the board is manufactured. To run at the 4-Mbps rate, remove the jumper by pulling it straight up, out of the board. Since you might want to run at the 16-Mbps speed in the future, you should be careful not to misplace the jumper. To prevent the possibility of losing the jumper, you should insert one side of the jumper onto just one of the pins on the adapter.



---

**Caution** – To prevent possible damage to electronic circuitry, you should always turn off power to the system when removing or installing any cables, adapters, or components on the adapters. This includes switching the data rate selection jumper.

---



## Card Slot Selection

When you install the SunTRI/S adapters in your machine for the first time, the instance numbers for the device driver start with 0 (for example tr0) and increase incrementally by one, for each additional adapter. The mapping between the instance number and the adapter follows the SBus slot position. The driver, tr0, will be associated with the first adapter. The driver, tr1, will be associated with the adapter in the next adapter. Under Solaris 2.5.1, Solaris 2.6, or Solaris 7, if a SunTRI/S adapter is moved from its initial slot to a different slot, the system will find the device in a different slot and assume that it is a new instance of the device. The system will then assign a new instance number to the device.

Once the system has seen the device in a particular slot, the instance number remains bound to it.

## Slot Selection Notes

When installing the SunTRI/S adapter on a Sun4d system, make sure that it resides by itself on a separate system board. Also ensure that no onboard devices are in use on the system board where the SunTRI/S adapter is installed.

When installing the SunTRI/S adapter on an Ultra Enterprise 3x00, 4x00, 5x00, or 6x00 system, make sure that it resides by itself on a separate SBus I/O channel. Also ensure that no onboard devices are in use on the SBus I/O channel where the SunTRI/S adapter is installed.

When installing the SunTRI/S adapter on an Ultra Enterprise 10000 SBus I/O board, make sure that only one SunTRI/S adapter is installed on each I/O board. Also ensure that you do not install any SBus adapter in the second slot.

Sun4d and Sun4u systems require a revision -03 SunTRI/S adapter.

The SunTRI/S adapter is not compatible with SBus slot 3 of the SPARCstation 1 or SPARCstation 1+.

---

## Installing the Adapter

The following section describes how to install the SunTRI/S adapter.

# To Install the SunTRI/S Adapter



---

**Caution** – Always turn off the power to your computer before installing or removing any adapters.

---

---

**Note** – Refer to your system installation or service manual for more detailed instructions on the following tasks.

---

- 1. Power off you system using the standard shut down procedures described in the *Solaris Handbook for Sun Peripherals* or your system service manual.**  
The *Solaris Handbook for Sun Peripherals* is shipped with the Solaris operating environment software and is available in the online AnswerBook documentation.
- 2. Open the system unit.**
- 3. Remove the filler plate from an empty SBus slot.**
- 4. Depending on the system you are installing into, you may have to remove the adapter bracket that is located on the backpanel of the card.**
- 5. Applying even pressure at both corners of the adapter, push the SBus adapter into the SBus slot until it is firmly seated.**
- 6. Secure the adapter by using the retaining screws from the filler plate that you removed.**
- 7. Connect a 9-pin D-type connector for STP cabling.**
- 8. Connect the other end of the data-cable to a token-ring network.**
- 9. Power-up the machine.**

You can now install the SunTRI/S 4.0 software packages.





# Installing the SunTRI/S 4.0 Software Packages

---

This chapter provides step-by-step instructions for installing the SunTRI/S 4.0 software. The package is installed using `pkgadd`, which installs the software from a mounted CD-ROM.

---

## Software Installation

This section provides instructions for installing the `tr` device driver.

## Installation and Configuration Overview

Complete installation of the SunTRI/S software packages requires these steps:

1. Install the SunTRI/S adapter. See “Installing the Adapter” on page 8 for installation instructions.
2. Run `pkgrm` to remove old instances of the SunTRI/S software, if necessary.
3. Install the SunTRI/S 4.0 software packages, from a mounted CD-ROM, using `pkgadd`.
4. Configure the software.
5. Reboot the system to bring it up on the token ring network.

# Summary of Installation Steps

Following is a summary of the steps required to install SunTRI/S software packages. Complete procedures for these steps are given in the following sections of this chapter.

1. Verify installation requirements as specified in “Requirements” on page 4. If you did not read this section, read it now.
2. Place the CD-ROM in the CD-ROM drive.
3. Mount the CD-ROM.
4. Install the SunTRI/S 4.0 software packages using `pkgadd`.
5. Unmount and eject the CD-ROM.

## ▼ To Remove Previous Versions of the SunTRI/S Software

Before installing the SunTRI/S 4.0 software on your system, check your system to see if previous versions of the SunTRI/S software are installed. If older SunTRI/S software (earlier than 4.0) exists, you must remove the older software and then install the new SunTRI/S 4.0 software.

1. Use the `pkginfo` command to check the system for an older SunTRI/S software package:

```
# /usr/bin/pkginfo | grep SUNWtris
system SUNWtris SunTRI/S Driver and Utilities
```

- If you do not find any SunTRI/S packages, skip to the next section, “To Mount the CD-ROM” on page 13 to continue with the software installation.
  - If you do find a SunTRI/S package, you must remove it as described in Step 2.
2. As superuser (root), use the `pkgrm` command to remove the existing SunTRI/S software package.

The following is an example of removing the package `SUNWtris`:

```
# /usr/sbin/pkgrm SUNWtris
The following package is currently installed:
SUNWtris SunTRI/S Driver and Utilities
(sparc) 3.x,REV=1.x
```

## ▼ To Mount the CD-ROM

- **Insert the SunTRI 4.0 CD-ROM into a CD-ROM drive that is connected to your system.**
  - If your system is running Volume Manager, it should automatically mount the SunTRI 4.0 Software CD-ROM under the `/cdrom/suntri_4_0/` directory.
  - If your system is not running Volume Manager, mount the CD-ROM as follows:

```
# mkdir /cdrom
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom
```

## ▼ To Install the Driver

1. **As superuser, type the following to install the SunTRI/S 4.0 software packages from the CD:**

```
# pkgadd -d /cdrom/suntri_4_0/Product/sbus SUNWtrsd SUNWtrsu
```

2. **If prompted, answer `y` to permit `pkgadd` to launch the post-installation packages.**

```
This package contains scripts which will be executed with super-
user permission during the process of installing this package.

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

`pkgadd` will continue to install the software and run the post-installation script. When the installation is complete, you will see messages saying that the software has been installed successfully.

3. **Use the `pkginfo` command to verify that the software has been installed.**

```
# pkginfo SUNWtrsd SUNWtrsu
system SUNWtrsd SunTRI/S (SBus) Driver
system SUNWtrsu SunTRI/S (SBus) Man Pages and Utilities
```

#### 4. Unmount and eject the SunTRI 4.0 CD-ROM.

```
# cd /  
# umount /cdrom/suntri_4_0  
# eject cdrom
```

#### 5. Configure your SunTRI/S interface(s).

You must set the network configuration of your SunTRI/S interface(s) before you can use it. See Chapter 3 for configuration instructions.

## Software Configuration

---

---

### Configuring the SunTRI/S Software for TCP/IP

This section describes how to configure the SunTRI/S adapter software for TCP/IP. You must reboot your system for these changes to become active.

1. **At the command line, use the `grep` command to search the `/etc/path_to_inst` file for `tr` instances.**

In the following example, the SunTRI/S adapter has the instance "0".

```
# grep tr /etc/path_to_inst
"/sbus@1f,0/network@1,100000" 0 "tr"
```

2. **Create an entry in the `/etc/hosts` file for each active `tr` interface.**

The following example shows how to edit the `/etc/hosts` file to add the 192.9.200.66 IP address for the `tr0` interface.

```
# echo "192.9.200.66 cat-sys" >> /etc/hosts
# cat /etc/hosts
#
# Internet host table
#
127.0.0.1      localhost
129.144.131.154 hs4      loghost
192.9.200.66  cat-sys
```

3. Set up each `hostname.tr<num>` file to correspond with each `tr` device in the `path_to_inst` file.

The following example shows the `/etc/hostname.tr0` file required for a system which has a `tr` interface with the hostname `cat-sys`.

```
# echo "cat-sys" > /etc/hostname.tr0
# cat /etc/hostname.tr0
# cat-sys
```

---

## Editing `tr.conf` Parameters

The following parameters can be set in the `/kernel/drv/tr.conf` file. You must reboot your system for these changes to become active.

### Setting the Maximum Transmission Unit (MTU)

Sun supports the IEEE 802.5 Token Ring Standard Maximum Transmission Unit (MTU) size of 17800 octets, (LF=0x111). All hosts should use the same MTU size on any particular network. Additionally, if different types of IEEE 802 networks are connected by transparent link layer bridges, all hosts on all of these networks should use the same MTU size.

To modify the MTU, change the MTU index in `/kernel/drv/tr.conf` and then reboot the system. The MTU index affects all SunTRI/S adapters in the system.

The maximum MTU sizes supported are: 4472 for 4 Mbps operation and 17800 for 16 Mbps operation. These are the rates specified by the Token Ring chip set on the SunTRI/S adapter.

TABLE 3-1 MTU Sizes

MTU Index	MTU Size (bytes)	MTU Index	MTU Size (bytes)
0	516	4	8144
1	1470	5	11407
2	2052	6	17800
3	4472		

The default value of the MTU index is 3 (4472 bytes).

## Disabling Source Routing

Source routing is the method used within the token ring network architecture to route frames through a multiple ring local area network. A route is a path taken by a frame as it travels through a network from the originating station to a destination station.

By default, source routing is enabled. To disable source routing, set the `sr` value in the `/kernel/drv/tr.conf` file to 0.

**TABLE 3-2** Source Routing Values

---

The <code>sr</code> values can be set to:	
<code>sr = 1</code>	Enables source routing
<code>sr = 0</code>	Disables source routing

---

## Disabling ARI/FCI Soft Error Reporting

In 1989, the Token Ring committee's recommendations on the use of the Address Recognized Indicator/Frame Copied Indicator (ARI/FCI) changed. The old recommendation was to use the bits to confirm the receipt or delivery of frames. The new recommendation is to use the bits to report soft errors. This new recommendation gave rise to issues in networks that had devices that were developed based on the old recommendation and in networks with devices developed since 1989 that did not adhere to the new recommendation.

The `ari` parameter in the `/kernel/drv/tr.conf` file can be used to set ARI/FCI soft error reporting. By default, the ARI/FCI soft error reporting parameter is enabled. If you have an older network device, you may need to disable ARI/FCI error reporting by setting the `ari` parameter to 1.

**TABLE 3-3** ARI/FCI Soft Error Reporting Values

---

The <code>sr</code> values can be set to:	
<code>ari = 0</code>	Enables ARI/FCI error reporting
<code>ari = 1</code>	Disables ARI/FCI error reporting

---

# Configuring the Operating Mode

The SunTRI/S adapter supports both classic and DTR mode of operation. The `mode` parameter in the `/kernel/drv/tr.conf` file can be used to set the operating mode.

By default, the adapter is set to classic mode (half duplex). If the `mode` is set to DTR (Dedicated Token Ring), the adapter will come up in full duplex mode. If the `mode` is set to auto, the adapter will automatically choose between classic and DTR mode, depending on the capabilities of the switch or MAU.

**TABLE 3-4** Operating Mode Values

---

The mode values can be set to:	
<code>mode = 0</code>	Classic mode
<code>mode = 1</code>	Auto mode
<code>mode = 2</code>	DTR mode

---

---

## Configuring the Locally Administered Address (LAA)

The LAA is part of the token ring standard specification. You may need to use an LAA for some protocols, such as DECNET or SNA. To use an LAA, create a file with execute permission in the `/etc/rcS.d` directory, such as `/etc/rcS.d/S20trLAA`, with



the `ifconfig trX ether XX:XX:XX:XX:XX:XX` command. The adapter instance is represented by `trX` and the LAA for that adapter is used in place of `XX:XX:XX:XX:XX:XX`.

```
# /sbin/sh
case "$1" in
    'start')
        echo "Configuring Token Ring LAA..."
        /sbin/ifconfig trX ether XX:XX:XX:XX:XX:XX
        ;;
    'stop')
        echo "Stop of Token Ring LAA is not implemented."
        ;;
    *)
        echo "Usage: $0 { start | stop }"
        ;;
esac
```

For example, to use an LAA of `04:00:ab:cd:11:12` on the `tr0` interface, use the following command within the `/etc/rcS.d/S20trLAA` file.

```
# /sbin/ifconfig tr0 ether 04:00:ab:cd:11:12
```

---

## Configuring the SunTRI/S Adapter as the Sole Network Interface

The SunTRI/S interface can be used as your sole network interface. You must reboot your system for these change to become active.

1. **To do so, delete the existing `/etc/hostname.le0` or `/etc/hostname.hme0` file on your system.**

```
# rm /etc/hostname.le0
```

2. **Edit the `/etc/nodename` and `/etc/net/*/hosts` files. Change the hostnames in these files to match that of the SunTRI/S adapter.**

In the following example, the SunTRI/S adapter is named `cat-sys1`.

```
# cat /etc/nodename
cat-sys1
#
# cat /etc/net/ticlts/hosts
cat-sys1 cat-sys1
# cat /etc/net/ticots/hosts
cat-sys1 cat-sys1
# cat /etc/net/ticotsord/hosts
cat-sys1 cat-sys1
```

---

## Configuring Internetwork Routing

Any system with two or more network interfaces (for example, Ethernet and Token Ring) may be an internetwork router. By default, the route daemon supports internetwork routing on any such system by broadcasting IP routing information regularly to all directly connected networks and hosts. If your system is not meant to be an internetwork router, you can suppress these broadcasts by creating an `/etc/notrouter` file using `touch /etc/notrouter`.

If you intend to configure the system as a router by installing a Token Ring interface alone with other LAN interfaces, make sure that the MAC addresses between `tr` and other LAN interfaces are different by configuring a locally administered address. See “Configuring the Locally Administered Address (LAA)” on page 18.

---

## Resource Configuration Parameter Tuning

The SunTRI/S driver is shipped with 64 buffers (in 2K bytes size each) for receiving and transmitting packets. This configuration should be adequate under normal situations. However, the Token Ring interface throughput may be sluggish under heavy load, and may even lock up for indefinite time. This is especially true under NFS related operations.

This problem can be resolved by increasing the number of buffers available in the driver. The tunable parameter `tr_nbufs` can be set in the file `/etc/system`. Add this line in the file if it does not already exist:

```
set tr:tr_nbufs=xxx
```

Where `xxx` is the number of 2K buffers desired. You should not see the value less than the default value (64). Proper setting of this parameter requires “tuning.” Numbers between 400 and 500 should be reasonable for medium load.

You need to reboot the system after you have updated the `/etc/system` file for the changes to take effect.



# Troubleshooting and Diagnostics

---

This section offers some suggestions to assist you when you are having trouble with your SunTRI/S adapter.

---

## Troubleshooting Tools

There are several Solaris operating environment utilities useful for troubleshooting and monitoring network traffic. These include `ifconfig`, `netstat`, `ping`, `snoop`, and `spray`. The `trsr` utility is also included with the SunTRI/S software packages for troubleshooting.

### Ifconfig

The `ifconfig` command can be used to check the status of interfaces used for TCP/IP traffic. `ifconfig` can also be used to check to see if the token ring interfaces were brought up.

To check all interfaces, use the following command:

```
# ifconfig -a
```

To check a particular interface, use a command similar to the following:

```
# ifconfig tr0
tr0: flags=1863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,MULTI-
BCAST> mtu 4472
    inet <IP_address> netmask <netmask> broadcast <address>
    ether <MAC address>
```

The output string:

<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,MULTI-BCAST>  
indicates that the SunTRI/S interface is correctly installed and configured.

## Netstat

After installing the SunTRI/S software, you can verify that the installation was successful with the `netstat` utility. Use `netstat (1M)` to check for the `tr` interfaces. For example:

```
# netstat -i
```

## Ping

`ping` is useful for checking to see if another host is alive, or if communication between two machines is possible. Note that the token ring interface hostname must be used. An example usage is:

```
# ping sake-tr
```

When attempting to get two machines to communicate, first ensure that the hardware and software are configured correctly by using `ifconfig`. Then, attempt to `ping` one system from the other and vice versa. If the `ping` succeeds, the two stations have been configured correctly and are communicating. `ping` might not work with some vendors' IP implementations.

# Snoop

The `snoop` utility allows you to monitor network traffic. The `snoop` utility monitors token ring activity by decoding certain token ring fields, such as the source routing information field. For example, to run `snoop` on the `tr0` interface, use the following command:

```
# snoop -d tr0
```

Refer to your system documentation, or the `snoop` man page for a more detailed explanation about the `snoop` utility.

# Spray

`spray` is a UCB-developed utility that only works between machines running the `sprayd` process. Do not `spray` a non-Sun system.

# Trsr

This the source route table examination utility is used to examine the Source Route Table.

The format for executing this utility is: `trsr -a | -d mac_addr`. The `-a` command displays all of the routes in the table. The `-d` command deletes the source route table entry for the specified `mac_addr`.

The source routing table gets updated every 4 minutes. This is only effective when IP periodically probes the route table based on the `ip_ire_flush_interval` (default is 20 minutes). This parameter is configurable in the IP driver through `ndd(1M)`. This is applicable if the network topology is changed.

An example of executing a source route examination follows:

```
# trsr -a
Dev      Mac-Addr      Flags      Timeout      Len  Dir  Mtu  Route
---      -
tr0      8:0:20:c:10:ab Rslvd      4 (min)      6   0   4472 (1)1(2)
tr0      8:0:20:a:8a:17 Local       2 (min)
tr0      8:0:20:11:e3:7 Rslvd      4 (min)      6   1   4472 (2)1(1)
```

The flags represent the status of the route. TABLE 4-1 describes each flag.

**TABLE 4-1** Flag Descriptions

Flag	Status
Pend	Waiting for response
Rslvd	All is well
Local	Local, don't use source route
Dying	Waiting for delete

---

## Maintenance Hints

Occasionally you may find it useful to disconnect the SunTRI/S adapter from the network. To do so, use the `ifconfig` command while logged in to your system as root. For example, to bring a `tr0` interface down, type:.

```
# ifconfig tr0 down
```

A subsequent `ifconfig up` command can be used to reconnect the adapter to the network.

```
# ifconfig tr0 up
```

---

## System Console Messages

Messages associated with token ring operation are displayed on the host system's console.



# Error Messages

SunTRI/S is brought on line with the `/sbin/ifconfig` command line issued as `root`, or by incorporating it into the boot-up process. If the `/sbin/ifconfig` command line is successful, the SunTRI/S adapter is brought on line and the `ring up` message appears. The following table lists error descriptions for each error index `<n>`.

**TABLE 4-2** Error Descriptions

Error Index	Error Description
01	Error during MAC code downloading
02	Adapter Bring Up Diagnostics failed
03	Initialization failed
04	Transmit halt command failed
05	Close command to remove adapter from ring failed
06	Close command to remove adapter from ring timed out
07	Adapter reset failed
08	Register tests on Eagle chip failed
09	Token ring adapter Prom test failed
10	Token ring adapter DMA chip test failed
11	Token ring adapter card failure indicated by adapter check interrupt
12	Network signal loss
13	Lobe wire fault
14	Bad adapter auto-closed from ring
15	Adapter removed from bad network ring
16	Network ring is bad
17	Ring beaconing
18	Duplicate node address on network ring
19	Ring Parameter Server not responding to initialization requests

Various error messages that can occur during bring up are listed below.

The first six messages pertain to the initial bring-up of the token ring. If an `ifconfig` is successful, none of these messages will appear.

`tr: attempt <number> to open controller failed`

Attempts to open the token ring interface into the ring have failed.

tr: device is a slave-only slot

The token ring card has been installed in a non-DMA master slot.

Action: Halt the machine and power down. Remove the adapter from the SBus slave slot and insert it into a DMA master slot. Power on the machine and retry the operation.

tr: Bad token ring adapter card <02>

The internal Bring-up Diagnostics (BUD) on the TMS380 chip failed.

Action: Halt the machine, power down, and replace the SunTRI/S adapter.

tr: Bad token ring adapter card <03>

The initialize command on the TMS380 chip failed.

Action: Halt the machine, power down, and power on. If it still fails, power down, replace the adapter, and power on.

tr: bring-up diagnostics not OK

There is a problem in the token ring Eagle chipset.

Action: Replace the SunTRI/S adapter.

tr: Contex/interrupt test error

Bring-up diagnostics detected that the Eagle chipset failed the contex or interrupt test.

Action: Replace the SunTRI/S adapter.

tr: ddi\_add\_intr failed

The system has failed to IO-map the token ring.

Action: Retry.

tr: ddi\_dma\_addr\_setup of bufs failed <size>

The system has failed to register the interrupt procedure for token ring.

Action: Retry.

tr: ddi\_map\_regs for tr prom failed

The system has failed to IO-map the token ring PROM.

Action: Check that SunTRI/S adapter is inserted correctly in the system.

tr: ddi\_map\_regs for tr regsp failed

The system has failed to IO-map the token ring registers.

Action: Check that SunTRI/S adapter is inserted correctly in the system.

tr: ddi\_map\_regs for tr s4dma failed

The system has failed to IO-map the token ring's DMA register.  
 Action: Check that SunTRI/S adapter is inserted correctly in the system.

tr: Init Error<error code>

The Token Ring chip set has failed initialization.

See TABLE 4-3 for the adapter initialization error codes.

If any other errors are displayed by the `tr` device, carefully note any error numbers and text and contact your Sun service provider.

**TABLE 4-3** Adapter Initialization Error Codes

Error Code Bits				Error Condition
12	13	14	15	
0	0	0	1	Invalid initialization block. 22 bytes were not passed.
0	0	1	0	Invalid options. This code is returned if the PARITY_ENABLE bits are not equal or if reserved bits are not zero in the INIT_OPTIONS. This code is also returned if the LLC_ENABLE bit is set and the LLC is not present.
0	0	1	1	Invalid receive burst size. RECEIVE_BURST_SIZE is odd.
0	1	0	0	Invalid transmit burst count. TRANSMIT_BURST_SIZE is odd.
0	1	0	1	Invalid DMA abort threshold. DMA_ABORT_THRESHOLD is zero.
0	1	1	0	Invalid SCB. SCB_ADDRESS is odd.
0	1	1	1	Invalid SSB. SSB_ADDRESS is odd.
1	0	0	0	DIO parity. A parity error occurred during a DIO write operation.
1	0	0	1	DMA timeout. The adapter timed out (11 seconds) waiting for a test DMA transfer to complete.
1	0	1	0	DMA parity error. A parity error occurred during the DMA tests, and the operation was tried as specified by DMA_ABORT_THRESHOLD.

**TABLE 4-3** Adapter Initialization Error Codes

Error Code Bits	Error Condition
1 0 1 1	DMA bus error. The DMA test encountered a bus error, and the operation was tried as specified by DMA_ABORT_THRESHOLD.
1 1 0 0	DMA data error. Initialize DMA test failed because of a data compare error.
1 1 0 1	Adapter check. The adapter encountered an unrecoverable hardware error.

tr: Init Incomplete: <error code>

An Open command to the adapter failed. The <error code> can be any of the failures listed in TABLE 4-3. The first five entries in the list below, indicate the phase of the Open operation. The remaining entries contain the particular errors that can occur during the various phases.

Action: If a failure occurs during the Lobe Media Test or the Physical Insertion Test, recheck your local adapter configuration, including the cables and the adapter data rate. If a failure occurs during one of the later three Open Command Phase tests, copy the status information and contact your System Administrator.

Verify that the ring and the MSAU are operational. Try using a different MSAU port or a different MSAU. Verify that the speed of the ring is the same as that of the SunTRI/S adapter. If Open command address problems occur, ensure that the locally-administered address is unique.

Phases of the Open Operation:

```
< lobe media failure >  
< physical insertion failure >  
< address verification failure >  
< ring poll failure >  
< request initialization failure >
```

**TABLE 4-4** Possible Errors During Open Phases

Error	Description
< function failure >	Either the adapter failed the Lobe Media test while transmitting through its lobe at the wiring concentrator, or MAC frames were received during the Lobe Media test phase of insertion.
< signal loss >	A signal loss condition was detected at the receiver input of the adapter during the Open process (either when it wrapped or inserted into the ring).
< time-out failure >	The adapter failed to logically insert onto the ring before the expiration of the insertion timer. Each phase of the insertion process must complete before the insertion timer expires (18 seconds).
< ring failure >	Occurs if the adapter becomes Active Monitor and times out when attempting to purge the ring. The adapter is unable to receive its own Ring Purge MAC frames.
< ring beaconing >	After physically inserting, the adapter received a Beacon MAC frame, or a Monitor Contention timeout occurred during Phase 1.
< duplicate node address >	During the Address Verification Phase, the adapter determined that another station on the ring had the same node address.
< request initialization >	The adapter determined that a Ring Parameter Server (RPS) is present on the ring and does not respond to a Request Initialization MAC frame. (If no RPS is present, the adapter will not return this code.)
< remove received >	The adapter received a Remove Adapter MAC frame during the insertion process.

tr: Initialization SSB or SCB Compare Failed

Token ring chipset has failed initialization comparison.

Action: retry.

tr: Initial receive or transmit command failed

After opening the controller onto the ring, there was a failure of either the initial receive or transmit command.

Action: Use ifconfig to bring down the interface, then use it to bring up the interface. Retry the operation.

tr: Initial test error

The bring-up diagnostic has failed due to initial test error.

Action: retry or use a new adapter.

tr: Instruction test error

The bring-up diagnostic has failed due to instruction test error.

Action: retry or use a new adapter.

tr: kstat\_create\_failed

System has failed to register token ring statistics.

Action: retry.

tr: Log command incomplete

The token ring chipset has failed the token ring Read Error Log command.

Action: do nothing.

tr: Protocol handler hardware error

The bring-up diagnostic has failed due to protocol handler error.

Action: retry or use a new adapter.

tr: RAM error

The bring-up diagnostic has failed due to RAM error.

Action: retry or use a new adapter.

tr: Software checksum error

The bring-up diagnostic has failed due to software Checksum Error.

Action: retry or use a new adapter.

tr: System interface register error

The bring-up diagnostic has failed due to system interface error.

Action: retry or use a new adapter.

tr: Token ring adapter insertion onto ring failed 3 times

Attempts to open the controller and insert into the ring, failed multiple times.

Action: Verify that the ring and the MSAU are operational. Try using a different MSAU port or a different MSAU.

tr: trintr: unknown interrupt

Interrupt that came in is unknown.

Action: contact your Sun service provider.

tr: trstart: msg too big <frame size>

Packet size exceeds the limit for transmission.

Action: please use smaller packet size in your application.

tr: tr\_sr\_create\_entry kmem\_alloc failed

System failed to allocate memory for the Source Routing Table.

Action: increase memory.

tr: trattach: kmem\_alloc failed

System failed to allocate memory for token ring software per device data structure.

Action: increase memory.

tr: trinit failed: out of buffers

During initialization, system failed to get a buffer for receiving packets

Action: retry.

tr: unknown bring up diag error

The bring-up diagnostic has failed due to a unknown hardware error.

Action: retry or use a new adapter.

tr: Unrecoverable hardware error

The bring- up diagnostic has failed.

Action: retry or use a new adapter.

tr: Access Priority Error

The transmit descriptors in the token ring driver have an error due to incorrect access priority.

Action: do nothing, tr will recover automatically.

tr: Frame Size Error

The transmit descriptors in the token ring driver have an error due to incorrect frame size.

Action: do nothing, tr will recover automatically.

tr: Frame Start Error

The transmit descriptors in the token ring driver have an error due to incorrect frame start format.

Action: do nothing, tr will recover automatically.

tr: Illegal Frame Format Error

The transmit descriptors in the token ring driver have an error due to incorrect frame format.

Action: do nothing, tr will recover automatically.

tr: Illegal Op-code

The token ring card has a fatal problem due to an illegal opcode.

Action: replace the adapter.

tr: Invalid Interrupt

The token ring card has a fatal problem due to an invalid interrupt within the Eagle chipset.

Action: replace the adapter.

tr: Invalid Error Interrupt

The token ring card has a fatal problem due to an invalid error interrupt within the Eagle chipset.

Action: replace the adapter.

tr: Invalid XOP Request

The token ring card has a fatal problem due to an invalid XOP request within the Eagle chipset.

Action: replace the adapter.

tr: Odd Address Error

The transmit descriptors in the token ring driver have an error due to illegal odd addressing.

Action: do nothing, tr will recover automatically.

tr: Parity Errors

The token ring card has a fatal problem due to parity errors.

Action: replace the adapter.

tr: Transmit Threshold Error

The transmit descriptors in the token ring driver have an error due to the transmit threshold being exceeded.

Action: do nothing, tr will recover automatically.

tr: Unenabled MAC Frame Error

The transmit descriptors in the token ring driver have an error due to a disabled MAC frame being detected.

Action: do nothing, tr will recover automatically.

trintr: Command Reject <error>/<command code>



A command issued by system to the token ring card has been rejected.  
 Action: The following tables list the command reject errors and command codes.

**TABLE 4-5** Command Reject Error Codes

Bit	Definition
0	ILLEGAL_COMMAND: Set when an unknown command is issued to the adapter. This bit is set when commands that are valid only when the LLC interface is enabled, are issued, and the LLC is not enabled.
1	ADDRESS_ERROR: Set if any address field in the SCB is odd (not word-aligned).
2	ADAPTER_OPEN: Set if a command is issued when an adapter is open and that command is normally honored only if the adapter is closed.
3	ADAPTER_CLOSED: Set if a command is issued when the adapter is closed, and the command is normally honored only when the adapter is open.
4	SAME_COMMAND: Set if a command is issued and the same command is already executing.
5 - 15	RESERVED: Set to zero.

**TABLE 4-6** Command Codes

Code	Command
0000	ILLEGAL COMMAND
0003	OPEN
0004	TRANSMIT
0005	TRANSMIT.HALT
0006	RECEIVE
0007	CLOSE
0008	SET.GROUP.ADDRESS
0009	SET.FUNCTIONAL.ADDRESS
000A	READ.ERROR.LOG
000B	READ.ADAPTER
000D	MODIFY.OPEN.PARAMETERS
000E	RESTORE.OPEN.PARAMETERS

**TABLE 4-6** Command Codes

Code	Command
000F	SET.FIRST.16.GROUP.ADDRESS
0010	SET.BRIDGE.PARMS
0011	CONFIG.BRIDGE.PARMS

trintr: Scb clear interrupt

The system has issued a interrupt to indicate to system that the system command block is clear and ready for usage.

Action: do nothing.

trintr: TR\_XMITSTS: TRXMT\_LSTERR

The transmit descriptors in the token ring driver have errors.

Action: do nothing, tr will recover automatically.

trintr: TRCMD\_XMITHALT ?

The Transmit Halt command issued by the system has completed successfully.

Action: do nothing.

trintr: TRCMD\_CLOSE ?

The Close command issued by the system has completed successfully.

Action: do nothing.

trintr: TRCMD\_FUNCADDR ?

The Set Functional Addresses command issued by the system has completed successfully.

Action: do nothing.

trintr: TRCMD\_MODOPEN ?

The Modify Parameters command issued by the system has completed successfully.

Action: do nothing.

trintr: TRCMD\_OPEN ?

The Open command issued by the system has complete successfully.

Action: do nothing.

tr: ring up

The ring is up and operational.

Action: Informational message, no action required.

tr: ring down

This message is displayed when the ring goes down due to a Auto Remove Error, Lobe Wire Fault, Remove Received, or simply because the interface was downed via ifconfig.

Action: Try to ifconfig the SunTRI/S adapter back up.

tr: Bad token ring adapter card <11>Token ring adapter check status=<status bits>

An Adapter Check interrupt was received indicating a serious problem with the controller. The controller will automatically be reinitialized. Run a full complement of diagnostics on the token ring adapter.

Action: Record the information displayed on the console, replace the adapter, if possible, and contact your Sun service provider.

**TABLE 4-7** Bad Token Ring Status Bits

Bit <error cause>
15 - Controller Parity Error
14 - DMA Read Abort
13 - DMA Write Abort
12, 11, 9, 8, 7, 6, 2, 1 Internal TMS380 Chip Problem
5 - Internal DMA Under-run when transmitting on to the ring
4 - Internal DMA over-run when receiving from the ring

Ring status: <status>

<status> is a hexadecimal value. This message is displayed when the adapter detects an unusual condition.

**TABLE 4-8** Ring Status Bit Function

Bit	Definition
15 msb	SIGNAL_LOSS: 1 = The adapter detected a signal loss on the ring.
14	HARD_ERROR: 1 = The adapter is transmitting or receiving beacon frames to or from the ring.
13	SOFT_ERROR: 1 = The adapter transmitted a report error MAC frame.
12	TRANSMIT_BEACON: 1 = The adapter is transmitting beacon frames to the ring.
11	LOBE_WIRE_FAULT: 1 = The adapter detected an open or short circuit in the cable between the adapter and the wiring concentrator.

**TABLE 4-8** Ring Status Bit Function

Bit	Definition
10	AUTO_REMOVAL_ERROR: 1 = The adapter failed the Lobe Wrap Test resulting from the Beacon Auto-removal Process and deinserted from the ring. The adapter is closed and remains so following adapter initialization.
9	RESERVED:
8	REMOVE_RECEIVED: 1 = The adapter received a Remove Ring Station MAC Frame Request and deinserted from the ring. The adapter is closed and remains so following adapter initialization.
7	COUNTER_OVERFLOW: 1 = One of the adapter's error counters incremented from 254 to 255. Use the READ.ERROR.LOG command to determine which counter overflowed.
6	SINGLE_STATION: 1 = The adapter sensed that it is the only station on the ring. This bit resets to 0 when another station inserts into the ring.
5	SINGLE_STATION: 1 = The adapter sensed that it is the only station on the ring. This bit resets to 0 when another station inserts into the ring.
4-0	RESERVED: Set to 0

Any of the following messages may also appear after the Ring Status message.

tr: signal loss

Bit 15. Indicates that the controller has detected a loss of signal on the ring.

Action: Contact your System Administrator for assistance

tr: beaconing activity: xmitting or receiving beacon frames

Bit 14. The adapter has detected an unusual occurrence that requires beaconing activity.

Action: No action required if the situation corrects itself. Otherwise, contact your System Administrator for assistance

tr: xmitting beacon frames

Bit 12. The adapter has detected an unusual occurrence that requires beacon activity.

Action: No action required if the situation corrects itself. Otherwise, contact your System Administrator for assistance

tr: open/short in cable

Indicates that the controller has detected an open, or short, circuit between the adapter and the wiring concentrator.

Action: Insure that the cable is plugged into the MSAU correctly. Repair or replace the cable or MSAU. Retry the operation.

tr: beacon auto-removal

Bit 10. Indicates that the adapter has failed the Lobe Wrap Test and has de-inserted from the ring.

Action: ifconfig the adapter back up. If the same error persists, contact your System Administrator.

tr: received remove ring station MAC frame

Bit 8. Indicates that the adapter has received a Remove Ring Station MAC frame and has de-inserted from the ring.

Action: ifconfig the adapter back up. If the same error persists, contact your System Administrator.

tr: error counter overflow

Bit 7. One of the adapter's error counters has exceeded 255. These counters keep track of conditions such as line error, receive congestion, and lost frame error. The following table describes these errors.

Action: Contact your System Administrator for assistance with error analysis.

**TABLE 4-9** Error Causes

Error	Cause
Line Error	Frame Check Sequence errors, code violations, etc.
Burst Error	Loss of signal for four half-bit times.
ARI/FCI Error	The upstream neighbor cannot set Address Recognized Indicator or Frame Copied Indicator
Lost Frame	Adapter is transmitting, but does not sense end of frame.
Recv Congestion	No buffer space for incoming frame.
Frame Copied	Receives a frame with ARI on and correct node address.
Token Error	Act monitor only missing/circulating token.
DMA Bus Error	Excessive bus errors.
DMA Parity Errors	Excessive parity errors.

tr: single station

Bit 6. The controller is the only token station on the ring.

Action: Informational message, no action required.

tr: MAC claim token frames detected

Bit 5. This message indicates that the adapter has sensed Claim Token MAC frames indicating that the ring is in recovery.

Action: Informational message, no action required.

tr: MTU of packet <size> from <rhost> not equal to system MTU of <size>

The MTU size of the source-routed packet from the source is less than this system's MTU size. As a result, the source cannot receive the packets from this system because they are too large. The incoming source-routed packets will be discarded. Action: Refer to Source Route Support, in Appendix A, for details about adjusting the MTU size.

ip\_rput: DL\_ERROR\_ACK for 29, errno 1, unix 0

The token ring driver received an Ethernet encapsulated packet on the streams fast path. This error can be ignored.

```
ip: joining multicasts failed on tr0 - will use link layer
broadcasts for multicast
```

This message occurs because token ring does not support Ethernet-type multicasts. This error can be ignored.

## Incompatibility with Local Source Routing Bridges

Symptom: Receive Congestion messages at the IBM Bridge machine.

The IBM Token Ring Bridge Software, Version 2.1 (and earlier), has a slight incompatibility with the SunTRI/S 4.0 software. Even though it results in spurious Receive Congestion messages on the Bridge console, no data is lost. Version 2.2 of the IBM Token Ring Bridge Software solves this problem.

The problem might appear in the following configuration:

- SunTRI/S, Version 4.0, at 16 Mbps
- IBM Token Ring Bridge Software, Version 2.1, and earlier
- Parallel bridge

The source of the problem is Routing Information packets sent by the in.routed daemon on the Sun workstation.

## Advanced Topics for Programmers

---

This appendix covers advanced topics that are not needed to install the token ring product. The topics covered here are intended to bridge the gap if you are not familiar with UNIX and/or 802.5. Specific aspects of the Sun implementation of the IEEE 802.5 Token Ring standard over the Solaris operating system are discussed. The areas covered include LLC Packet Encapsulation, MAC Frame Formats, Source Route Support, the Stages of the Open command, Multicasting, and the `trkstat` Data Structure.

---

### LLC Packet Encapsulation

This information is a summary of the standard method of encapsulating IP packets on IEEE 802 networks. A complete reference is given in RFC 1042. IP packets are sent on IEEE 802 networks encapsulated within the 802.2 LLC and SNAP data link layer headers. The standard release supports LLC type 1 communication. LLC I support is provided from within the driver and no additional software must be registered above the driver. If an LLC type 2 is registered above the driver, the LLC 1 modifications are omitted and the upper layer LLC II deals with LLC level packet encapsulation issues. Trailer encapsulation is not supported.

If no LLC II is registered above the token ring driver, packets arrive at the device driver in the same format as ethernet packets. See the following example:

DA	SA	Type	Data
----	----	------	------

The token ring driver interface modifies the packets to conform to the RFC 1042 standard by inserting the 802.2 LLC and SNAP headers. It also completes most of the MAC header. The format of packets sent to the TMS380 chipset is shown below:

<b>AC</b>	<b>FC</b>	<b>DA</b>	<b>SA</b>	<b>RI*</b>	<b>LCC_SNAP_HDR</b>	<b>Data</b>
-----------	-----------	-----------	-----------	------------	---------------------	-------------

\* RI field is optional.

A more detailed inspection of the LLC and SNAP headers is shown below:

<b>DSAP</b>	<b>SSAP</b>	<b>Control</b>	<b>Org Code</b>	
<b>0xaa</b>	<b>0xaa</b>	<b>0x3 (UI)</b>	<b>0x0</b>	

The DSAP and SSAP values are set to 0xaa, the assigned global SAP for SNAP. The control value is 0x3, representing Unnumbered Information Frames. If there is no LLC II registered above the driver and the control field is not equal to Unnumbered Information, the received packet must be either a TEST or XID packet with a DSAP and SSAP of 0x0; Otherwise, the packet is dropped. The EtherType field is the value of the ethernet type field.

## MAC Frame Headers

The SunTRI/S driver (`tr`) completes most of the MAC header including the AC, FC, Source and Destination addresses, and the optional Routing information field. The entire frame sent to the chipset is shown below:

<b>AC</b>	<b>FC</b>	<b>DA</b>	<b>SA</b>	<b>RI*</b>	<b>Data</b>
-----------	-----------	-----------	-----------	------------	-------------

\* RI field is optional.



The AC, or Access Control field, is set to 0x10 indicating that it is part of a frame. The priority portion is set to zero. Multiple priority levels are not currently supported through a programmatic interface. The FC, or Frame Control field, is set to 0x40 indicating that this is an LLC frame as opposed to a MAC frame.

Sending neither a MAC nor LLC packet type is currently undefined in the 802.5 specification, and the value of 1x for the frame type bits of the FC is reserved for future use. Therefore, using this out-of-hand mechanism may result in fundamental incompatibilities with the 802.5 standard.

## Source Route Support

A non-source route host can coexist on a large network with source route stations, and can communicate to the local stations on the ring, but cannot communicate through the source route bridge.

A more detailed inspection of the routing information field is shown below:

<b>RT</b>	<b>LTH</b>	<b>D</b>	<b>LF</b>	<b>r</b>	<b>RD1</b>	<b>RD2</b>	<b>...RDn</b>
-----------	------------	----------	-----------	----------	------------	------------	---------------

The Route Type fields initiated on output are: Specifically Routed (RT=0xx), all Routes Explorer (RT=100), and Spanning Tree Explorer (RT=110). The All-Routes Explorer is used with ARP, TEST, and XID packets to discover a route. Spanning Tree broadcasts are used for datagrams with an IP broadcast address. This is necessary to prevent multiple copies being received by a destination. All route types including Spanning Tree Routed (RT=111) are accepted on input for backwards compatibility.

The Length value (LTH=XXXXX) indicates the length in octets of the Routing field. Currently, Sun supports even values between two and eighteen, inclusive. Thus, a maximum of seven intermediate bridges may separate two stations that are attempting to communicate using the source route enhancement.

The Direction field (D=X) indicates, to a bridge, whether a frame is traveling from the originating station to the target or vice versa.

The Largest Frame (LF) bits specify the largest frame size of the data field. Sun supports the IEEE 802.5 Token Ring Standard Maximum Transmission Unit (MTU) size of 17800 octets, (LF=0x111). All hosts should use the same MTU size on any particular network. Additionally, if different types of IEEE 802 networks are connected by transparent link layer bridges, all hosts on all of these networks should use the same MTU size.

To Modify the MTU:

- Change the MTU value in `/kernel/drv/tr.conf`.
- Reboot.

The maximum mtu sizes supported are: 4472 for 4 Mbps operation and 17800 for 16 Mbps operation. These values are specified by the Token Ring chip set on the SunTRI/S adapter.

The next four bits (r) are reserved.

The RDn fields are 16 bit fields that indicate the path or the route between nodes on different physical rings.

Initially, an All-Routes Broadcast packet is sent out to determine a route. An entry is created in the Source Route Table indicating that a route is pending. All packets to be transmitted during the period that an entry is pending are sent out immediately without a Source Route Header. If an entry does not complete within three minutes, it is purged. If a packet is received, the entry is completed. The rule is that if the packet is on the local ring, RI length field of 2 (LTH=2), then all further packets sent to that particular station are tagged as non-source route. Otherwise, the routing information is cached into the Source Route Table. If there is a period of inactivity greater than 30 minutes, the entry is purged. Currently, there is no programmatic interface to the Source Route Table, thus the only way to purge the table is to `ifconfig` the interface down and then back up again. However, individual entries can be deleted by using the `trsr` command.

## OPEN Stages

On a ring topology many functions must be performed before a station is opened on the ring and is fully operational and ready for data transfer. When you attempt to bring up SunTRI/S through the `ifconfig` command, the initial step is to download code to the RAM on the SunTRI/S controller. After this is accomplished, the internal Bring-up Diagnostics (BUD) on the TMS380 chipset are initiated. This currently takes approximately two seconds. After BUD completes successfully, there is a short initialization command and then an OPEN command.

There are five stages to the OPEN command when using the TMS380 chipset. TABLE A-1 describes these stages.

TABLE A-1 OPEN stages

Stage	Description
Lobe Media Test	This is a 1-2 second test performed by the TMS380 chipset that sends 2048 1500-byte packets in a loopback mode. This tests the receivers and the cabling up to the MSAU.
Physical Insertion	This is the time required to charge up a capacitor and trip a relay on a passive wiring concentrator (MSAU). This can take a few seconds.
Address Verification Participation in Ring Poll Initialization	These last three stages can take up to 18 seconds when the first station attempts to Request Initialization onto the ring. This is because there are no <b>Active Monitor Present MAC</b> frames which are sent every seven seconds.

If there is any problem opening onto the ring due to an error message indicating that the OPEN command failed, the existence of the error message itself indicates that the software was probably installed correctly but the failure is more likely due to a hardware problem. Recheck all hardware connections including the SBus connection and both ends of the token ring cable. Note that even if everything is installed correctly and there are multiple stations on the ring, the ifconfig command will take approximately 10 seconds to complete.

## Multicasting

The TMS380 Token Ring chipset supports two types of addresses: (1) Group, and (2) Functional.

A Group Address is one that is typically an unrestricted address except that the group bits are 0xC0 in the high-order byte. There must be an exact MAC Address match for the Group Address to be received. There is only a single Group Address available, so implementing multiple Multicast Addresses is impossible through the single Group Address.

Functional Addresses are bit-specific addresses. If a bit position matches, the packet is received. The following is a list of reserved bit positions.

- 0xC00000000001 - Active Monitor
- 0xC00000000002 - Ring Parameter Server
- 0xC00000000008 - Ring Error Monitor
- 0xC00000000010 - Configuration Report Server
- 0xC00000000080 - NetBios
- 0xC00000000100 - Bridge
- 0xC00000002000 - LAN Manager

When Multicast Addresses are added using the ADDMULTI dlpi message, all the bits are or'ed together with all the other Multicast Addresses currently being received. Then, the Active Monitor, Ring Parameter Server, Ring Error Monitor, Configuration Report Server, and Bridge bits are masked out to prevent reception of those packets. Only 27 unique groups are supported through this scheme, so if you want to receive NetBios packets, for example, an Add Multicast operation could be invoked with the address 0xC00000000080.

## The `trkstat` Data Structure

This section describes how to obtain Token Ring Network Statistical information about the Token Ring LAN Network operation that is maintained in the device driver.

A set of Token Ring network statistical counters regarding the network operation can be programmatically accessed through the Solaris `trkstat` command. The data structure with which the `tr` device driver keeps those statistics can now be obtained. Sun Microsystems, Inc. reserves the right to change the following data structure, in the future, without notice.

```
struct trkstat {
    struct kstat_named    trs_ipackets;
    struct kstat_named    trs_ierrors;
    struct kstat_named    trs_opackets;
    struct kstat_named    trs_oerrors;
    struct kstat_named    trs_inits;
    struct kstat_named    trs_notmds;
    struct kstat_named    trs_notbufs;
    struct kstat_named    trs_norbufs;
    struct kstat_named    trs_nocanput;
    struct kstat_named    trs_allocbfail;
    struct kstat_named    trs_sralloc;
    struct kstat_named    trs_srfree;
    struct kstat_named    trs_lineerrors;
    struct kstat_named    trs_bursterrors;
    struct kstat_named    trs_arifcierrors;
    struct kstat_named    trs_lostframeerrors;
    struct kstat_named    trs_rxcongestion;
    struct kstat_named    trs_frmcopiederrors;
    struct kstat_named    trs_tokenerrors;
    struct kstat_named    trs_dmabusererrors;
    struct kstat_named    trs_rcvbytes;
    struct kstat_named    trs_xmtbytes;
};
```

**TABLE A-2** trkstat Statistical Counters

Counter	Definition
trs_ipackets	This counter is incremented when a frame is received.
trs_ierrors	This counter is incremented when a frame received is in error.
trs_opackets	This counter is incremented when a frame is transmitted.
trs_oerrors	This counter is incremented when a frame transmitted is in error.
trs_init	Not implemented.
trs_notmds	This counter is incremented when the driver fails to allocate transmit message descriptor (TMD).
trs_notbufs	This counter is incremented when the driver fails to allocate transmit buffer for an outgoing frame.
trs_norbufs	Not implemented.
trs_nocanput	This counter is incremented when an incoming frame cannot be put on the queue of the upper layer protocol component.
trs_allocbfail	This counter is incremented when the driver cannot allocate buffers from the Kernel space.
trs_sralloc	This counter is incremented when a route is created and is entered in the Source Route Table.
trs_srfree	This counter is incremented when a route is deleted from the Source Route Table.
trs_lineerrors	<p>This counter is incremented whenever:</p> <ol style="list-style-type: none"><li>1. A frame is repeated, and</li><li>2. The Error Detected Indicator (EDI) is zero in the incoming frame, and</li><li>3. At least one of the following conditions occurs:<ul style="list-style-type: none"><li>- A code violation between the starting delimited and the ending delimiter of the frame.</li><li>- A code violation in a token.</li><li>- A Frame Check Sequence (FCS) error.</li></ul></li></ol> <p>When the line error is incremented, the EDI of the frame is set to one so that no further adapters count the error.</p>
trs_bursterrors	This counter is incremented when the adapter detects the absence of transitions for five half-bit times between SDEL and EDEL, or EDEL and SDEL. Only one adapter detects the burst five conditions because the adapter that detects a burst four condition directs its transmitter to transmit idles if the burst five condition is detected.

**TABLE A-2** trkstat Statistical Counters

Counter	Definition
trs_arifcierrors	This counter is incremented when an adapter receives an Active Monitor Present (AMP) MAC frame with the ARI/FCI bits equal to zero and a Standby Monitor Present (SMP) MAC frame with the ARI/FCI bits equal to zero, or more than one SMP MAC frame with the ARI/FCI bits equal to zero, without receiving an intervening AMP MAC frame. This condition indicates that the upstream neighbor is unable to set the ARI/FCI bits in a frame that it has copied.
trs_lostframeerrors	This counter is incremented when an adapter is in transmit (stripping) mode and fails to receive the end of the frame it transmitted.
trs_rxcongestion	This counter is incremented when an adapter in the repeat mode recognizes a frame addressed to it but has no buffer space available to copy the frame.
trs_frmcopiederrors	The token error counter is active only in the active monitor station. It is incremented when the active monitor detects an error in a passing token frame.
trs_dmabuserrors	The DMA Bus error counter counts the occurrences of DMA bus errors on the adapter card (not on the system SBus).
trs_rcvbytes	This indicates the total number of bytes received.
trs_xmtbytes	This indicates the total number of bytes transmitted.

# Token Ring Architecture Overview

---

This appendix provides a general overview of the token ring architecture.

The International Standards Organization (ISO) has established a seven-layer model that provides a standard reference for intercommunication of computer systems through a network by using common protocols. The seven-layer model has become an international standard.

Each layer of the model defines a specific section of the total architecture. The layers of the Open Systems Interconnection (OSI) are, from top to bottom:

**TABLE B-1** Seven Layers of Open Systems Interconnection (OSI)

Layer	Description
Application	Gives the user access to all the lower layer functions and deals with the exchanges between user applications.
Presentation	Represents user data including required translations and code conversions.
Session	Provides the mechanisms for organizing and structuring the interactions between applications and devices.
Transport	Provides transparent end-to-end data transfer relying on lower-level functions for handling any peculiarities of the actual transfer medium.
Network	Contains the definitions that provide the means for establishing connections between networks. The definitions include procedures for the operational control of inter-network communications and the routing of information through multiple networks.
Data Link	Provides the functional and procedural means for transferring data between network entities as well as the means for detecting and correcting errors that occur on the network.
Physical	Provides the mechanical, electrical, and functional and procedural standards for accessing the physical medium.

The Institute of Electrical and Electronic Engineers Computer Society has also drafted standards for local area networks. A reference model similar to that of the OSI model was created. Two layers in the IEEE model correspond to the two lower levels of the OSI model. In the IEEE model, however, the Data Link layer is further divided into two sub-layers: the Logical Link Control (LLC) sub-layer and the Medium Access Control (MAC) sub-layer.

Various media access control techniques have been created and have become standards. Among them is the Token Ring Standard -IEEE 802.5.

---

## The Data Link Control Layer

Standards for local area networks are based on the two lower layers, Data Link Control and Physical Control. As a result, token ring networks can co-exist with SNA systems. In an SNA mode, the Data Link Control layer is replaced by the LAN Data Link Control layer (which is further divided into the Logical Link Control and Medium Access Control sub-layers) and the LAN Physical Control layer.

Each IBM node in an IBM token ring network contains a Data Link Control layer called DLC.LAN. DLC.LAN consists of a manager function called DLC.LAN.MGR, a Logical Link Control sub-layer which includes one or more link stations, an access channel control, and a user datagram service and a Medium Access Control (MAC) sub-layer which includes one medium access control.

A link is a logical connection between two link stations providing data transfer between two nodes. A node is either of the endpoints of the link. A link station is a protocol machine that manages the elements of procedures required for data exchange and schedules data transfer over the link.

### DLC.LAN.MGR

DLC.LAN.MGR includes both the LLC and MAC sub-layers. It also provides management functions for both. DLC.LAN.MGR supervises the operation of the DLC.LAN and directs the flow of information through the LLC and MAC sub-layers. It also controls link activation and the attachment of ring stations to the ring and removal of ring stations from the ring.

DLC.LAN.MGR acts as the interface between DLC.LAN and the Physical Unit. It handles the records from the Physical Unit that require action by the DLC.LAN.MGR. It also passes other records from the Physical Unit to the appropriate link stations.



---

## The Logical Link (LLC) Sub-layer

The LLC sub-layer provides sequential, connection-oriented data transfer and non-sequential, connectionless data transfer. The link stations provide sequential connection-oriented data transfer and error recovery for one or more links.

The Path Control layer passes basic transmission units (BTUs) to the link stations which in turn add appropriate control information. On each link, the local and remote stations use the HDLC asynchronous balanced mode of operation to keep the LLC Protocol Data Units (LPDUs) in sequence, and to detect and correct LPDUs that are out of sequence. Link stations pass the BTU and control field to the access channel control which multiplexes message units flowing between link stations and DLC.LAN.MGR and the MAC sub-layer. The access channel control builds LPDUs from the information in its internal table for that link station. It then transmits the LPDUs to the MAC sub-layer. The access channel control also routes the LPDUs it receives from the MAC sub-layer to the appropriate link station or to the DLC.LAN.MGR.

Connectionless data transfer is provided by the user datagram service where data is sent and received without any correlation to previous or subsequent data and without the need for establishment of a data link connection.

A higher-layer protocol passes data to be transferred and the source and destination addressing information to the user datagram service which adds appropriate control information. The user datagram service passes the addressing information, control field, and data to the access channel control. It does not provide acknowledgment of data or flow control or error recovery procedures.

---

## The Medium Access Control (MAC) Sub-layer

The MAC sub-layer controls the routing of information between the Physical layer and the LLC sub-layer. The MAC sub-layer provides:

- Address Recognition - For initiating copying of a frame based on the destination address in the physical header. Each ring station must be able to recognize its own MAC address and an all-stations broadcast address. Additionally, ring stations should be able to recognize one or more group addresses.
- Frame Copying - For copying a frame off the ring.
- Frame Control Recognition - For determining the frame format and the type.
- Delimiting Frames - For delimiting the start and end of a frame.

- Frame Status Generation and Verification - For providing and verifying additional information in each frame to detect transmission errors.
- Priority Management - For gaining access to the transmission medium based on priority.
- Routing - For determining which function in the node should process the frame.
- Timing - For providing timers required by the token and MAC management protocols.
- Token Management - For gaining access to the physical transmission medium including appropriate supervision protocols for handling errors.

---

## The Physical Layer

Each node in a token ring network contains a physical layer. The Physical layer provides attachment to the transmission medium and contains the cable and the circuit switches used to configure the equipment.

The primary function of the Physical layer is to encode, transmit, and recognize bits, code violations, and signal losses.

The Physical layer also provides master timing and voltage control functions.

---

## Basic Operating Concepts

This section presents an overview of the terms and procedures that are basic to the operation of token ring networks. Details are given at the MAC sub-layer level because this level provides the protocols unique to token ring operation.

### The Ring

In a token ring network, a ring consists of ring stations and the transmission medium that interconnects them. A ring station consists of functions that allow a device to connect to the ring and use the link access protocols. See chapter 1, for diagrams illustrate a simple ring configurations.

A ring station transfers data in the form of tokens and frames to the ring where the data travels sequentially from ring station to ring station. The tokens and frames are identified by unique formats that begin with a Starting Delimiter and end with an Ending Delimiter. The actual flow path has a definite direction that is determined by the physical location of the ring stations on the ring. Each ring station repeats the

data it receives, checking it for errors, and copying it if it is the destination (the intended recipient). When the data returns to the originating ring station, the station removes it from the ring.

The protocols provide error detection and recovery functions, such as acknowledgment, beaconing, retransmission, and timeouts.

Each station can serve one or more attached devices, such as terminals or printers, and allow them to communicate with other attached devices on the ring.

A delay of no less than 24 bits is required on the ring to allow the 24-bit token to circle the ring successfully. The total delay on the ring consists of the ring station delays plus the propagation delay caused by the transmission medium.

The token ring network requires addressing functions so that communication between any two rings can be uniquely identified. Addressing is independent of the physical configuration.

The token ring network also requires data checking functions to preserve the integrity of its access control and ring management transmissions.

---

## Multiple Ring Connections

The token ring architecture supports multiple ring connections using bridges, through which data can be transmitted from a ring station on one ring to a ring station on any other ring. Refer to Appendix C for more information.

The bridge also acts as a ring station on any ring to which it is attached. The bridge copies frames destined for other rings and transmits frames from other rings that are destined for the local ring, or for rings beyond it. The destination is based on addressing information. The routing information field of the frame determines whether or not the bridge copies the frame.

## MAC Sub-layer Operating Modes

This section describes the normal token protocols used to transmit a frame on a token ring network.

- *Normal Repeat Mode*—In this mode, a ring station checks the data in the tokens and frames it receives and sets the Error-Detected, Address-Recognized and Frame-Copied bits, as appropriate, as it repeats the token or frame.
- *Transmit Pending Mode*—This is the mode in which a ring station with data to transmit waits for a free (available) token.

A ring station may receive a token of usable priority and change it to a frame and append data and control information to it. At this point, the station enters the Normal Transmit Mode and transmits the information. If a ring station discovers that it has incorrectly identified a frame as a token, it transmits an Abort sequence and returns to the Transmit Pending Mode. After a station has sent data, it starts its physical trailer (the ending delimiter and frame status field) timer. If the ring station does not receive its physical header by the time it transmits the frame status field, it transmits idles and waits for the physical header to return. If the physical header timer expires and the physical header has not returned, the ring station returns to the Normal Repeat Mode without originating a token.

When the physical header returns, the ring station compares the returned Source Address with the Source Address it transmitted. If the two addresses are identical, the ring station transmits a token followed by idles. Otherwise, the ring station continues to wait for its transmitted physical header. The ring station transmits idles until it receives its transmitted physical trailer and has completely removed its frame from the ring. It then returns to Normal Repeat Mode. If the physical header is never returned, the ring station transmits idles until its physical trailer timer expires, at which time it enters the Normal Repeat Mode.

---

## The Active Monitor

The Active Monitor function is part of the architecture of every ring station. One ring station on each ring is designated as the Active Monitor. It provides token monitoring as well as various other functions. Any operating ring station can be assigned as the Active Monitor. Other ring stations act as standby monitors in case the currently designated Active Monitor fails.

The Active Monitor resolves certain error conditions on the ring such as: lost tokens, frames and priority tokens that circle the ring more than once, more than one Active Monitor on the same ring at the same time, a short ring condition in which the bit delay is too short to hold a token.

## Maintaining the Master Clock

The Active Monitor maintains the ring's master clock which controls timing and ensures that all other clocks on the ring are synchronized.

## Ensuring Proper Ring Delay

The Active Monitor ensures that a token can be completely transmitted before returning to the originating station. It ensures sufficient delay by inserting a 24-bit delay (the length of a token) into the ring.

## Initiating Neighbor Notification

The Active Monitor periodically broadcasts the Active Monitor Present MAC frame to all ring stations on its ring. Each ring station then acquires the address of its Nearest Active Upstream Neighbor (NAUN)

Neighbor Notification begins when the Active Monitor broadcasts an Active Monitor Present MAC frame to all the ring stations on its ring and resets the Neighbor Notification Complete flag. The first ring station that receives the Active Monitor Present MAC frame copies it and sets Address Recognized (A bit) and Frame Copied (C bit). The ring station then saves the source address as its NAUN address and transmits a Standby Monitor Present MAC frame with the A and C bits set, to all other ring stations on the ring.

The next ring station downstream disregards the Active Monitor Present MAC frame because the A and C bits are already set. The ring station then copies its NAUN address from the Standby Monitor Present MAC frame, sets the A and C bits, and transmits its own Standby Monitor Present MAC frame.

In this manner, Neighbor Notification proceeds around the ring with each ring station receiving and transmitting Standby Monitor Present MAC frames until the Active Monitor copies the last Standby Monitor Present MAC frame whose the A and C bits are set. The Active Monitor then sets the Neighbor Notification Complete flag which indicates that the process has been successfully completed.

## Monitoring Neighbor Notification

Any time during the Neighbor Notification cycle an exception event could occur and impede the notification process as it makes its way around the ring.

The Neighbor Notification timer has a long time-out. When it expires, the Active Monitor transmits an Active Monitor Present MAC frame. If the previous Neighbor Notification is not complete, the Active Monitor also transmits a Report Neighbor Notification Incomplete MAC frame to the Ring Error Monitor.

The Active Monitor uses its receive notification timer to ensure that during Neighbor Notification each Active Monitor Present MAC frame circles the entire ring. If the timer expires, the Active Monitor Present MAC frame did not circle the ring and the Active Monitor initiates token claiming. Token claiming is a process that allows a ring station to elect an Active Monitor.

If the Active Monitor receives a Standby Monitor Present MAC frame after Neighbor Notification has successfully completed, the frame is ignored.

If the Active Monitor copies an Active Monitor Present MAC frame that contains a source address other than its own, it means that an Active Monitor is already on the ring and that the Active Monitor that copied the frame can now deactivate its Active Monitor functions.

A hard error can cause the suspension of the token protocol and as a result cause the suspension of the Neighbor Notification process. The Active Monitor transmits a Report Neighbor Notification Incomplete MAC frame to the Ring Error Monitor. If the ring is busy transmitting data, especially priority traffic,

Neighbor Notification could be delayed or interrupted. The Active Monitor transmits a Report Neighbor Notification Incomplete MAC frame to the Ring Error Monitor.

## Monitoring Token and Frame Transmission

The monitor bit in the access control field is reset in every newly transmitted token or frame. When the Active Monitor repeats a frame or non-zero priority token, it sets the monitor bit. If the bit is already set, the Active Monitor assumes that the token or frame has already circled the ring. The Active Monitor then purges the ring and releases a new token.

## Detecting Lost Tokens and Frames

The Any Token Timer has a short time-out but it still exceeds the time for the longest possible frame to circle the ring. The Active Monitor restarts this timer whenever it repeats a Starting Delimiter. If this timer expires, the Active Monitor assumes that the token or frame was lost on the ring. The Active Monitor purges the ring and releases a new token.

## Purging the Ring

The Active Monitor broadcasts a Ring Purge MAC frame to all ring stations on the ring before originating a new token. Receipt of the returned frame indicates to the Active Monitor that a frame can circle the ring without incident. The Active Monitor then restarts the token protocol.

The Ring Purge MAC frame resets ring stations to Normal Repeat Mode and cancels or restarts appropriate timers. The Active Monitor ignores the value of the monitor bit in returning Ring Purge MAC frames. The Active Monitor copies the reservation bits of the last returning Ring Purge MAC frame into the new token it originates.

An Active Monitor that is interrupted by a Ring Purge MAC frame that it did not originate, assumes itself to be a duplicate monitor and becomes a standby monitor. (It removes its 24-bit delay and master clock functions.)

---

## Attaching the Ring

This section describes the various phases a ring station must successfully complete to attach and function on a token ring network. The only assumption about the ring station is that it has a pre-assigned address.

### Phase 0: Lobe Test and Physical Attachment

Lobe testing is part of the initialization process that is performed before the ring station has attached to the ring. A lobe is the section of cable that attaches a device to an access unit. The access unit allows the devices to access the ring from a central point.

A lobe test consists of sending a series of Lobe Test MAC frames only on the ring station's lobe to ensure that there is no fault on the lobe itself. If the frame passes through the lobe successfully, the ring station attaches to the ring. Otherwise, the ring terminates the attachment process with an error. There is no wait for a token because the Lobe Test MAC frames are only sent on the ring station's lobe.

After the power-on diagnostics and initialization, a device containing a ring station is physically attached to the ring and proceeds to Phase 1.

## Phase 1: Monitor Check

The ring station starts an attach timer and the following events occur:

If the ring station receives an Active Monitor Present, Standby Monitor Present, or Ring Purge MAC frame before the timer expires, it assumes that an Active Monitor is present on the ring and proceeds to Phase 2.

If the ring station does not receive any of the above mentioned frames before the timer expires, either it is the first station on the ring or no Active Monitor exists on the ring. In either case, the ring station initiates token claiming.

A ring station does not activate its standby monitor functions until it has attached to the ring.

## Phase 2: Duplicate Address Check

In this phase the ring station checks for the presence, on its ring or another ring station, of the same address. The check is performed by sending a Duplicate Address Test MAC frame. If a duplicate address is found, the ring station removes itself from the ring.

## Phase 3: Participate in Neighbor Notification

The ring station participates in Neighbor Notification, learns its NAUN, and identifies itself to its nearest downstream neighbor.

If the ring station detects beaconing or the need for beaconing before it has completed Neighbor Notification, it removes itself from the ring and indicates the need for beaconing (described under “Hard Error Detection and Reporting”). After Neighbor Notification is complete, the detection of beaconing, or the need for beaconing, causes the ring station to begin beaconing as though it had completed attaching to the ring. Attachment continues after the ring recovers.

## Phase 4: Request Parameters

To maintain consistency across the network, various ring parameters must be used by all ring stations. Changeable parameters include: Physical Location, Soft Error Report Timer Value, and Ring Number. The Ring Parameter Server acts as focus to ensure that ring stations on the ring have the same Soft Error Report Timer and Ring Numbers.



The ring station requests changed operational parameters from a Ring Parameter Server, if one is present. If one is not present, default values are used.

If this information is incorrect or threatens ring integrity, the Ring Parameter Server can notify the LAN manager, which can issue a Remove Ring Station MAC frame, forcing the ring station to remove itself from the ring.

---

## Soft Error Detection and Reporting

Soft errors are usually intermittent faults that temporarily disrupt normal operation of the token ring network. Soft errors are normally tolerated by error recovery procedures. Most soft errors are indicated by architectural inconsistencies, such as cyclic redundancy checks or time-outs, in received or repeated frames or by a ring station's inability to process received frames. If soft errors result in degradation of the ring's performance, the LAN manager can reconfigure the ring to bypass the faulty node.

Each ring station maintains a soft error report timer and a set of counters to measure how often the most critical errors occur. When the soft error report timer expires, the ring station transmits a Report Soft Error MAC frame to the Ring Error Monitor. After reporting the soft error, the ring station resets the error counters.

The Report Soft Error MAC frame reports the number of errors detected since the previous report. The report identifies the transmitting ring station's address and NAUN. It includes all the errors and their values.

---

## Hard Error Detection and Reporting

Hard errors are usually permanent faults in equipment that cause the ring to stop functioning with the normal token ring network protocols. A ring station downstream from the hard fault recognizes a hard error at the receiver side of its attachment. The ring must be reconfigured to bypass the error. The reconfiguration restores the ring to the normal operational state. Repairs might be required to restore the ring to full operation.

When a ring station detects a hard error, it transmits a Beacon MAC frame, with an All-Stations Address to its ring only, until either its input signal is restored or until it removes itself from the ring. This is called beaconing. All other ring stations that receive the Beacon MAC frame enter the Beacon Repeat Mode.

The Beacon MAC frame identifies the beaconing ring station's NAUN and the type of error detected. When the beaconing ring station's NAUN has copied eight Beacon MAC frames, the NAUN removes itself from the ring. It then tests itself and its lobe using Duplicate Address Test MAC frames. If the test is successful, the NAUN re-attaches itself to the ring.

If the ring does not recover after the NAUN tests itself, the beaconing ring station removes itself from the ring and follows the same test sequence described above, for its NAUN.

If the ring still does not recover after both the beaconing station and its NAUN have tested, the error requires manual intervention.

If the ring still does not recover after both the beaconing station and its NAUN have tested, the error requires manual intervention.

---

## Addressing and Routing Concepts

Each transmitted frame on a token ring network has Destination Address and a Source Address. Their purposes are to identify the receiving and sending ring stations, respectively.

In a multiple-ring environment in which two or more rings are connected by bridges, the destination addresses for frames originating in one ring station and directed to one or more ring stations on other rings, can be created using a source routing technique. The route is determined at session connection time when an initiating ring starts the process of collecting routing information that will be included in subsequent frames. This method eliminates the need to maintain routing tables at predefined nodes on the ring.

### Source Routing

Source routing is the method used within the token ring network architecture to route frames through a multiple-ring local area network. A route is the path taken by a frame as it travels through a network from the originating station to a destination station. Source routing does not require centralized routing tables because each frame carries information about the route it should follow. The routing information is obtained using a search process that originates at the source ring station and fans out through the network, to the destination ring station. The search is initiated by executing an All-Rings Broadcast, (discussed later in this section).

Routing information is acquired using the TEST, or XID, LPDU commands. The originating station sends a TEST, or XID, command LPDU to all rings on the network. The command frame passes through all the interconnected rings by way of bridge stations, searching for the Destination Address. During the search process, the command frame accumulates routing information as it passes through the bridges.

As the LPDU passes through the multiple-ring network, copies are created, and all continue to search for the Destination Address. If more than one route to the Destination Address exists, then more than one LPDU will reach the destination ring station. As the destination ring station receives each LPDU, it returns the acquired routing information to the originating ring station in a TEST, or XID, response LPDU which follows the original route in a reverse manner.

If more than one route to the Destination Address is found, all of them are returned to the originating ring station which then chooses a route. The originating ring station learns the preferred route when it receives the first response from the destination ring station. The originating ring station then uses the preferred route for any subsequent transmissions to the destination ring station.

In addition to the Destination Address and the Source Address carried in each frame sent in a multiple-ring network, an ordered list of the ring and bridge numbers, through which the frame must pass to reach the Destination Address, is sent. Each bridge is responsible for adding this routing information to the routing information filed in All-Rings Broadcast frames. Each bridge must also check the routing information in non-broadcast frames and make forwarding decisions based upon what it finds.

## Addresses

The token ring network associates a ring station, or a group of ring stations, with a unique MAC Sub-Layer Address. This enables any ring station to attach to the token ring network.

### Individual and Group Addresses

An Individual Address Identifies a particular ring station on the token ring network.

A Group Address identifies a group of destination ring stations on the token ring network.

## Universal and Local Address Administration

Universal administration means that all individual addresses are assigned, administered, and guaranteed by the IEEE to be unique across all local area networks by the IEEE. This method eliminates customer involvement in the administration of individual addresses which in turn eliminates the need for site address administrators and address administration programs.

Local administration means that all individual addresses are administered by someone other than the IEEE. Local addresses must still be unique within the token ring network where they occur.

## Null Addresses

An address of all zeroes is considered a Null Address. A frame with a Null Address is not addressed to any ring station. It can be sent but not received. When a frame with a Null Address returns to its originating ring station, its data is stripped and a new token is issued.

## All-Stations Broadcast Addresses

An All-Stations Broadcast Address means the frame is sent to all stations on a given ring, or interconnected rings. All stations must be able to recognize All-Stations Broadcast Address. Whether or not the frame leaves the source ring, is determined by the routing information in the frame.

## Functional Addresses

IBM token ring network architecture provides bit-specific Functional Addresses for various uses. Ring stations use Functional Address masks to identify these functions. The masks consist of the sum of the bits for the functions for a particular ring station.

All Functional Addresses are locally-administered Group Addresses. The architecture allows up to 31 functional addresses to be specified. Refer to IBM bridge documentation for actual address definitions.

## Broadcast Terminology

To broadcast a frame is to send it to more than one ring station, or to more than one ring. A frame is also broadcast when a specific Destination Address is known but the location and route to the destination is not.

All-Rings Broadcast is the sending of a frame to all interconnected rings on a local area network. All-Rings Broadcast is indicated by frame with the broadcast bit in the routing information set to "1". The address of particular ring stations on those rings is determined by the Destination Address field.

An All-Rings Broadcast is independent of an All-Stations Broadcast which is the sending of a frame to all stations of a ring.

Within the bridge concept, there is also a Limited Broadcast Frame which refers to the forwarding of specially designed broadcast frames by bridges that have limited broadcast ability enabled.



## Network Configuration Planning

---

Planning your network configuration is an important prerequisite to installing SunTRI/S. Read this appendix thoroughly and plan your configuration before you continue the installation

The performance of your token ring network depends not just on the SunTRI/S itself, but on the overall network configuration and individual system administration. Much of this appendix describes how to configure your network to maximize performance. Topics include:

- Token Ring Network Topologies
- Establishing an all-token ring network
- Incorporating token ring into an existing Ethernet network
- Setting up token ring-to-Ethernet routers
- Improving distributed applications
- Selecting SunTRI/S adapter's MAC address

---

## Token Ring Network Topologies

Factors that influence the selection of a network topology for a particular installation, include:

- Number of Stations
- Type of Applications
- Affinity Groups
- Backup Paths
- Performance
- Capacity
- Reliability
- Availability
- Expected Growth

---

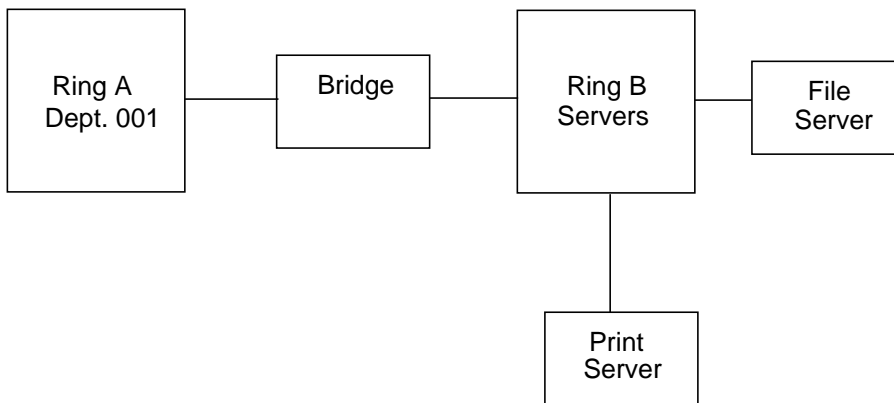
## General Guidelines

When the number of workstations is small, less than 40, the distribution of workstations is determined primarily by physical considerations. Selecting a particular physical layout is determined by the locations of various departmental groups, or groups, of users performing related tasks.

The first decision to be made concerns the number of rings to have on the token ring network. The answer depends upon various factors that are examined in the following discussion.

## Performance

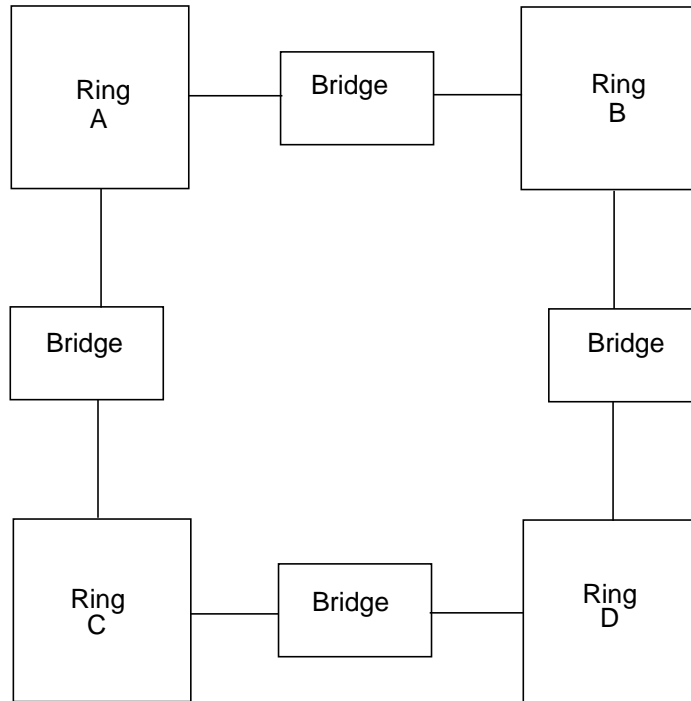
Depending upon the applications running on each workstation, the performance might be improved by using a multiple-ring network. Performance can often be improved by choosing a two-ring network with the workstation on one ring and the servers on the second ring, as illustrated in the following diagram.





# Reliability

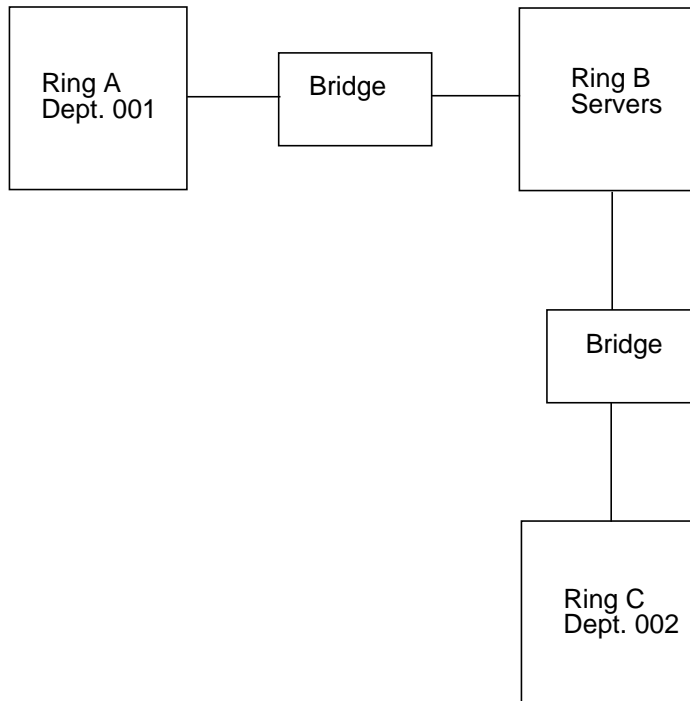
A network configuration of multiple rings might permit easier fault isolation. Within a multiple-ring network, a transient beaconing condition on one ring does not preclude the use of workstations on another ring, though the beaconing ring is inoperable for data transfer. For example, in the following diagram, a four-ring configuration is illustrated. If ring C is beaconing, workstations attached to rings A, B, and D can continue exchanging information with no problem. (Beaconing is sent from a ring station indicating that a serious ring problem exists.)



# Affinity Groups

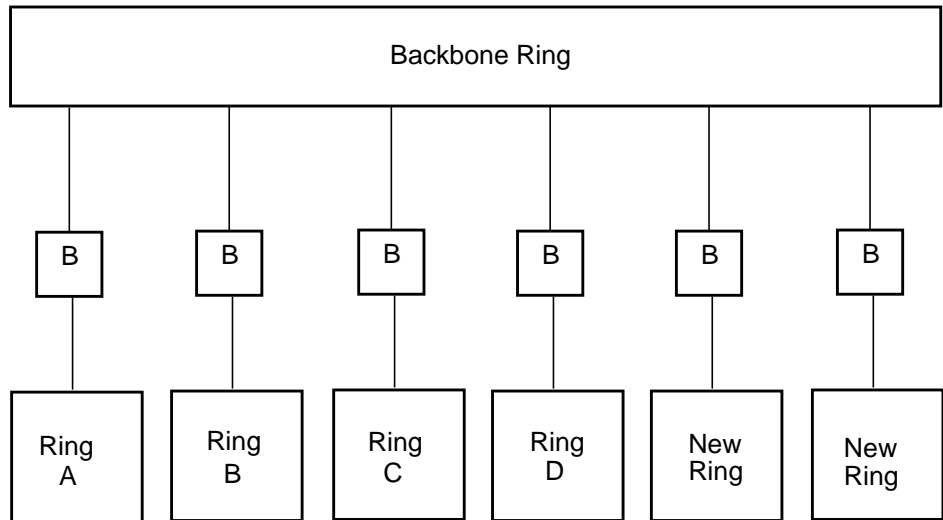
If several end-users perform related tasks on the network, and have little information interchange with groups of other end-users, each of the affinity groups can be attached to different rings within a network. This configuration, as illustrated

below, decreases the network workload and improves overall performance. In this three-ring configuration, Ring A and Ring C are two separate departments with little inter-communication. Ring B contains the servers common to Rings A and B.



# Growth

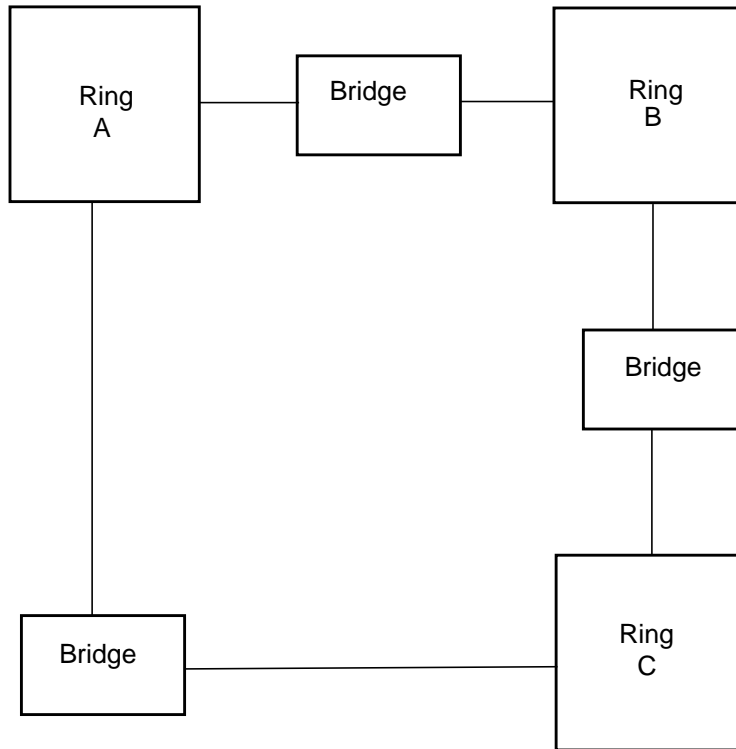
In the course of designing your network, the probability of future expansion should also be considered. Within a multiple-ring network, the addition of new workstations, or new rings, can be performed with a minimum of impact on network activity. The following diagram shows a backbone configuration in which expansion can be added easily.



In the configuration illustrated above, new rings can be added to the Backbone Ring by adding one bridge for each new ring that is added.

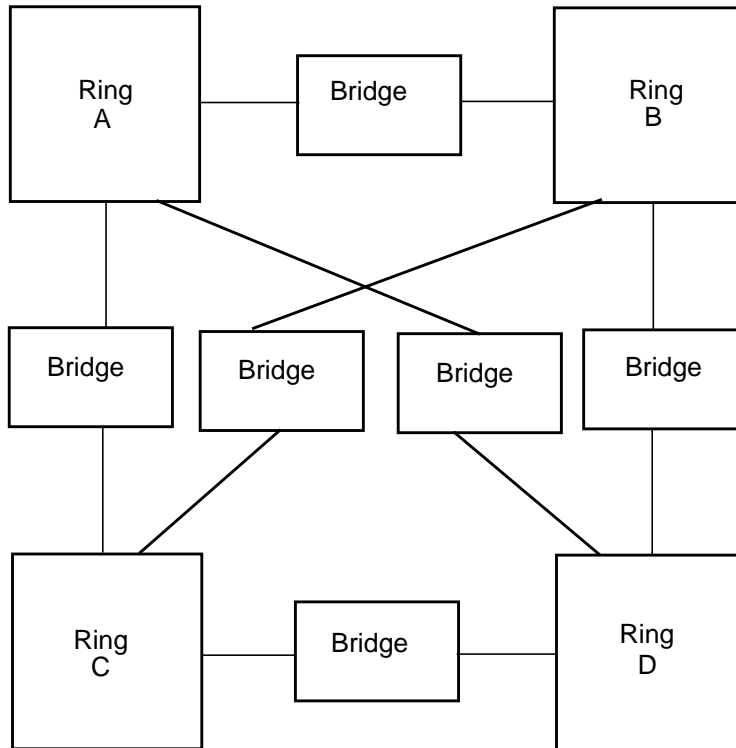
Having decided how many rings to have within the token ring network, the next logical subject deals with the actual inter-connection of multiple rings within a token ring network.

The loop configuration illustrated below, has several unique characteristics. For example, the information exchange between any two workstations on different rings always has only one bridge to cross. This reduces bridge overhead. Further, if an error condition occurs on one bridge and it goes out of operation, an alternative path still exists.



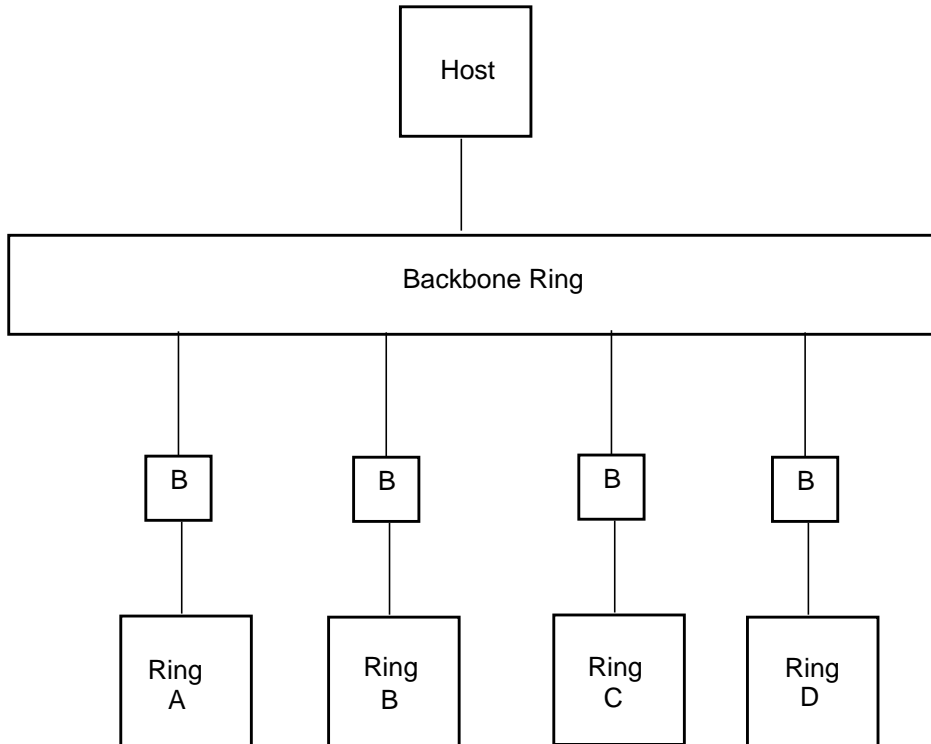
A disadvantage to this loop configuration is that each ring should be bridged to every ring on the network to for maximum performance and connectivity.

Additionally, loop configurations are difficult and expensive to expand. To add one more ring to the loop configuration shown above, would require the addition of three more bridges. See the diagram below.



From this discussion, you can see that if you want to maintain the same loop configuration and add more rings, the number of bridges increases significantly.

For large networks with high growth and host connections, the backbone configuration is more practical than the loop configuration. See the illustration, below for a backbone configuration.



## Backbone Configuration Strengths

The method for interconnecting the rings simplifies the physical attachment or relocation of workstations and rings.

Adding a new ring merely requires the addition of one new bridge.

The hosts attached to the backbone ring as are various file, compute, and print servers. This means that rings with low server or host access demands will not load the backbone ring. Additionally, the host and servers are isolated by the bridges from possible beaconing conditions on the other rings.

## Backbone Configuration Weaknesses

The exchange of information between lower rings requires crossing two bridges.

The lower-level rings rely on one bridge for connectivity to the host, servers, and to other workstations attached to different rings. So, if one bridge fails, the lower-level ring which it interconnects will lose host and server access.

Lower reliability results because any two workstations have one possible path through which the information can flow. If the path is not available, communication is lost.

Other possible configurations exist and can be evaluated. Actual requirements of your network determine the configuration you design.

## Impact on Performance

Bridge processing within multiple-ring token ring networks adds overhead to network throughput. Factors that determine the amount of overhead are listed below.

**TABLE C-1** Overhead Determining Factors

Factor	Description
Throughput	An increase in frame size to a certain value, increases bridge throughput.
Hop Delay	This is the elapsed time between the end of the receive stage for a frame entering the bridge and the end of the transmit process for the same frame leaving the bridge. The delay increases as frame size and ring load increase.
Application Delay	Some applications create greater degradation due to bridge processing than others. Applications that involve disk access or host delays will not experience much degradation. Fast applications, such as program loads or memory-to-memory copies, might notice greater impact on performance when the frames cross heavily used bridges.
Frame Flow	Through a bridge passes with higher priority than normal ring station processing because the architecture assigns a higher priority to it.

In general, during normal workload conditions, bridge processing does not cause a major slowdown of activity in a multiple-ring network.

---

# Token Ring Bridge Concept

A bridge is a device that forwards frames from one ring to another within a local area network. It connects multiple rings into one logical network and appears to each ring as a standard ring station. It performs frame checking and participates in token claiming and neighbor identification like any other ring station on the ring.

The bridge's operation on a token ring network is implemented at the Medium Access Control (MAC) sub-layer level. A bridged network is considered to be one large IP, or Network, layer. Thus, all bridged rings would have the same internetwork number. If another ring station on the ring is communicating to the bridge, rather than through it, the bridge is addressed in the same manner as any other ring station - at the MAC sub-layer through the Destination Address field.

Logical link control (LLC) procedures are implemented between pairs of source and destination ring stations. They are not terminated, examined, or affected by intermediate bridges.

Each bridge is assigned a pair of 2-byte ring numbers that correspond to the rings to which it is attached. The Segment Number, also known as the ring address, is used only for routing information and should not be confused with the MAC-level ring station address of the bridge.

During initialization, the bridge requests configuration parameters using the Request Parameter MAC frame. In addition to the MAC configuration parameters, default parameters are used unless their values are modified by an operator.



---

**Caution** – Bridges should never be placed in parallel with routers or in a situation where they will interconnect two distinct IP networks already interconnected by a router. You must treat bridged rings as a single subnet for IP addressing purposes.

---

## Source Routing

Each transmitted frame on a token ring network contains a Source Address and a Destination Address. These addresses identify the sending and receiving ring stations. In a multiple-ring environment with two or more rings connected by bridges, the Destination Addresses for frames originating in one ring station and destined to one or more stations in other rings, are built using a unique Source Routing technique. The route is found through bridge stations at session connection time when the initiating station starts collecting routing information that will be included in subsequent frames. This method eliminates the need for maintaining routing tables at predefined nodes on the ring.



Source Routing functionality lies within the Data Link layer of the OSI reference model. IP routing (described on the following pages) lies within the Network layer. The two types of routing are functionally exclusive.

You need not select Source Routing if you are installing your SPARCstation on an isolated token ring network, or on one with interfaces to other Ethernet networks and intend to communicate only with other hosts using TCP/IP. A SPARCstation with multiple network interfaces can route IP packets, but not source-route packets, between the networks.

A more detailed discussion of Source Routing can be found in Appendix A “Advanced Topics for Programmers” and in Appendix B “Token Ring Architecture Overview”.

## Setting Up Routers

Whenever you install the SunTRI/S in a machine with an existing Ethernet connection, that machine becomes a router serving both the token ring and Ethernet networks. The router is known to the Ethernet network by its Ethernet IP address and hostname, and to the token ring network by its token ring IP address and hostname.

In general, stations in an Internet can be reached via multiple routes. When the Solaris operating environment is booted, the route daemon (*in.routed*) sends out packets over both the Ethernet and token ring. The route daemon determines the optimum path between stations by relative hop count and not by the speed of the interface. Therefore, in cases where the hop counts are equal, the use of the token ring is not guaranteed.

Users can guarantee the use of token ring for network commands by specifying the target system’s token ring IP name.

## Selecting the MAC Address

Each station on the network is identified by its unique MAC address. SunTRI/S supports IEEE globally-assigned 48-bit MAC addresses. You must select the MAC address for a new station *before* it is brought up on the ring. The following paragraphs describe how to select from the available MAC addresses for diskful stations.

---

**Note** – Once a new station starts transactions on the network, the Address Resolution Protocol (ARP) tables on all other stations are updated to include the new station. After this occurs, making changes is more difficult because you would need to delete ARP entries on every active station in the ring.

---

The choices of MAC address include:

- The MAC address that is resident in the host IDPROM (default). This 48-bit value is administered globally by the IEEE. It is the same IDPROM value used by your SPARCstation's on-board ethernet controller.
- A locally-administered MAC address set by ifconfig. This 48-bit value is administered by the network or system administrator.

In most cases, you will use the default—the MAC address resident in the host IDPROM. The advantages to using the host IDPROM MAC address are that:

- You do not have to change any settings for the SunTRI/S adapter.
- You conform to Sun's philosophy of using the same host IDPROM value for each LAN attachment that supports 48-bit MAC addresses.
- You avoid conflicts if you use other network protocol stacks that require the host IDPROM MAC address (for example, the XNS networking protocols).
- You need not assure MAC address uniqueness.

# Index

---

## SYMBOLS

/etc/hosts file, 15  
/etc/notrouter file, 20  
/etc/path\_to\_inst file, 15

## A

adapter installation instructions, 9  
ARI/FCI Parameter, 17  
ARP tables, 76

## C

CD-ROM  
  mount, 13  
compatible platforms, 5  
configuration overview, 11  
console terminal  
  system, 26

## D

data rate, 7  
debugging tools, 23  
diagnostics and troubleshooting, 39  
disconnect from network, 26  
DMA ASIC, 3  
driver installation, 13

## E

edit  
  /etc/path\_to\_inst file, 15  
  hostname.tr file, 16  
error messages, 27

## H

hardware overview, 3  
host IDPROM MAC address, 76  
hostname.tr file, 16

## I

IBM STP cable, 5  
ifconfig command, 24  
increasing buffers, 20  
install  
  driver, 13  
  software, 13  
installation overview, 11  
installation requirements, 4  
instance numbers, 8  
internetwork routing, 20

## J

jumper, 7

## L

local source routing bridges, 40

## M

MAC Address, 75

MAC address, 76

maximum performance, 65

maximum transmission unit (mtu), 16

messages, 26

mount CD-ROM, 13

mtu

    default, 16

    sizes, 16

## N

netstat command, 23, 24

network configuration

    planning, 65

    setting up routers, 75

network traffic

    routing, 75

## O

OBP, 6

OBP requirement, 5

Open Boot PROM (OBP), 6

OSI 7 layer model, 4

## P

performance factors, 65

ping, 23

pkgadd, 11, 13

pkgadd command, 13

pkgrm command, 12

platform compatibility, 5

## R

reconnect to network, 26

removing old instances, 12

resource configuration, 20

ring speed

    data rate selection jumper, 7

    default, 3

routing network traffic, 75

## S

script, post-installation, 13

set Sole Network Interface, 19

setting up routers, 75

SNAP encapsulation, 4

snoop command, 23

Soft Error Reporting, 17

software installation, 13

software overview, 4

source route table, 25

source routing, 4, 25

    bridges, 40

spray, 23

STP cable, 5

Sun4d requirements, 8

SunTRI/S adapter overview, 3

system messages, 26

## T

token ring networks, 1

tr\_nbufs parameter, 20

TRI/S

    TRI/S console messages

        PAGE, 39

troubleshooting and diagnostics, 39

## U

Ultra Enterprise xx00 requirements, 8

utilities, 23

## **V**

verify install, 24

