

Lucent Technologies
Bell Labs Innovations



Data Networking Products Session Maintenance Guide

255-100-209
Issue 2

© Copyright 1997 Lucent Technologies
All Rights Reserved
Printed in USA

Datakit and *StarKeeper* are registered trademarks of Lucent Technologies.
UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company, Ltd.

The information in this document is subject to change without notice.
Lucent Technologies assumes no responsibility for any errors that
may appear in this document.

Contents

Preface	vii
Feature Documentation	vii
Other Documents	vii
How This Guide Is Organized	viii
Overview	1-1
Fault Tolerance Through Session Maintenance	1-4
Feature Components	1-5
Feature Concepts	2-3
Functional Description	2-1
Concepts and Terminology	2-6
Feature Capabilities	2-15
Network Planning and Design	3-1
Network Topology and Session Maintenance	3-3
Trunk Resources	3-6
Scope of <i>StarKeeper II</i> NMS Control	3-7
Configuration, Simulation, and Analysis	4-1
Configuration Process	4-3
Simulation and Analysis	4-13
Administration	5-1
Reroute Administration	5-5
Reversion Administration	5-14
<i>StarKeeper II</i> NMS Programmers Interface	5-19
Appendix. Commands Reference	A-1
Index	I-1

Figures

1-1.	Fault Tolerance Through Session Maintenance	1-4
1-2.	Required Components for Session Maintenance	1-6
2-1.	Session Maintenance Events Scenario	2-5
2-2.	Parallel Path Rerouting	2-5
2-3.	Rerouting Through Assisting Nodes	2-6
2-4.	Node Types with Session Maintenance	2-9
2-5.	Channel Sets	2-11
2-6.	Sample Network and Node Reroute Tables	2-13
3-1.	Location of Nodes without Session Maintenance	3-4
3-2.	Ring Network Design	3-5
3-3.	Trunk Balance	3-5

Tables

5-1. Messages and Session Maintenance Events	5-4
5-2. Failure Reason Messages	5-10
5-3. Reason Messages for Dropped Channel Set	5-18

Screens

5-1. Trunk Failure Report Alarm—Session Maintenance Enabled	5-6
5-2. Manual Reroute Requested Status Message	5-7
5-3. Successful Reroute Report Alarm	5-8
5-4. Reroute Unsuccessful Report Alarm	5-8
5-5. Secondary Node Assigned Reroute Paths Status Message	5-11
5-6. Assisting Node Participation Report Alarm	5-12
5-7. Assisting Node Failed to Participate Status Message	5-12
5-8. Trunk Already Rerouted Status Message	5-13
5-9. Reroute Unnecessary Status Message	5-13
5-10. Trunk Recovery Report Alarm	5-15
5-11. Reversion Requested Status Message	5-15
5-12. Reversion Succeeded Status Message	5-15
5-13. Reversion Failure Report Alarm	5-16
5-14. Assisting Node Participated in a Reversion Status Message	5-16
5-15. When a Trunk-PQ Downloads Prior to Reversion	5-17
5-16. Node Drops Channel Set Message	5-17
5-17. Sample Script for Route Status	5-20
5-18. Sample Script for Automatic Reversion	5-21
A-1. enter trunk Command	A-2
A-2. enter trunk Command for an SMDS Trunk	A-3
A-3. enter trunk Command for a PQ Trunk	A-4
A-4. change trunk Command	A-5
A-5. delete trunk Command	A-6
A-6. enter node Command	A-6
A-7. change node Command	A-7

Preface

Session Maintenance is a network feature that provides automatic rerouting for active calls affected by trunk facility failures. It is a feature provided in the node software that must be administered through the *StarKeeper*® II Network Management System (NMS) Network Builder Application Package. Potential feature users (primarily network planners and administrators), therefore, need information about Session Maintenance for feature planning and network design; they also need user instructions for *StarKeeper* II NMS Network Builder for feature configuration and administration.

Feature Documentation

The *Session Maintenance Guide* assumes that readers have a general understanding of data networks, network administration experience, and some familiarity with *StarKeeper* II NMS. Readers who are actively involved in node and network administration should also see other technical documentation available for their network switch. Readers who have this background can then use the *Session Maintenance Guide* to plan and implement Session Maintenance in their networks. The guide provides

- a high-level feature description of Session Maintenance
- a functional discussion of feature operation with instructions on how to optimize the performance of Session Maintenance
- operational information, from an administrator's perspective, that covers configuration, analysis, tuning, and troubleshooting
- a description of the product line integration supporting the Session Maintenance feature and a discussion of the roles played by feature components

The *StarKeeper II NMS Graphic Systems Guide* (255-114-732) is a user's guide for all graphic applications. The guide includes a Network Builder section to define its capabilities and provide task-oriented procedures explaining how to use Network Builder tools to support Session Maintenance in the network. Network Builder support is *required* for Session Maintenance.

Other Documents

Planners and administrators of Session Maintenance should be familiar with the content of additional technical documentation supporting their network switch and *StarKeeper* II NMS. The following table shows generic document titles that cover topics relevant to Session Maintenance. Use the *Publications* brochure for the specific network switch for current titles and document numbers.

Network Switch	<i>StarKeeper II NMS</i>
<i>System Description</i>	<i>StarKeeper II NMS (Introduction)</i>
<i>Planning Guide</i>	<i>Planning Guide</i>
<i>Data Networking Products Messages Reference</i>	<i>Core Systems Guide</i>
<i>Data Networking Products Trunk Module Reference</i>	<i>Graphic Systems Guide</i>
<i>BNS-2000 Node Reference</i>	
<i>Data Networking Products Commands Reference</i>	<i>Commands Reference (on line)</i>
<i>Administration Quick Reference</i>	

In addition, *Data Networking Products Terminology* lists and defines many technical terms found within this document.

How This Guide Is Organized

- **Overview** summarizes the advantages of Session Maintenance, describes required components, and discusses how Session Maintenance interworks with other features to provide transport reliability and trunk fault tolerance.
- **Feature Concepts** provides a functional description of Session Maintenance and presents concepts and terminology related to Session Maintenance.
- **Network Planning and Design** discusses the planning process for Session Maintenance, covering network topology, internodal connectivity, and consideration of trunk and network management resources.
- **Configuration, Simulation, and Analysis** discusses Network Builder tools as building blocks for configuration, and when to use them to set up and place Session Maintenance in service in a network. The chapter focuses on how to use the Network Builder Simulator tool to design and analyze the operation of Session Maintenance in an actual or proposed network, and how to interpret Simulator output to make decisions about redesigning the network.
- **Administration** contains detailed information about real-time events that may occur in networks that employ Session Maintenance, and points out their implications for administrators. The chapter describes how to interpret system messages sent to *StarKeeper II NMS*, get additional information for making administration decisions, and add optional programming via the *StarKeeper II NMS Programmer's Interface* to automate Session Maintenance administration.
- **Commands Reference** provides examples of sessions from a local administration console that show usage of restricted node command options.

The guide includes an **Index** for alternative access to the subjects covered in this document.

Overview

Fault Tolerance Through Session Maintenance 1-4

Feature Components 1-5

Overview

Session Maintenance is an administrable feature that increases data transport reliability on the trunks providing connection-oriented services between network switches (nodes). User sessions are maintained despite trunk failures by automatically rerouting active sessions from the failed facility over spare capacity on other trunks within the network.

The feature is designed to take advantage of several network capabilities and strategies:

- Session Maintenance is supported on all trunk types except Trunk-T3I (TRK-T3I). This support allows network-wide administration to provide recovery from facility failures, and adds fault tolerance for trunks that complements node high availability features.
- Most existing networks with sufficient bandwidth and internodal connectivity can support failure recovery via Session Maintenance without physical modification. Administrators can configure previously unused trunk capacity for use as standby trunk capacity that supports the alternate reroute paths required for Session Maintenance activities.
- Session Maintenance eliminates the need to back up regular trunks with spare trunks that are only used when the regular trunks fail. It allows administrators to make optimal use of the bandwidth capacity available in the network by allowing normally routed and rerouted traffic to share the same trunks.
- Processing capabilities in the central processing unit (ECPU/CCM) provide performance that supports the quick session switching required for certain higher level protocols, and Switch modules provide the capacity to support administering the additional number of channels needed to implement the Session Maintenance feature.
- The integration of node operations and administration with *StarKeeper II* NMS provides a means for administrators to implement the feature in their networks. Configuration and monitoring activities for Session Maintenance are centralized with the *StarKeeper II* NMS Network Builder Application Package. Network Builder also provides a simulator tool that models trunk failure and recovery events throughout the network, allowing administrators to preview the expected performance of trunks set up for Session Maintenance without putting actual traffic at risk.
- Network switches that support Session Maintenance provide the feature in the node software, but mixed networks of Session Maintenance and non-Session Maintenance trunks are also supported.

Fault Tolerance Through Session Maintenance

Session Maintenance in the network provides an integrated solution for overall network fault tolerance (Fig. 1-1).

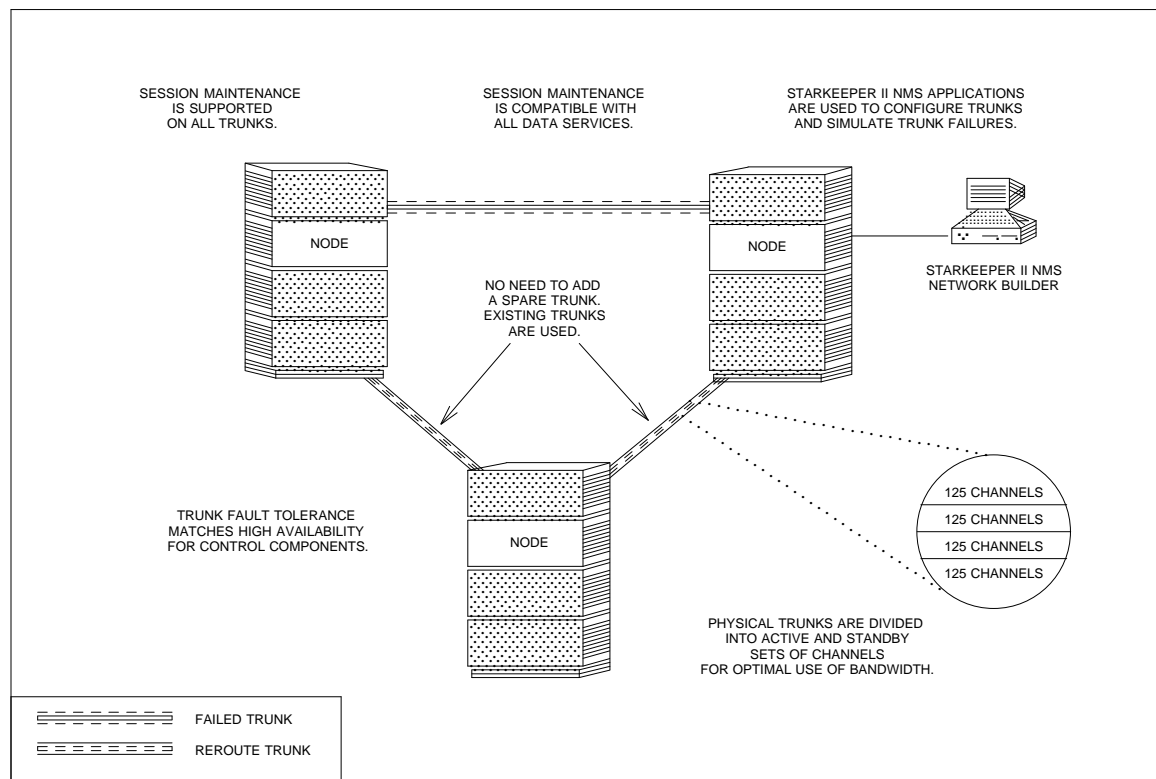


FIGURE 1-1. Fault Tolerance Through Session Maintenance

Any network switch capable of providing Session Maintenance also supports optional redundant node control modules. These options, and redundant Switch modules, provide exceptional fault tolerance for all highly critical node control hardware. Session Maintenance complements this advantage by increasing internodal trunk availability, adding the capability to avoid active call disconnections owing to errors that result in trunk failure.

Feature administration includes flexibility. Networks with alternate internodal connectivity over trunks that support Session Maintenance can rely on swift automatic rerouting controlled in part by configurable parameters. Although the parameters supplied by the system when the feature is configured are generally sufficient, most parameter values, as well as the order in which the system selects alternate trunk paths, can be changed by administrators. This flexibility allows the direction of rerouted calls over certain trunks, either to maintain necessary throughput or isolate call traffic to specific paths in the network.

The centralized administration services provided by *StarKeeper II* NMS Network Builder simplify ongoing feature management as the network grows or as network services and traffic patterns change. Network administrators can make use of capabilities that permit snapshots of network behavior during selective simulated trunk failures and rerouting events. These capabilities permit administrators to model a variety of trunk failure scenarios, using the output to make reliable decisions about tuning trunks throughout the network for optimal performance.

Session Maintenance provides some efficiencies that allow cost savings; administrators can reconfigure existing physical trunk resources so that excess trunk bandwidth is used to alleviate potential trunk failures.

Feature Components

Session Maintenance requirements include node and *StarKeeper II* NMS components. Required elements are

- Node software supporting the Session Maintenance feature, either a full feature package or a base feature package supplemented by a network management feature package.
- a *StarKeeper II* NMS Core System
- a *StarKeeper II* NMS Graphics System running the *StarKeeper II* NMS Task Manager Software Package and Network Builder Application Package

Supporting nodes are required at both ends of a trunk configured for Session Maintenance. Figure 1-2 shows feature package combinations that include the required node software and *StarKeeper II* NMS components.

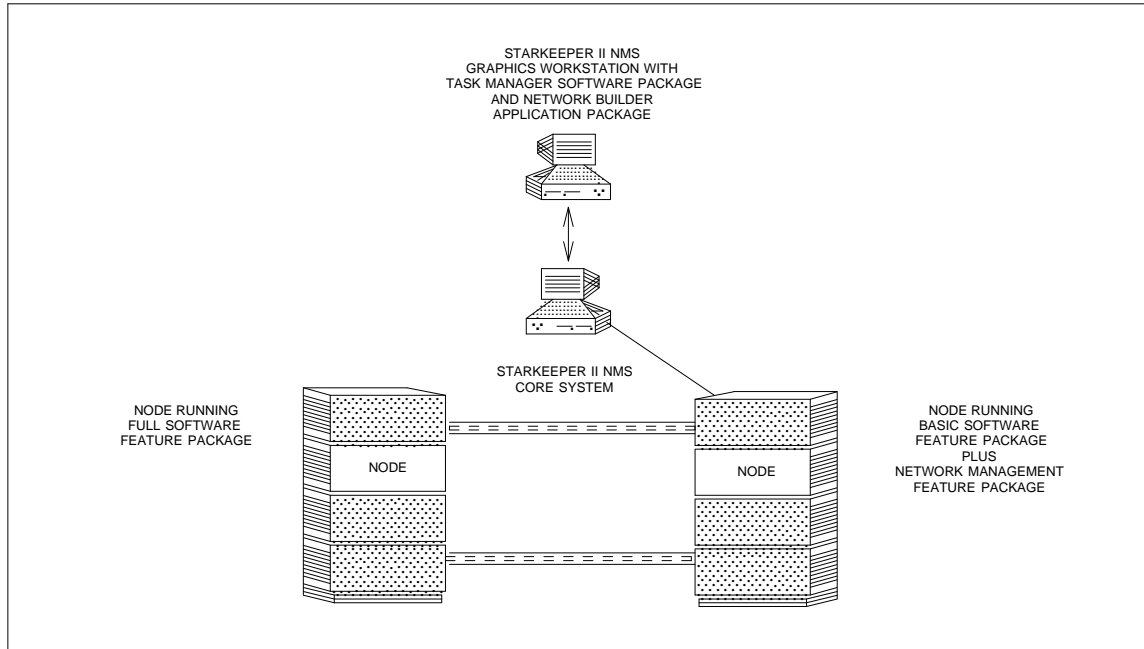


FIGURE 1-2. Required Components for Session Maintenance

Software feature packages for a specific network switch are detailed in the *System Description* for that network switch. Hardware and software requirements for running *StarKeeper II NMS* and *Network Builder* are described in the *StarKeeper II NMS Planning Guide*.

Feature Concepts

Functional Description	2-3
Concepts and Terminology	2-6
Node Types	2-7
Node Naming	2-9
Session Maintenance Trunks	2-9
Bandwidth	2-10
Channel Sets	2-10
Node Reroute Tables	2-11
Reroute Process	2-14
Network Builder	2-15
Feature Capabilities	2-15

Feature Concepts

This chapter includes functional details on how the feature components introduced in Chapter 1 interwork to provide Session Maintenance in the network. It introduces and discusses concepts and terminology needed to understand the feature. The chapter also lists Session Maintenance capabilities and discusses constraints that must be observed.

Functional Description

To configure trunks for Session Maintenance, administrators use Network Builder to partition the existing channels on physical trunks into logical channel sets. As the channel sets are configured for each trunk, administrators exercise the option to designate some of them *active* channel sets, which support normal call setup and data transport. New sessions on trunks configured for Session Maintenance are always set up over the active channel sets on the trunk.

The remaining channel sets on the trunk can be designated as *standby* channel sets. Standby channel sets are not used for normal data traffic; that is, the node central processing unit will not set up a new session on a standby channel set. The standby channel sets are reserved to provide facilities that support alternate paths through the network when calls on active channel sets from a failed trunk must be rerouted.

Using the data supplied by administrators for trunk configuration, Network Builder also generates configuration database tables, called Node Reroute Tables (NRT), for each node in the network. These tables contain all the information needed for nodes to determine, in real time, available reroute paths around a failed trunk. Network Builder facilitates downloading a Node Reroute Table to each node in the network, distributing an accurate, intelligent database for the Session Maintenance feature throughout the network. With trunks configured for Session Maintenance, Node Reroute Tables downloaded, and some additional node tuning parameters defined, the network can actively support Session Maintenance.

Continual, real-time testing routines permit the nodes to detect highly errored or failed trunks within an administrator-defined window of time (2–120 seconds). When a node detects that the error rate of a trunk configured for Session Maintenance exceeds an administered threshold, it invokes a reroute, locating standby channel sets on other trunks that can provide alternate routes for the affected trunk channel sets. The node attempts to locate standby channel sets even if the affected channels do not happen to contain active calls at that time.

To locate the standby channel sets, the node looks at information about neighbor nodes in its Node Reroute Table. A node that is seeking alternate routes for channel sets on the failed trunk is thus capable of locating a path parallel to the failed trunk or, in the absence or insufficiency of a parallel path, of negotiating with other nodes for unused bandwidth on other facilities. In the second case, the node issues requests to neighbor nodes for reroute paths over any available standby channel sets through these neighbor nodes. The alternate path that is eventually

negotiated may be a multi-hop path around the failed trunk traversing up to three neighbor nodes. The total number of requests sent is equal to the number of active channel sets on the failed trunk plus a number of additional requests consistent with the value assigned to a node tuning parameter. The additional requests increase the probability that all calls will be successfully rerouted; see **Node Configuration and Tuning** in Chapter 4.

After reroute requests are honored and acknowledged by neighboring nodes and a full path is negotiated, channel sets are rerouted to the newly established paths. Detection of a failed or highly errored trunk and the steps taken to reroute the affected channel sets can be accomplished within ten seconds, transparent to the devices and users engaged in sessions that may have been set up over the trunk. Rerouting is performed by the central processing unit and Switch module.

When a Trunk-PQ (Priority Queuing) is rerouted, the Committed Information Rate (CIR) is not maintained on the reroute trunk, even if it is also a Trunk-PQ. In addition, CIR calls cannot be set up while the trunk is in the reroute state. When a Trunk-PQ is reverted back to the original trunk, CIR maintenance is resumed unless the trunk has been downloaded. If the trunk has been downloaded, CIR calls are dropped and will be set up again automatically (since they are Predefined Destinations [PDDs]), so that the CIR information is restored. An alarm is displayed to inform the administrator.

After the failed trunk is declared healthy (when it again passes the continual, real-time testing routines), channel sets that were routed around the failure can be moved back to their original path manually via a single administrative command or automatically through the *StarKeeper II* NMS Programmer's Interface. A basic scenario depicting these events is shown in Figure 2-1.

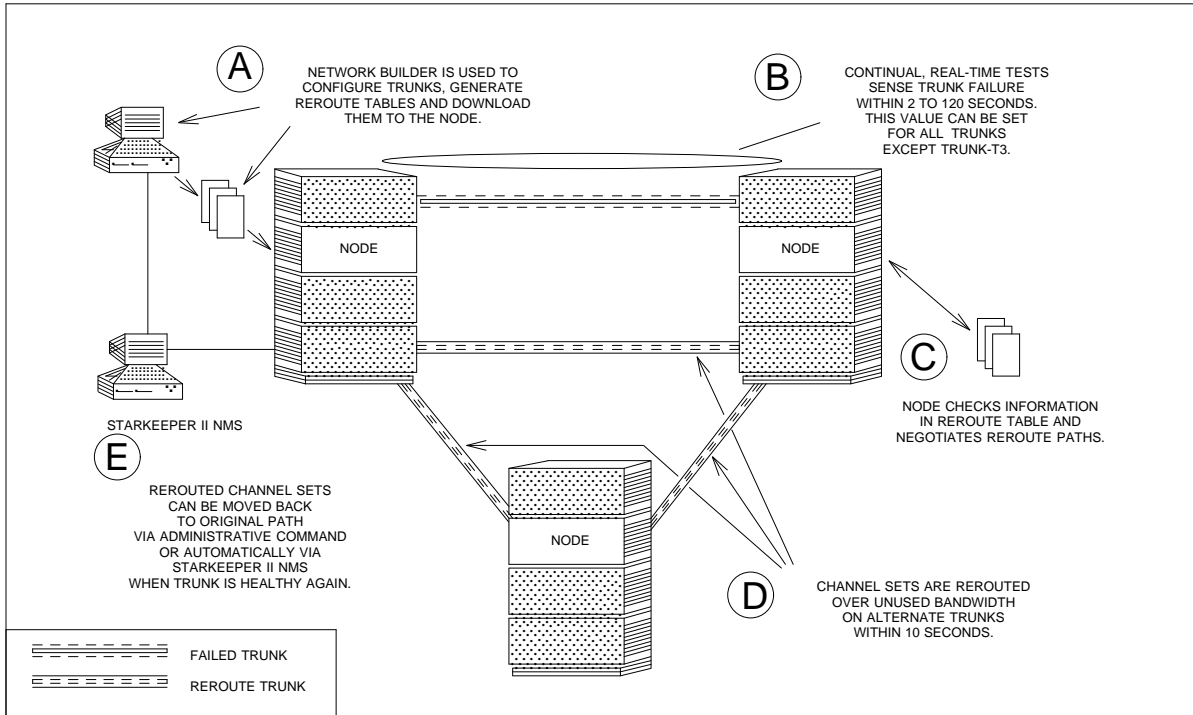


FIGURE 2-1. Session Maintenance Events Scenario

The negotiated reroute path may be either a trunk parallel to the failed trunk (Fig. 2-2) or one that passes through up to three assisting nodes (Fig. 2-3).

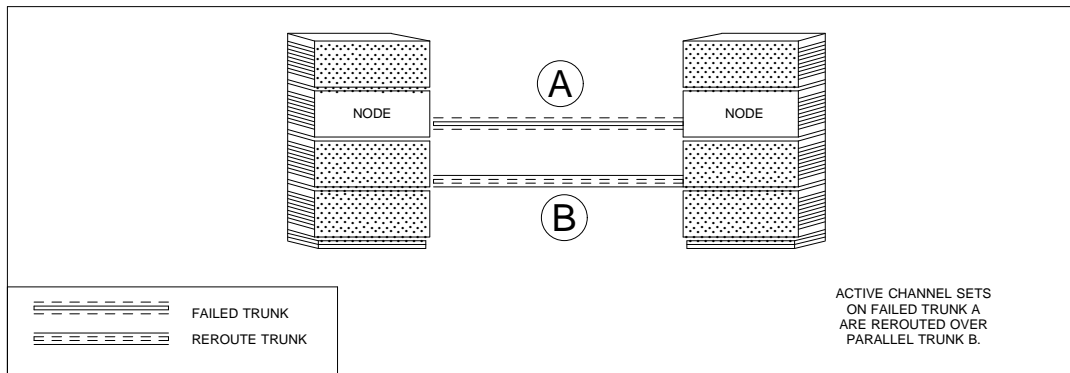


FIGURE 2-2. Parallel Path Rerouting

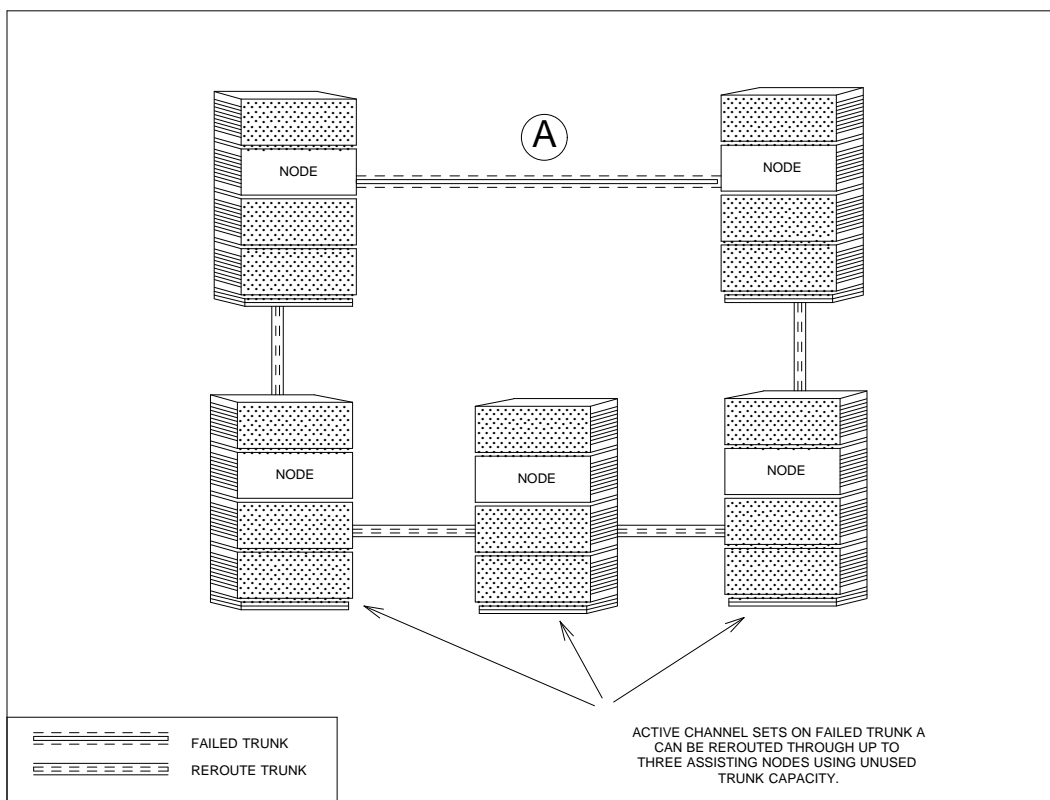


FIGURE 2-3. Rerouting Through Assisting Nodes

Concepts and Terminology

The following sections provide details about concepts and terminology related to Session Maintenance, including:

- node types
- node naming
- Session Maintenance trunks
- bandwidth
- channel sets
- Node Reroute Tables
- reroute process

- Network Builder

Node Types

A node at either end of a Session Maintenance trunk is called a reroute node (Fig. 2-4). For each pair of nodes joined by a Session Maintenance trunk, one node is designated the *primary* node and the other is the *secondary* node. An *assisting* node is a one- or two-hop neighbor of the primary node that may be requested to participate in a reroute. Entries made in the trunk database via Network Builder indicate which node is the primary node for the connection. All trunks between a given node pair have the same primary node.

Primary Nodes

The primary node is responsible for:

- detecting a trunk failure
- initiating a reroute
- issuing requests for reroute paths based on information in its Node Reroute Table
- reporting the number of reroute requests and their status to *StarKeeper II* NMS
- detecting and reporting trunk recovery

Secondary Nodes

The secondary node is responsible for:

- responding to each reroute request received from the primary node
- assigning an active channel set from the failed trunk onto the negotiated reroute path
- rejecting unnecessary additional requests after all of the channel sets on the failed trunk have been rerouted
- reporting to *StarKeeper II* NMS that the assignment of channel sets to reroute paths is complete

Assisting Nodes

An *assisting* node is a one- or two-hop neighbor of the primary node that may be requested to participate in a reroute. (A hop is an internodal trunk.) Assisting nodes are responsible for:

- forwarding requests from the primary node or other assisting nodes toward the secondary node
- setting (assigning) the new path upon acknowledgment from the next node and thus becoming a participant in a reroute
- reporting to *StarKeeper II* NMS (if the capability is enabled) the final status of all requests they received

Requests forwarded by assisting nodes reflect the number of available standby channel sets on the potential reroute trunks.

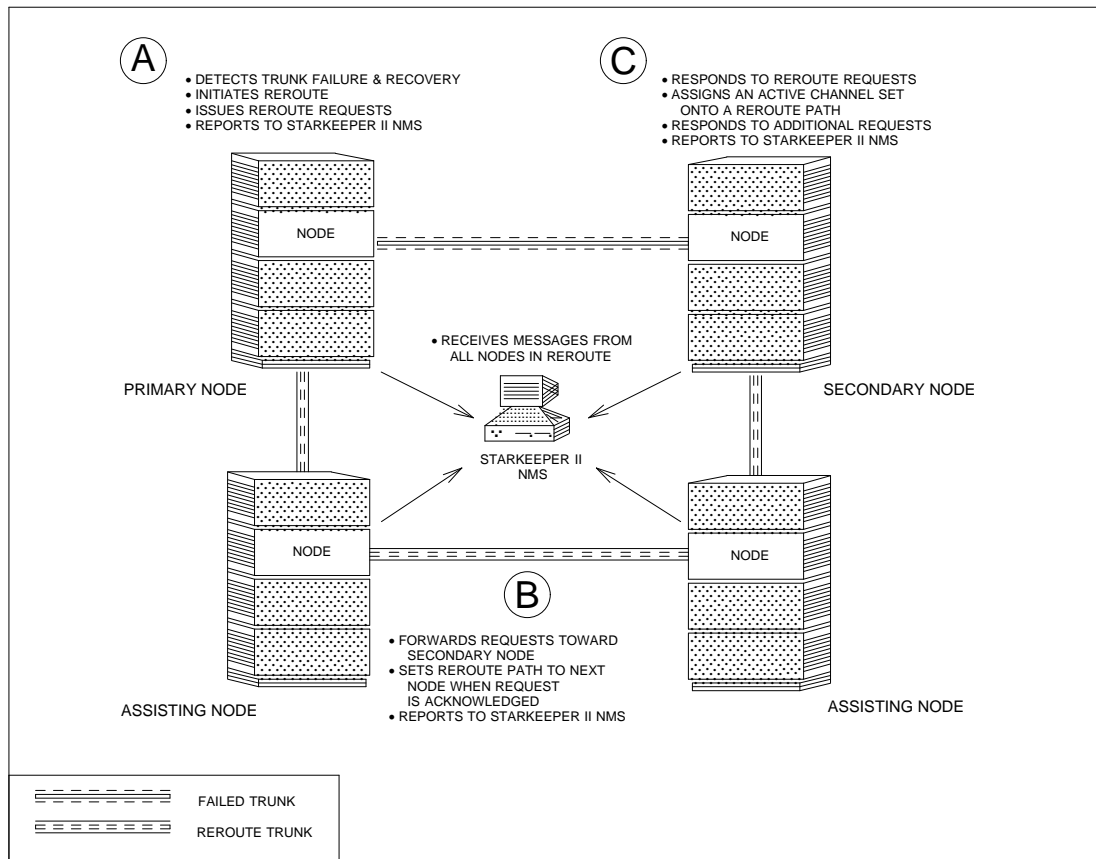


FIGURE 2-4. Node Types with Session Maintenance

Node Naming

Session Maintenance requires care in assigning node names. Administrators must ensure that node names within a network are unique, at least within four hops of any node involved in a reroute.

Session Maintenance Trunks

Session Maintenance trunks must be configured using Network Builder. These trunks differ from other network trunks in that additional data is required to administer them. Rerouting is performed without regard for the trunk type. Channel sets from one type of trunk can be rerouted on any other type of trunk, as long as there are standby channel sets with sufficient bandwidth available in the alternate path.

Bandwidth

Node bandwidth calculations include both user data and associated overhead (except for the Trunk-T3I). The Session Maintenance feature uses an accounting system for bandwidth primarily intended to reduce the likelihood that excessive delays will occur on trunks that participate in reroute paths.

Trunks are configured to have an expected bandwidth usage. This value is used to determine:

- the expected bandwidth usage associated with each reroute request
- the standby bandwidth usage of a given trunk

When a standby channel set is called into service, the expected bandwidth associated with the reroute request is subtracted from the standby bandwidth. Note that this accounting system is used just to facilitate reroute negotiation. Full actual bandwidth of the trunk is available to all sessions on the trunk being shared by normal trunk queuing algorithms.

Channel Sets

A trunk that is configured for Session Maintenance is divided into channel sets that contain up to 125 contiguous user channels. Each channel set is designated as either an active or standby channel set. Active channel sets carry traffic just like standard physical trunks. Standby channel sets normally carry no calls, but are reserved to be ready to take over the load of active channel sets from other trunks if a reroute is needed.

When a physical trunk fails, the primary node's Session Maintenance recovery process locates standby channel sets in the network that are capable of carrying the total expected bandwidth utilization represented by each active channel set on the failed physical trunk. All channels within each active channel set are then mapped into replacement channels within standby channel sets.

For most network topologies, it is possible for active channel sets from a single failed trunk to be rerouted over different trunks and assisting nodes. Most reroute paths are one or two hops, but can consist of up to four trunks with three intervening assisting nodes.

Figure 2-5 shows the active channel sets on one trunk being mapped into the standby channel set space of another trunk. In this case, the two reroute paths happen to be one hop and are over the same trunk. They need not be. They could be up to four hops long and follow significantly different paths through the network.

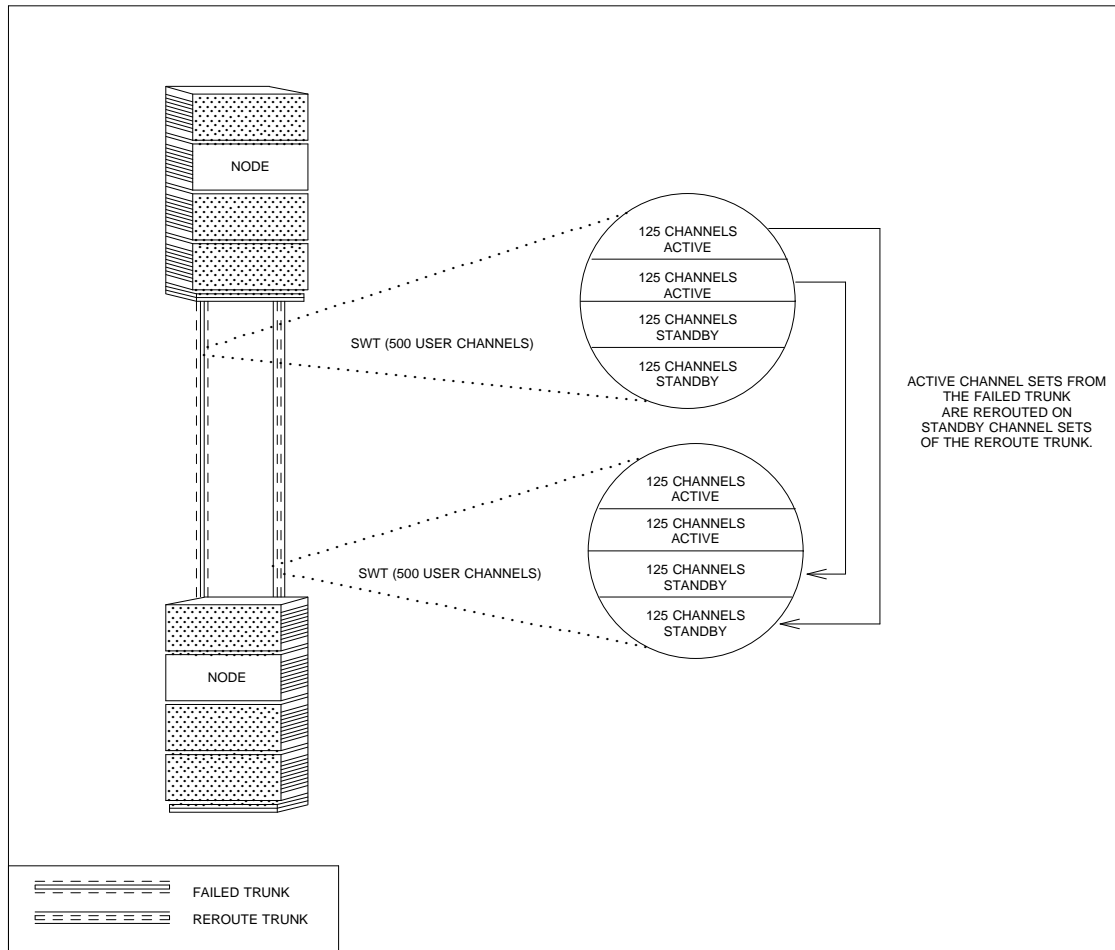


FIGURE 2-5. Channel Sets

Node Reroute Tables

The Node Reroute Table (NRT) is a node database structure used by the Session Maintenance feature. It contains an entry for each node that is reachable via one or two hops over Session Maintenance trunks. For each such node, the NRT contains a list of preferred assisting nodes and, for each one-hop neighbor, a list of physical trunks that connect to the node.

On a primary node, the NRT is consulted to find where to send the initial requests around the failed trunk to the secondary node.

On an assisting node, the NRT is consulted to determine if the secondary node is one hop away. If it is one hop away, a trunk is selected. If not, a next assisting node is determined.

Figure 2-6 shows a five-node sample network. As an example, if an *smverify* command (a *StarKeeper II* NMS command that provides a report) is entered for the node named HOLM, the output obtained from the NRT on node HOLM is shown adjacent to each of HOLM's neighbor nodes. The figure indicates that complete and valid NRTs have the following characteristics:

- *Neighbor Node Name.*
An entry is supplied for all one- and two-hop neighbors.
- *Number of Hops to This Neighbor.*
This number will be either one or two.
- *Assisting Node Names.*
For a one-hop neighbor the list includes nodes that are one hop from the owner node (HOLM) and within three hops of the neighbor node referred to in this list. The three-hop path cannot include the owner node. See the entry for node FREE.
For a two-hop neighbor the list includes nodes that are one hop from the owner node (HOLM) and within two hops of the neighbor node referred to in this list. The two-hop path cannot include the owner node. See the entry for node SUMM.
- *Trunk Module Addresses.*
For one-hop neighbors this is a list of trunk modules that go to the neighbor node referred to in this list.

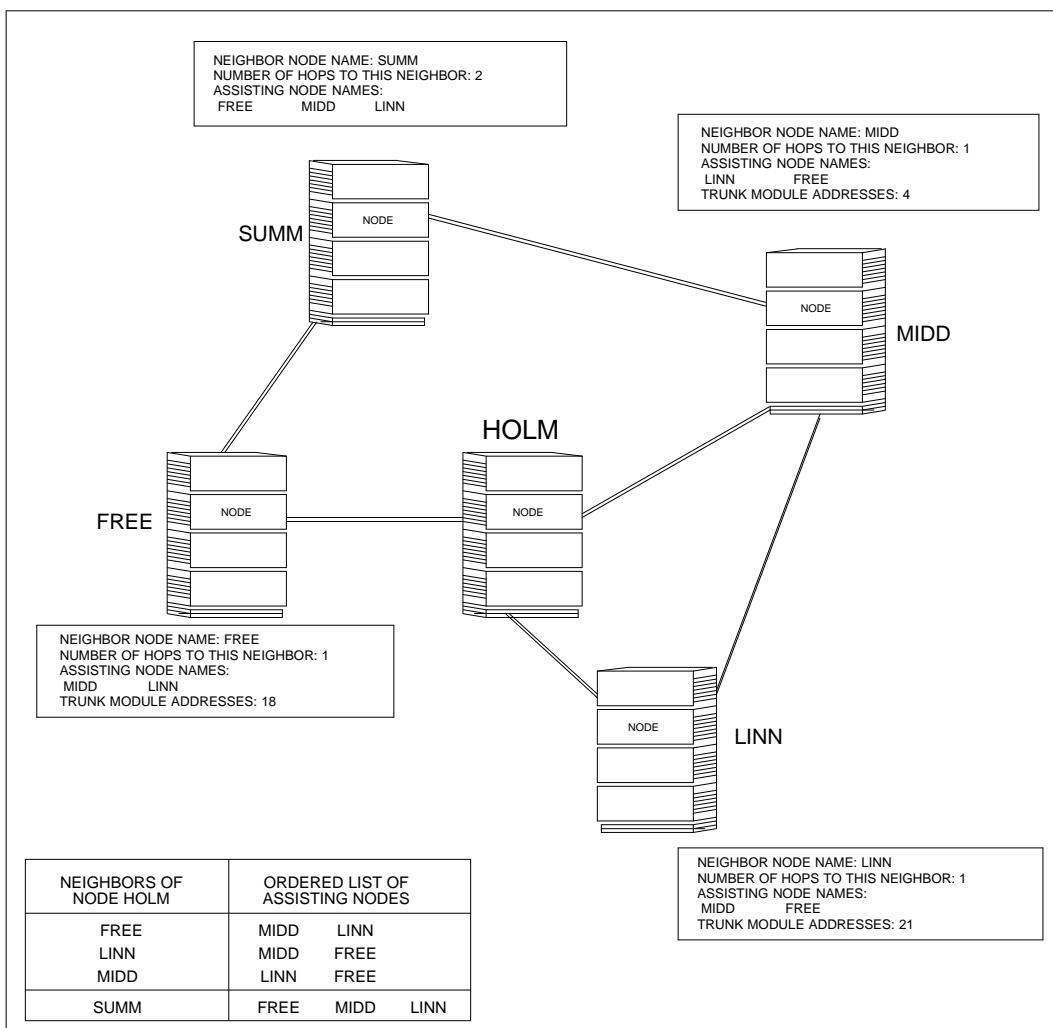


FIGURE 2-6. Sample Network and Node Reroute Tables

Node Reroute Tables are generated by Network Builder based on trunk configuration data. Network Builder software creates NRTs that order assisting nodes based on a figure of merit that considers the number of hops from the assisting node to the specified neighbor. If an administrator disagrees with the choices made by the software (for example, if the software lists a node connected by a 56 Kbps trunk ahead of one connected by a 1.5 Mbps trunk), the NRT can be edited within Network Builder to reflect the administrator’s choices of preferred neighbor nodes and trunks.

For example, in each sample output shown in Figure 2-6, under "Assisting Node Names," there is a preferred order in the list of node names that indicates the figure of merit for assisting nodes derived from the number of hops from the neighbor node. This order corresponds to information supplied in the table that appears in the figure. If the trunk from node HOLM to node MIDD failed, the NRT for HOLM specifies that assisting node LINN, a one-hop neighbor of MIDD, is tried first. Node FREE, a two-hop neighbor of MIDD, would be tried second.

Reroute Process

Rerouting is a three-phase process. First, a failure must be detected. Then, appropriate reroute paths must be negotiated and reserved, and active channel sets must be rerouted onto these paths. These two steps can occur within ten seconds of trunk failure (depending on the administered threshold value). Finally, when the failed trunk is considered healthy, the traffic can be switched back to its original path with a single command.

- *Phase 1: Failure Detection.*

Physical trunk failure is detected only on the primary node. Keep-alive messages that monitor the health of the facility are sent over a loopback path at a rate of two per second. The node keeps track of the number of consecutive seconds in which at least one keep-alive message is lost in a sliding window of time. Except for the Trunk-T3, the detection threshold window is settable, with a default of four seconds. If the threshold number of consecutive seconds is reached, a failure is declared. Failure detection for the Trunk-T3 is handled by the module on the basis of standard facility alarms.

- *Phase 2: Negotiation and Rerouting.*

The negotiation phase is required in order to determine that complete reroute paths exist and are in working order, and to make certain that the primary and secondary nodes agree on the reroute paths to be used. To account for the possibility that not all requests lead to successful reroute paths, negotiation is attempted for slightly more paths than are needed. The first ones to be successful are used, and the excess paths are freed.

The rerouting phase includes the actual rerouting of channel sets from the failed trunk to the appropriate reroute paths.

- *Phase 3: Switching Back.*

At some point the failed trunk is functional again and the active channel sets that had been rerouted around the trunk can be switched back to it. An informational message is sent by the primary node when the facility reaches a user-specified service quality. This message typically is followed by a single network administrator command to the primary node to restore the channel sets to the original trunk.

The process of actually switching back is fairly simple; all active channel sets that belong to the restored physical trunk but are currently rerouted are located and remapped to their home locations on the physical trunk. The assisting nodes then dismantle their reroute paths.

Network Builder

The *StarKeeper II* NMS Network Builder Application Package is a software package that runs on a workstation connected through the network to a *StarKeeper II* NMS Core System. The package is required to administer Session Maintenance. It is a forms-based application that facilitates building network configuration databases. It also simplifies the downloading and population of configuration databases at the node and at *StarKeeper II* NMS providing point-and-click data entry via a menu-driven interface. In addition to providing a vehicle for database entries and changes, Network Builder generates reports and performs analyses.

Besides generating NRTs and downloading them to the nodes, Network Builder can be used to:

- configure other node entities (trunks, groups, and addresses)
- evaluate network topology
- evaluate existing routing patterns
- analyze Session Maintenance trunk failures
- provide comprehensive reports

For configuration, Network Builder provides capabilities to enter configuration data for nodes and trunks. For analysis, Network Builder can be used to simulate network behavior for single or multiple trunk failures. Chapter 4 discusses the implications of using Network Builder for configuration and analysis; see also the appropriate *StarKeeper II* NMS documentation for details.

Feature Capabilities

- Session Maintenance provides flexibility that allows administrators to specify the maximum number of active user channels the trunk module will use, and the number of active/standby channel sets on the trunk. Thresholds for declaration of trunk failure and trunk recovery can also be specified, along with primary, secondary, and assisting node names, and the percentage of total trunk capacity expected to be used in normal operation.
- All network services are compatible with Session Maintenance.
- Session Maintenance can be automated in conjunction with the *StarKeeper II* NMS Programmer's Interface, as detailed in Chapter 5.
- Session Maintenance is designed to handle trunk failures, not node failures. Complementary node reliability options include redundant central processing units, Switch modules, disk/tape subsystems, and power supplies.
- Certain network switches are capable of achieving additional diversity with Session Maintenance when trunk modules are distributed among different shelves of multi-cabinet nodes. In such configurations, a shelf failure affects only the modules on the failed shelf, not every trunk the node maintains to other nodes.

- If a trunk fails, any virtual circuits that it is currently backing up on its standby channel sets are dropped; they are not rerouted. Healthy trunks should be switched back to normal routing as soon as possible. This practice improves overall performance and lessens the likelihood of having traffic on standby channel sets.
- If a node that may be involved in a reroute does not have up-to-date information about the network topology, the likelihood of the reroute succeeding is diminished. Timely regeneration of NRTs after a change in network topology, followed by a simulation, ensures up-to-date, correct reroute tables.
- For multiple simultaneous trunk failures, a completely successful reroute for all active channel sets is not guaranteed. For most cases, however, success can be predicted by analyzing the output of simulation conducted through Network Builder.
- Any node that lacks the Session Maintenance capability (for example, a SuperStar node) should not be included in the path of a potential reroute. Nodes without Session Maintenance can, however, be located at the edges of the network where rerouting through them is not a concern.
- If a Trunk-PQ fails, the Committed Information Rate (CIR) for calls on that trunk are not propagated to standby trunks. When the calls are reverted back to the original path after a reroute, CIR values will be recovered unless the original Trunk-PQ has been downloaded. If the Trunk-PQ has downloaded, causing the loss of CIR values, calls will be dropped. Since these calls are PDDs, they will be automatically set up again, allowing the CIR values to be redefined.

Network Planning and Design

Network Topology and Session Maintenance	3-3
Nodes Without Session Maintenance	3-3
Parallel Trunks	3-4
Ring Network Design	3-4
Hierarchical Networks and Trunk Balancing	3-5
Mesh-Grid Networks	3-6
Trunk Resources	3-6
Number of Trunks and Trunk Location	3-6
Trunk Types	3-6
Trunk Speeds	3-7
Committed Information Rate	3-7
Scope of <i>StarKeeper</i> II NMS Control	3-7

Network Planning and Design

Network planning and design for Session Maintenance should consider a number of high-level issues regarding

- network topology
- trunk resources
- scope of *StarKeeper II* NMS control

Network Topology and Session Maintenance

For Session Maintenance to have value, a node must maintain more than one trunk to the rest of the network. On nodes with two or more internodal trunks, any trunk that meets the design prerequisite (a trunk between two nodes capable of supporting Session Maintenance) should be a Session Maintenance trunk.

The sections that follow contain some design criteria for various aspects of network topology, including:

- nodes without Session Maintenance
- parallel trunks
- ring networks
- hierarchical networks and trunk balancing
- mesh-grid networks

Nodes Without Session Maintenance

An extended network may contain nodes that lack the Session Maintenance capability. The network locations of nodes that do not support Session Maintenance should be considered so that rerouting is not affected. For the best value in network performance, nodes without the Session Maintenance feature should be located at the network edges, as shown in Figure 3-1. At the edges, they avoid interrupting services provided by nodes with Session Maintenance and also receive some benefit, because the Session Maintenance portions of their multi-hop calls can be rerouted.

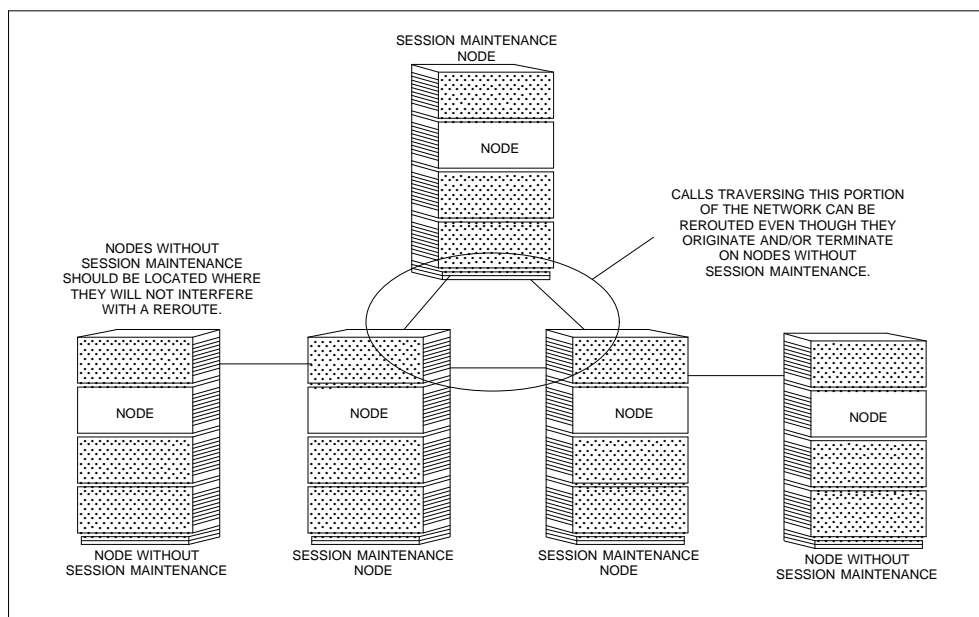


FIGURE 3-1. Location of Nodes without Session Maintenance

Parallel Trunks

Session Maintenance minimizes the need for parallel trunks. Reroute paths can follow multiple hops throughout the network. If parallel trunks are used, take care to ensure trunk route diversity; request minimal common facility equipment and separate physical paths from the facility provider. Place trunk modules in separate shelves of multi-cabinet nodes, if possible.

Ring Network Design

Keep ring network designs limited to five or fewer nodes per ring. If the number of nodes exceeds five, configure a diagonal trunk across the ring to create a double ring. This configuration is necessary because Session Maintenance operation is limited to a maximum of four hops between nodes. Remember, a hop is synonymous with an internodal trunk. Two nodes connected by a trunk are one hop apart even though they may be at the same location; see Figure 3-2.

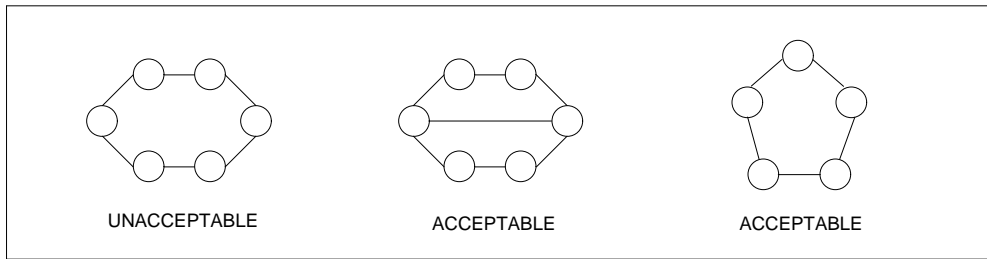


FIGURE 3-2. Ring Network Design

Hierarchical Networks and Trunk Balancing

The number of internodal trunks on hub nodes in a hub-leaf type network should be balanced throughout the network. When a primary node seeks reroute paths, the extent to which hub nodes are balanced in the network configuration increases the potential for a primary node to successfully reroute all channel sets because enough standby channel sets exist to ensure connectivity. In Figure 3-3, trunks are redistributed until the network achieves a configuration in which the hub nodes in both East and West have four internodal trunks.

The network need not be re-engineered all the way to the *best* design. A Simulator run (see Chapter 4) could predict that a *poor* design may be more than sufficient for the specific networking application. Geography and facility costs may be driving reasons to not have a totally balanced network.

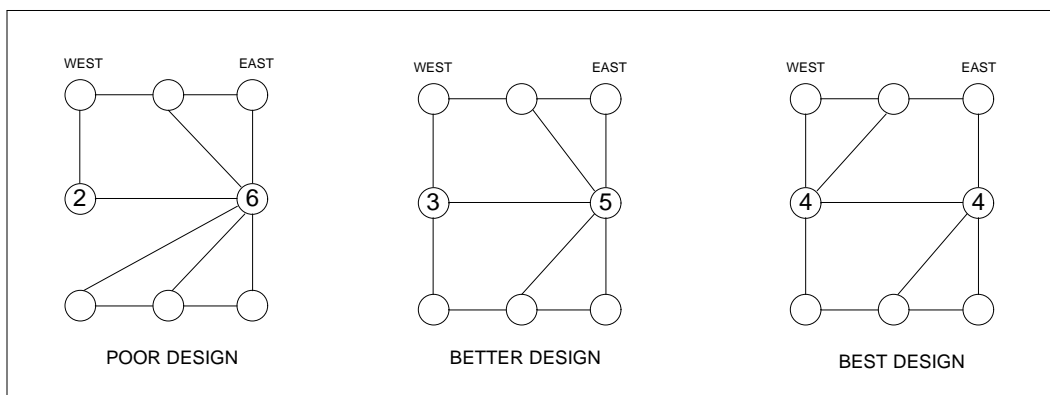


FIGURE 3-3. Trunk Balance

Mesh-Grid Networks

Mesh-grid networks tend to be the best type of network for overlaying Session Maintenance. The four-hop reroute path limitation usually presents no problem in mesh topologies. In addition, most mesh networks are balanced by default.

Trunk Resources

Existing networks have established trunk configurations that provide a variety of services at an accustomed level of performance. Administrators should give some consideration to the effect that either a single or multiple reroute might have on that configuration. In general, an existing network that provides sufficient data transport performance after a trunk failure (with predefined destination (PDD) automatic re-establishment functionality) will perform at least as well with Session Maintenance enabled. Given the control the administrator has over reroute paths, the performance can be much better.

A new network that is designed to the standard design values—that is, 60–75% busy hour trunk utilization— should be able to handle most failure situations if there is also sufficient connectivity.

For either existing or new networks, a few Simulator runs (see Chapter 4) should illustrate the quality of the solution and point out capacity bottlenecks, if any exist.

Things to consider include the following:

- number of trunks and trunk location
- trunk types
- trunk speeds

Number of Trunks and Trunk Location

In most cases a sufficient number of trunks would be configured for normal traffic. Trunks may have to be added in consideration of the topology guidelines mentioned above.

In multi-cabinet nodes, distribute the location of trunk modules so all trunk modules are not located on the same shelf. This provides some diversity for nodes that have the capability for shelf administration.

Trunk Types

Some trunk types support a larger number of channels, and will therefore support a greater number of channel sets. For example, the Trunk-T3 module can support 8,000 channels and up to 64 channel sets. Where four or more active channel sets are needed on a trunk (500 or more user channels), consider using trunks that support a large number of channels, if possible, so that some standby channel sets can also be configured on the trunks.

Trunk Speeds

For best results, Session Maintenance is designed to operate on trunks running at 56 Kbps or higher. In general, a network engineered with enough trunk bandwidth for any reroute strategies should have trunk speeds high enough to handle Session Maintenance reroutes.

Committed Information Rate

A Trunk-PQ can perform as a rerouted trunk unless it has been configured for CIR only. This is because whenever a Trunk-PQ acts as a reroute trunk, all channel sets attempting to be rerouted to it resemble non-CIR channel sets (regardless of whether they were originally configured for CIR or non-CIR). A Trunk-PQ configured for "both" must have sufficient bandwidth reserved for non-CIR traffic in order to service rerouted channel sets.

Scope of *StarKeeper II* NMS Control

All nodes that support Session Maintenance must be monitored by *StarKeeper II* NMS. This control can be a single *StarKeeper II* NMS Core System with a Workstation running Network Builder or a suite of *StarKeeper II* NMS Core Systems with a Workstation running Network Builder that has full access to the network of Session Maintenance nodes.

It is necessary that all nodes that may participate in a Session Maintenance event be managed by the same *StarKeeper II* NMS. This requirement ensures that Network Builder has sufficient knowledge of configuration data to generate complete and accurate NRTs.

Configuration, Simulation, and Analysis

Configuration Process	4-3
Trunk Configuration	4-4
Node Reroute Table Configuration	4-9
Node Configuration and Tuning	4-10
Simulation and Analysis	4-13
Simulator Application	4-13
Using the Simulator	4-14
Interpreting the Output	4-14

Configuration, Simulation, and Analysis

Session Maintenance requires centralized network management, and can only be configured using the *StarKeeper II* NMS Network Builder. Configuration from the node administration console is confusing and error-prone. Certain node commands, however, display secondary prompts with options for Session Maintenance for nodes that support Session Maintenance. These prompts are shown in Appendix A, with the recommendation to refrain from responding to them from the node console. The Node Reroute Tables (NRTs) necessary for Session Maintenance are designed to be generated (built) by *StarKeeper II* NMS. The main problem with using node commands to administer Session Maintenance is that data entered in the administrative session will be unknown to *StarKeeper II* NMS and missing from the network's NRTs, which adversely affects service. For Session Maintenance to perform as designed, the feature must be configured through *StarKeeper II* NMS.

This chapter discusses the implications of entering certain values for Session Maintenance parameters via Network Builder forms. It is assumed readers have gained knowledge of Network Builder from *StarKeeper II* NMS documentation.

Configuration Process

The sequence of operations involved in Session Maintenance configuration is summarized in the numbered list that follows. Inherent in this sequence is a configuration philosophy that is supported by functionality designed into Network Builder. Network Builder provides the ability to enter data and either place it on hold or submit it to the network. Simulation and analysis can be performed on either held or real data. Thus, the configuration process using Network Builder provides for a series of iterations that allow administrators to examine the results of held configuration data and perhaps return to readminister certain entities before submitting the data to the network. In general, potential changes should be put on hold so that the effect of the changes can be analyzed before changes are actually made to the node databases.

1. *Enter the configuration.*

This step refers to the Network Builder *Configure: Node* and *Configure: Trunk* tasks, and within the task, either entering data or selecting one of the available options. Each task brings up a window and subsequent panes that contain fields sensitive to the specific object being configured. The *Configure: Node* task may be used to tune node parameters so that the node is optimized for Session Maintenance. The *Configure: Trunk* task is used to configure individual trunk parameters or to add or delete trunks to or from the network.

2. *Generate Node Reroute Tables (NRTs).*

This step refers to the Network Builder *Configure: NRT* task. The NRTs are generated from the real data entered during configuration of nodes and trunks. The NRTs can be held in Network Builder and sent to either the *StarKeeper II* NMS or node databases at a later time. This capability can be useful before node hardware is in place.

The step also refers to the view and edit option within the Network Builder *Configure: NRT* task. The view option allows the user to view a report of the output of the NRT generation task. The edit option can be used to make changes to the NRTs before the NRTs are submitted to the nodes.

3. *Analyze the results.*

This step refers to the Network Builder *Analyze: Session Maintenance Simulation* task to determine how appropriate the network topology is, and whether the best values were selected for node tuning parameters.

If problems are found during this step, users can return to the appropriate configuration task to retune the configuration and provide a remedy for the problem.

4. *Place the configuration in service.*

This step refers to bringing the new configuration elements into service and populating the *StarKeeper II* NMS Core database and node databases with the configuration data.

StarKeeper II NMS commands are used to determine which modules and addresses remain out of service, to restore all out-of-service elements into service, and to load and synchronize databases throughout the network.

Trunk Configuration

The first stage of administering Session Maintenance in a new or existing network includes administration of any trunks selected for Session Maintenance. Administrators may use Network Builder to enter all the entities for one node at the same time, or enter all similar type network elements at one time. Familiarity with Network Builder will help in determining the best scenario to use. This section is concerned only with trunk administration for Session Maintenance.

Trunks are entered via the Network Builder *Configure: Trunk* task. When this task is selected, a window with the Trunk Configuration Form is displayed. This form allows users to identify trunks as participating in Session Maintenance and then specify values for the additional parameters for Session Maintenance.

An administrator using Network Builder to enable Session Maintenance on an existing trunk enters a slightly different scenario than when configuring new trunks. See the appropriate *StarKeeper II* NMS documentation for administration tips for using Network Builder to enable Session Maintenance on existing trunks.

Data for trunks that are to be included in the production network should be identified, but held until all data have been entered for the complete set of trunks being configured. Default values will usually suffice for all trunk parameters.

This guide is concerned only with identifying the trunk as a Session Maintenance trunk and the meaning of the relevant parameters. See the appropriate *StarKeeper II* NMS documentation for details about using the application.

The sets of trunk configuration parameters for Session Maintenance are associated with three different aspects of each trunk:

- end node specification
- trunk parameters
- specific Session Maintenance parameters

Each set is described below.

End Node Specification

For a Session Maintenance trunk, the node on each end of the trunk must be monitored by a *StarKeeper II* NMS to which the (Network Builder) workstation has access. When a trunk that will be configured for Session Maintenance is entered, two parameters must be specified for each end (node) of the trunk:

- *Node Type*
- *Node Name*

Trunk Parameters

In addition to the Node Type and Node Name parameters, the following parameters relate to the trunk as an entity:

- *Session Maintenance Enabled/Disabled.*

This parameter specifies whether a trunk will participate in Session Maintenance. Selecting *Enabled* indicates the trunk is a Session Maintenance trunk.
- *Active User Channels.*

This parameter specifies the number of active user channels the trunk will carry. The valid ranges are adjusted depending on the trunk type. For all trunk types, the value defaults to 125. The value chosen for this parameter should be based on known trunk usage. If no user channels are administered, this trunk can be configured for standby channel sets only.

Session Maintenance Parameters

The following parameters relate to Session Maintenance administration for the trunk:

- *Active Channel Sets.*

This parameter specifies the number of active channel sets residing on the trunk. The valid ranges depend on the trunk type and the number of user channels configured.

For the Trunk-PQ, the number of active channel sets is configured separately for CIR and non-CIR traffic. The expected bandwidth, however, is assumed to be equally distributed over all of the channel sets (both CIR and non-CIR). Additional channel sets can be configured for either CIR or non-CIR so that the expected bandwidth per channel set more accurately reflects the distribution between CIR and non-CIR traffic.

As with other trunks, each channel set, CIR or non-CIR, is considered to be a member of the same group. A CIR call is only set up on a CIR channel set and a non-CIR call is only set up on a non-CIR channel set.

If only one channel set is needed, consider putting in a second channel set so that, upon reroute, the traffic load can be distributed over additional reroute paths. The additional path reduces the traffic impact on any single trunk. This consideration should be balanced with the need to find a second reroute path if this trunk fails.

If the expected bandwidth per channel set on a given trunk is much higher than the network average utilization per channel set, add another active channel set to the trunk. This data can be obtained from the Engineering Data Report of the Session Maintenance Simulation task.

Trunks, channel sets, and node group administration are related. When a Session Maintenance trunk is assigned to a group, each channel set on the trunk is considered a member of that group. This practice is slightly different for a trunk without Session Maintenance, where the entire trunk is considered a member of the group.

There are implications for groups that use round robin call assignment. Consider an example in which two trunks with Session Maintenance, Trunk A and Trunk B, are assigned to the same group, group XYZ. Trunk A is configured with two active channel sets and Trunk B is configured with three active channel sets. Group XYZ consists of five members, as follows:

- Trunk A (channel set 1)
- Trunk A (channel set 2)
- Trunk B (channel set 1)
- Trunk B (channel set 2)
- Trunk B (channel set 3)

If group XYZ is optioned for round robin, during normal operation Trunk A will get two-fifths of the calls, and Trunk B will get three-fifths. The calls will not be split equally over the two trunks.

- *Standby Channel Sets.*

This parameter specifies the number of standby channel sets residing on the trunk. The default number of standby channel sets depends on the type of trunk and the number of active channel sets configured on that trunk. The Trunk-T3 defaults to 0, so a value must be entered for the trunk to participate in reroutes.

When possible, at least one standby channel set should be configured for every trunk. Configuring a standby channel set is not a requirement, but if none are configured, the trunk cannot be used as part of a reroute path.

- *Failure Declaration Threshold.*

This parameter specifies the number of consecutive seconds that must pass during which at least one keep-alive signal is lost before a trunk is failed. Trunk-T3 modules require no administration of this parameter. In choosing a value, consider the following:

- Is this trunk a "noisy" trunk?

- Is this trunk commonly used at or near maximum capacity?

If the answer to either question is yes, adjust the number of seconds to a value higher than the default (four seconds). A low failure threshold here may result in unnecessary reroutes of healthy trunks.

- What timers do end devices at the edge of the network have?

If the end devices utilizing this trunk have stringent timers, leave the threshold at four, or perhaps set it lower. Otherwise, provide a level of leniency on the trunk by setting the threshold higher than four.

- *Recovery Declaration Threshold.*

This parameter specifies the number of seconds that must pass with no loss of keep-alive signals before a trunk is declared healthy. Trunk-T3 modules require no administration of this parameter. In choosing a value, consider the history of that trunk. Is the trunk known to fluctuate between healthy and faulty? If so, a higher recovery threshold may be appropriate.

When a failed Session Maintenance trunk in the process of being rerouted recovers within 50 seconds, a reversion is performed automatically. Active channel sets being rerouted automatically revert to the original path. If an automatic reversion under these conditions is undesirable, set the recovery declaration threshold equal to or greater than 60 seconds.

- *Expected Bandwidth.*

The display field associated with this parameter shows the actual expected bandwidth in bits per second for the entire trunk.

The parameter specifies the percentage of total trunk bandwidth designated for the active channel sets; the remaining bandwidth may be considered for standby channel sets.

When configuring a Trunk-T3, the percentage of total trunk bandwidth to be allocated to connectionless traffic must be considered. For example, during trunk configuration of a Trunk-T3, suppose the expected bandwidth for connectionless traffic is set at 60%. Of the

trunk's total bandwidth, 40% now remains available for partitioning as channel sets for Session Maintenance. Moreover, some of this remaining 40% must be left available to support standby channel sets for the trunk to support reroutes.

Overhead must also be considered when configuring a Trunk-T3. Because of overhead, the 45 Mbps signaling rate of the Trunk-T3 has an effective carrying capacity of 33 Mbps for connectionless traffic, and 18 Mbps for connection-oriented traffic.

When configuring a Trunk-T3 for Session Maintenance, the values entered for the percentage of bandwidth used for connectionless and connection-oriented traffic represent the percentages of the trunk's total capacity actually used by the traffic. In other words, the expected bandwidth values are the actual effective bandwidths divided by the maximum possible effective bandwidths. These values are *not* the same as the effective throughput divided by 45 Mbps.

Suppose, for example, that the effective connectionless bandwidth on a Trunk-T3 is expected to be 8.25 Mbps and the effective connection-oriented active channel set bandwidth is expected to be 9 Mbps. The following information shows how the percentage values for the expected bandwidth parameter for the trunk are derived by dividing the expected rate by the maximum effective rate for that type of traffic.

Traffic Type	Effective Rate	Maximum Effective Rate	Percentage
Connectionless	8.25 Mbps	33 Mbps	25
Connection-oriented active	9.00 Mbps	18 Mbps	50
Connection-oriented standby	4.50 Mbps	18 Mbps	25

To set the parameter, the administrator specifies that 25% of the traffic is used for connectionless traffic and 50% is for connection-oriented active channel sets. The BNS-2000 software will infer that the remaining 25% is used for standby channel sets. The percentages add up to 100%, but the effective bandwidths do not add up to 45 Mbps.

For the Trunk-PQ, the expected bandwidth should be the sum of the expected bandwidth for CIR traffic and non-CIR traffic.

Expected bandwidth should be based on utilization. When determining the bandwidth values throughout your network, use a consistent concept, such as busy hour, daily average, or weekly average. This parameter value is for bookkeeping, and is used in the negotiation process for finding sufficient bandwidth during a reroute. The full bandwidth of the trunk is available to the sessions on the trunk at all times.

- *Primary Node.*

This parameter specifies the primary node for the trunk. An important criterion to consider when specifying the primary node is that it should be the least connected node of the node

pair for the trunk. When the primary node is the least connected, the likelihood of finding successful reroute paths is improved and a single node is not responsible for checking the quality of many trunks, thus providing the best chance for a successful reroute.

Data supporting whether the least connected node was selected can be verified through the Engineering Data Report of the Session Maintenance Simulation task. If the Engineering Data Report indicates that the primary node for a trunk is the most connected node, consider changing the primary node, remembering that all parallel trunks must have the same primary node. The best way to proceed is to first remove all parallel trunks between the two nodes from service, and then respecify (change) the primary node for one of the trunks. The change will automatically be applied to all trunks involved. Restore all trunks to service.

Node Reroute Table Configuration

Node Reroute Tables are configured (generated) by the Network Builder application from configuration data entered for nodes and trunks (see **Node Reroute Tables** in Chapter 2). Network Builder also provides the capability to edit existing NRTs.

The Network Builder application treats the set of network NRTs as a single unit. Using the Network Builder *Configure: Node Reroute Tables* task and opening the NRT Configuration Form provides access to the entire set of NRTs for the network. Within the form, administrators can gain access to:

- the NRT for a given node
- a neighbor node within the given node's NRT
- a list of the assisting nodes (for editing)
- a list of the trunks that includes module addresses (for editing)

These Network Builder capabilities permit viewing existing NRTs, changing fields within NRTs (via insert, delete, or reordering functions), and generating new NRTs for the network. Other capabilities permit administrators to view:

- NRTs using either committed and pending data or committed data only
- a generation report (the list of the NRTs for the nodes)
- a task log

See the appropriate *StarKeeper II* NMS documentation for details on how to use the application to generate NRTs and for editing instructions.

Editing NRT Assisting Node and Trunk Lists

Administrators may want to edit the order of assisting nodes and the list of trunks included in NRTs generated by Network Builder for the following reasons:

- *Trunk speed.*

In creating the list of assisting nodes for a given node, the application may have ordered the faster trunks last. Administrators may want to change the list so that the first choice for a reroute path is through an assisting node connected via a high-speed trunk.

- *Trunk type.*

The first-listed assisting node may be connected by a wire trunk with the capacity for four channel sets. Other assisting nodes in the list may be connected via fiber trunks with the capacity for up to 16 channel sets. The larger trunk capacity may imply a greater number of standby channel sets, thus increasing the potential success of a reroute through that assisting node.

- *Unstable node.*

The first-listed assisting node may be experiencing some type of problem. It can be moved to the bottom of the list until the problem is solved.

- *Busy node.*

Control Computer or backplane utilization on the first-listed assisting node may be high because of a specific network application. It may not be the best node through which to attempt reroutes.

Node Configuration and Tuning

Another stage of configuring a network with Session Maintenance includes tuning node configurations by adjusting several node parameters on nodes participating in Session Maintenance. Network Builder can be used to adjust these parameters to provide flexibility for some basic node capabilities, fine-tuning these node-wide Session Maintenance parameters for all existing nodes in the network. The adjustment of these settings is optional; default values are available and will be used by the system if no adjustments are made.

Administrators can either tune the node parameters in advance, or await the results of a Network Builder simulation before tuning the parameters. The defaults can be used initially with fine-tuning done later, after the simulation. *StarKeeper II* NMS documentation describes how to use the *Configure: Node* task to enter a node's configuration data. This section is concerned only with the node tuning parameters for Session Maintenance.

Four specific parameters are administrable for Session Maintenance from within the Network Builder *Configure: Node* form:

- *Assisting Node Status Messages*
- *Reroute Request Withdrawal Time*
- *Additional Reroute Requests*
- *Reroute Bandwidth Reduction*

The parameters are discussed below. The defaults should be sufficient at first for everyone, and, in most cases, should suffice for normal operation of the node. The defaults are automatically selected, so it is not necessary to interact with this part of the form if the defaults are acceptable.

Assisting Node Status Messages

The Assisting Node Status Messages parameter can be enabled or disabled. A node participating in a reroute as an assisting node has the capability to send messages to *StarKeeper II* NMS. This parameter controls whether the node will issue these status messages. Assisting node status messages may be useful when monitoring Session Maintenance performance, but they can be turned off at any time to reduce the number of messages sent in real time.

The numerous messages generated when a reroute occurs can be useful in tracking the direction and progress of a reroute, but in normal operation the messages produced by the primary and secondary nodes are sufficient to track reroutes. Therefore, assisting node status messages should be Disabled for normal operations of Session Maintenance. They should be Enabled for the following reasons:

- for initial testing and tracking of the feature
- to track reroutes when a problem is suspected

Reroute Request Withdrawal Time

The Reroute Request Withdrawal Time parameter provides each primary node with the capability to set a reroute request withdrawal time in seconds. The node will withdraw any unanswered reroute requests after the specified time has elapsed. This parameter controls the time in seconds until a reroute request is withdrawn. All outstanding requests for reroute will be withdrawn after this time period following initial invocation. Calls on channel sets that do not have a reroute path will be disconnected. The parameter defaults to 15 seconds.

Additional Reroute Requests

The Additional Reroute Requests parameter specifies the percentage of additional reroute requests the node issues for a reroute. Primary nodes have the capability to issue a percentage of additional reroute requests above what is needed. Issuing additional requests increases the probability that all active channel sets will be successfully rerouted. This parameter controls the percentage of additional reroute paths requested by the primary node above what are needed by the failed trunk. All reroute requests may not be effective, so additional initial reroute requests will increase the probability of successful reroutes.

The value of this parameter can be adjusted after running the Simulator. Adjusting this value upward will increase the number of reroute requests issued by the primary node. The value should be the number that gives you the highest success rate, as determined by the Simulator.

Note that the additional request factor is applied to the remaining active channel sets only after trunks parallel to the failed trunks have been exhausted.

Reroute Bandwidth Reduction

The Reroute Bandwidth Reduction parameter specifies an acceptable amount of reduced reroute bandwidth. For primary nodes or nodes acting as assisting nodes, the parameter provides the capability to specify an acceptable percentage of bandwidth reduction on reroute paths. The specified percentage is relative to the estimated amount used by each active channel set being rerouted. Reduced reroute bandwidth increases the probability that the node will find a successful reroute path for more active channel sets at the expense of higher data transport delay on the reroute trunks that are chosen.

If more bandwidth is needed than the available standby bandwidth provides, the primary or assisting node will send the needed requests on trunks that have standby bandwidth amounting to the required bandwidth minus the specified percentage. For example, if the required bandwidth is 100 Kbps and the Reroute Bandwidth Reduction value is 20%, the node will send reroute requests to trunks that have from 99 to 80 Kbps of available standby bandwidth if trunks with 100 Kbps of standby bandwidth are not available.

The value chosen here is a consequence of the criterion used when expected bandwidth values are selected during trunk administration. For example, if busy hour was the criterion used, you may wish to increase the acceptable amount of bandwidth reduction since chances of needing a reroute during a busy hour are slim. If either daily or weekly average was used, however, traffic may be above or below the specified expected bandwidth at any given time. Therefore, you will not want to deviate from the expected bandwidth by more than 20 or 30%. Note that bandwidth reduction is attempted only after all paths with full bandwidth are considered.

Consider the following important points with this parameter:

- Use the same value for this tuning parameter on every node throughout the network. Because each node in the path of a reroute uses its own administered value for bandwidth reduction, a consistent approach throughout the network is needed to ensure that this parameter will work as designed.
- This parameter allows one channel set to be rerouted on trunks with bandwidth reduction. That is, when all paths with full bandwidth are considered, and then additional requests are sent on trunks that meet the reduced bandwidth requirement, only one additional channel set may be rerouted when a trunk that meets the reduced requirement is found. Any remaining channel sets are not rerouted.
- Use this parameter to determine the effectiveness of network engineering for reroutes. If simulations repeatedly show that successful reroutes depend on varying this parameter, it may be time to re-engineer the network. If your expectations for performance are met with a certain parameter value, re-engineer the network to match the successful simulation.

Simulation and Analysis

Network Builder provides a simulation tool that models how a network of nodes deploying Session Maintenance is expected to behave if trunks fail. The Simulator's output contains information that can guide tuning and replanning efforts for Session Maintenance trunks.

Every effort has been made to model the actual node behavior as realistically as possible. There are some real-life dependencies, however, that may result in an actual solution different from the simulated result.

A good time to use the Simulator would be when the following configuration activities have been accomplished:

- after all the appropriate trunks have been configured for Session Maintenance (and the data has been put on hold)
- after NRTs have been generated for the network (and the data is being held)
- when the configuration was entered using default values

After these procedures are complete, run a simulation using the pending data and other defaults provided with the Simulator. If the Summary Report (explained later in this chapter) yields 100% success at rerouting active channel sets from the failed trunk(s), Session Maintenance can be placed in service for the network.

If the Simulator run produces different results, however, there may be a need to review the output for indications of what can be done to make Session Maintenance more robust throughout the network. The purpose of this section is to explain implications of the output to ensure 100% rerouting under most conditions, or possibly reconfigure certain parameters to optimize the performance of Session Maintenance.

Simulator Application

The Simulator Application is a Network Builder tool that simulates the performance of Session Maintenance by providing a model of channel set rerouting throughout the network based on network trunk failure(s). The application uses node and trunk configuration data supplied via Network Builder configuration tasks. The Simulator can be run to predict the behavior of the network as configured. In addition, the Network Builder capability to *hold* pending data before submitting it to the network is utilized by the Simulator to predict behavior of a theoretical network that would incorporate the pending configuration changes.

Regardless of which set of data is used for the simulation, the process is characterized by events occurring at discrete time intervals based on node and trunk delays. Users create the events by failing trunk(s) in various modes that are designed to provide realistic models of facility failures that may occur in the network. Reports based on the simulated rerouting events are generated and are available for analysis, allowing administrators to view rerouting success or to deduce other potential problems that might affect service in the working network.

Using the Simulator

The Simulator allows the user to choose parameters for the simulation and then view the results. It is invoked from the Network Builder control window through the *Analysis: Session Maintenance Simulation* task.

Failure Mode

Even when a sequential Simulator run is 100% successful, look at specific concurrent trunk failures on trunks that follow the same geographic route through the network.

Additional Reroute Requests

You can respecify the percentage of additional reroute requests the primary node makes. You have the option of specifying a different start and stop value in 10% increments.

Reroute Bandwidth Reduction

This parameter controls the percentage by which requests for bandwidth can be reduced when sufficient bandwidth is not available. Reduced bandwidth requests are sent only after all trunks with sufficient bandwidth have been exhausted. The primary or assisting node will then send requests over trunks with standby bandwidth that is at least equal to the required bandwidth reduced by the specified percentage.

Interpreting the Output

The Simulator provides three output reports that are based on the results of the Simulator run:

- Network Summary Report
- Detailed Report
- Engineering Data Report

See the *StarKeeper II NMS Network Builder Guide* for examples of these reports.

Network Summary Report

The simulation run can use either administered tuning parameter values or overridden values for two parameters:

- *Additional Reroute Requests*
- *Reroute Bandwidth Reduction*

When the simulation run uses administered tuning parameter values, the Network Summary Report produces one single row of information, even when several trunks are chosen. When

multiple trunks are individually specified, each trunk is listed explicitly. The single row of data output represents a compilation of the outcome of routes for the individually specified trunks that were failed during the simulation run.

When the simulation run uses overridden tunable parameters, multiple lines of data for the parameters are output in addition to values for the overridden tuning parameters. For example, if administered tuning parameter values were overridden with Additional Reroute Requests ranging from 0 to 30% and Reroute Bandwidth Reduction ranging from 0 to 20%, there would be a total of twelve lines of output corresponding to the twelve combinations (0, 10, 20, 30 Additional Reroute Requests) and (0, 10, 20 Reroute Bandwidth Reduction). Output fields in the Network Summary Report are discussed below.

- *Failure Mode.*

This column shows which failure mode was used. See **Failure Mode** in the previous section.

- *Failed Trunks.*

This column shows the ordered list of trunks failed. If all the trunks were failed, this field will say "ALL."

- *Additional Requests.*

This column shows the percent of additional channel sets requested. If "Override Node Tuning Data" was not selected, this field is left blank. See **Additional Reroute Requests** in the previous section.

- *Bandwidth Reduction.*

This column shows the percent of reduced bandwidth. If "Override Node Tuning Data" was not selected, this field is left blank. See **Reroute Bandwidth Reduction** in the previous section.

- *Channel Sets Needed.*

This column shows the number of channel sets needed. If the report is for a single trunk failure, this column represents the active channel sets on that trunk. Otherwise, this column represents the sum of all the active channel sets on all the selected trunks.

- *Channel Sets Successful.*

This column shows the number of channel sets that were successfully rerouted.

- *Requests Sent.*

This column shows the number of reroute requests sent by the primary node(s). This number is a function of the number appearing under the Channel Sets Needed column (see above) and the value specified for Additional Reroute Request parameter.

- *Requests Rejected.*

This column shows the number of requests rejected at the secondary node as being superfluous. These requests followed a valid reroute path, but other requests made it to the secondary node first.

- *Requests Dead.*

This column shows the requests that followed a path that did not end up at the secondary node. The detailed report illustrates the reason.

- *Success Percentage.*

This column shows the reroute success as a percentage of the combination of the Additional Reroute Requests and Reroute Bandwidth Reduction parameters for the specified trunks. A good run is where 100% of all the Channel Sets Needed are successfully rerouted.

- *Average Hops.*

This column shows the average number of hops.

Detailed Report

If, when reviewing the Network Summary Report, the Success Percentage is not 100%, you may wish to review the Detailed Report. The first part of the Detailed Report gives the trunk name, primary node, and the secondary node values for a failed trunk, followed by the same fields in the Network Summary Report. The second part provides full reroute path definition as well as the result of each individual reroute request. By scanning the Result column, you can determine the reasons for failure. These reasons are the same as those which would occur in an actual live reroute.

The following list describes the output fields for the second part of the Detailed Report.

- *Request.*

This field identifies the request number. There is a maximum of 32 requests, so there could be up to 32 request numbers in each report. A more likely number is less than ten.

- *Result.*

The result could be either Accepted, Rejected, or a reason code that specifies why the request did not succeed. Valid reason codes are Insufficient Bandwidth, No Channel Set, Hop Limit, and Drop. The Drop reason will not occur for sequential trunk failure simulations. The Drop reason indicates a previously rerouted channel set has been dropped when the standby trunk itself was failed. No corrective action is needed. See Table 5-2 for explanations and recommended actions for all other results.

- *Path Definition.*

The remaining fields represent the reroute path. Each row represents a hop. The reroute could go over up to four hops, so there could be up to four rows per reroute path.

Engineering Data Report

The Engineering Data Report includes calculations that are useful for checking reroute path load balancing and expected traffic usage on channel sets during a reroute. It can be used to replan Session Maintenance trunks and trunk parameters.

Users may want to scan the Engineering Data Report for unusually high values for Expected Bandwidth/Active Channel Sets and an over-connected primary node. Neither condition by itself is bad. Either may indicate potential bottlenecks, however, and provide data that will help Session Maintenance planning. Four fields are provided for each trunk name, as explained below.

- *Expected Bandwidth/Active Channel Set.*

This column should be scanned in search of unusually high or low values. A trunk with an unusually high Expected Bandwidth (BW)/Active Channel Sets (CS) ratio may have trouble finding sufficient bandwidth for reroute paths. Adding more channel sets could alleviate the problem. For example, a Standard Fiber Trunk (SFT) configured with four active channel sets and 50% Expected Bandwidth would show a value of 1000 Kbps Expected BW/Active CS. A bandwidth requirement this large might have trouble finding standby channel sets that provide sufficient bandwidth. This trouble can be anticipated especially if the report indicates no similar ratios elsewhere in the network. Adding eight active channel sets (for a total of 12) reduces the value of Expected BW/Active CS to 333.3 Kbps, a bandwidth requirement that can be handled more successfully in most networks.

A low ratio could be an indication of too many active channel sets being defined. This condition could result in more reroute requests than necessary. For example, a Standard Wire Trunk (SWT) with a line speed of 64 Kbps configured with four active channel sets and 50% Expected Bandwidth would show a value of 8 Kbps Expected BW/Active CS. This is a fairly low bandwidth requirement. Finding standby channel sets that have sufficient available bandwidth should be no problem in most networks. Since four active channel sets are configured, however, the node will issue six reroute requests. The unusually low value would provide a tip that reconfiguring the trunk is appropriate. If the same trunk is configured for two active channel sets, only three reroute requests are issued.

The expected bandwidth/active channel set can be adjusted for the Trunk-PQ to more accurately reflect the breakdown between CIR and non-CIR traffic. Configure the appropriate number of channel sets for each traffic type so that the ratio of channel sets is the same as the ratio of traffic.

- *Active Channels/Active Channel Set.*

Scan this column for unusually high or low numbers of active channels per active channel set. Determine if this channel per channel set allocation is appropriate. If this ratio is not appropriate, add or subtract active channel sets accordingly.

- *Total Standby Bandwidth.*

This column is useful in determining why reroutes did not behave as expected. If the Detailed Report shows reroutes (whether successful or unsuccessful) following unexpected reroute paths, it may indicate insufficient standby bandwidth on the expected path. A quick check of the standby bandwidth on the trunks composing the expected reroute path will determine if sufficient standby bandwidth has been configured. If not, you may reconfigure the trunk accordingly.

- *Reroute Node Connectivity.*

This column indicates whether the primary node for this trunk has fewer or more Session Maintenance trunks to the network than the secondary node. When the primary node has more, the difference is appended to the row. Consider making the primary node the least connected.

Administration

Reroute Administration	5-5
When Reroutes Are Invoked Automatically	5-5
When Reroutes Are Invoked Manually	5-7
When a Reroute Is Successful	5-7
When a Reroute Is Partially Successful	5-8
When Secondary/Assisting Nodes Participate in Reroutes	5-11
Reversion Administration	5-14
When the Failed Trunk Has Recovered	5-14
When a Reversion Is Requested	5-15
When a Reversion Has Succeeded	5-15
When a Reversion Has Failed	5-16
When an Assisting Node Has Participated in a Reversion	5-16
When a Trunk-PQ Downloads Prior to Reversion	5-17
When the Node Drops Channel Sets	5-17
When PDDs Are Restored	5-18
StarKeeper II NMS Programmers Interface	5-19
Automatic Request for Reroute Status	5-19
Performing an Automatic Reversion	5-21

Administration

The configuration activities described in Chapter 4 represent one aspect of Session Maintenance administration. After configuration is complete, node parameter values control the automatic rerouting process. NRTs generated by Network Builder and the availability of standby channel sets control reroute path definitions. Administration may also be necessary when real-time Session Maintenance events occur, whether they are initiated automatically or manually. Working networks may require steady-state administration and a level of ongoing management because traffic loads have changed, new resources have been configured, or a persistent problem is causing service adversity. For example, administration activities could include the following:

- initiating reroutes manually, usually for testing purposes
- displaying and reviewing the effects of an automatic reroute
- reviewing the number and duration of trunk failures in the network, and the network's success rate in rerouting all the affected channel sets
- reverting the network to its normal routing when failed trunks are once again healthy
- steady-state performance monitoring and replanning based on empirical data available from the working network
- optimizing rerouting to make best use of the feature

When a trunk configured for Session Maintenance fails and the rerouting process occurs, report status messages associated with each event are, or can be, generated at each node affected by the reroute. For example, the following messages may be generated:

- a status message that indicates the detection of a trunk failure and the beginning of an automatic reroute
- a status message that reports the completion of a reroute

This is not a complete list of potential messages. The point is that administrators can use any of the messages reported to gain information about how the network behaves during Session Maintenance rerouting events, and administrators can take any of the following actions:

- capture the message and take no action
- capture the message and perform some manual operation for administrative purposes
- capture the message so that some user-programmed action can be initiated to perform an automatic administrative operation

These actions apply not only to administration related to the events reported by various messages, but also to operations based on the level of administration desired. That is, administrators also need to decide whether to react to messages generated on primary, secondary, or assisting nodes. Table 5-1 shows message numbers and their relationship to Session Maintenance events. The table also indicates those messages that are cleared automatically by *StarKeeper II* NMS, and those that must be cleared by an administrator. Additional information related to each message is referred to in the table.

TABLE 5-1. Messages and Session Maintenance Events

Message Number	Session Maintenance Event	Occurs with	Auto-cleared by <i>StarKeeper II</i> NMS
7103	Automatically Invoked Reroute	Automatic Reroute	When reroute is completed
7105	Unsuccessful Reroute	Automatic/Manual	At reversion
7106	Reversion Failed	Reversion	No
7107	Rerouted Channel Set Dropped	N/A	No
7601	Manually Invoked Reroute	Manual Reroute	Auto-clearing
7602	Already Rerouted	Automatic/Manual	Auto-clearing
7603	Reversion Requested	Reversion	Auto-clearing
7604	Reroute Unnecessary	Automatic/Manual	Auto-clearing
7605	Successful Reroute	Automatic/Manual	At reversion
7606	Secondary Node Assigned Paths	Automatic/Manual	Auto-clearing
7607	Assisting Node Participating in Reroute	Automatic/Manual	At reversion
7608	Assisting Node Failed to Participate in Reroute	Automatic/Manual	Auto-clearing
7609	Trunk Recovered	N/A	At reversion
7611	Reversion Succeeded	Reversion	Auto-clearing
7612	Assisting Node Participating in Reversion	Reversion	Auto-clearing
7615	CIR Calls Dropped	Reversion	No

The presentation in this chapter first covers reroutes, then reversion, and includes adding automatic operations through capabilities provided in the *StarKeeper II* NMS Programmer's Interface. The Programmer's Interface does not provide customization scripts, only the interface for them. Users will have to provide the customization scripts. Sample scripts are provided later in this chapter.

Reroute Administration

All reroutes generate report and status messages, whether the reroute is initiated automatically by the node in response to a trunk failure, or manually from an administration console. The following sections discuss status messages that are related to rerouting events, including messages that are sent when the reroute is invoked and messages sent later that indicate if the reroute was successful or not. These messages provide keys to identifying the behavior of the working network and include recommended actions that can be used to direct Session Maintenance administration.

Subsections discuss messages and events by the following topics:

- Automatic Reroute Invocation
- Manual Reroute Invocation
- When Reroutes Are Successful
- When Reroutes Are Unsuccessful
- Node Participation Status Messages

When Reroutes Are Invoked Automatically

Automatic rerouting is initiated by the node's control software when a trunk failure is detected. There are four ways a Session Maintenance reroute can be invoked automatically:

- loss of keep-alive signal because of a facility failure
- module failure
- automatic removal of errored trunks (if this option has been selected)
- removal of shelf or shelf failure

When a reroute has been initiated by a loss of keep-alive because of a facility (not module) failure, the trunk is left in service. Administrators can remove it from service to perform diagnostics and other troubleshooting without dropping any of the calls on the rerouted channel sets. Of course, the rerouted calls can not be reverted back to the module until it is restored to service and regains a healthy trunk status.

Module failure refers to anything that would put the module into a fault state, such as hardware failure, taking the board out of its slot, disconnecting the I/O connector, or toggling the faceplate switch (DISABLING). If the module is a trunk module using Session Maintenance, the fault state would cause the initiation of a reroute. If an errored Session Maintenance trunk is automatically removed from service, a reroute would also be invoked.

Under normal conditions, when a module fails for any of the previously mentioned reasons, it is automatically and immediately removed from service. If the secondary node trunk module fails, the disabled module may not be removed from service for one to three minutes to allow ample time for the primary node to detect a failure and initiate a reroute on its own.

The following message is associated with automatically invoked reroutes: 7103 — Trunk Failure Report Alarm.

Trunk Failure Report Alarm (7103)

The message shown in Screen 5-1 (and others elsewhere in the chapter) represents a node console view of a message sent from the node. (The message has a different format in *StarKeeper II* NMS.) It reports that failure of a Session Maintenance trunk has been detected by the primary node, and the process of automatically rerouting the trunk has begun.

This message is generated by the primary node of the failed trunk. It will be followed by a message indicating the success or failure of the reroute.

Other than ensuring that the alarm is logged, no action is recommended because data collected on a reroute in progress would have little value for administration purposes.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
* 7103 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
  REPORT ALARM: trunksig: Session mtce trunk failure, attempting reroute.
```

SCREEN 5-1. Trunk Failure Report Alarm—Session Maintenance Enabled

The messages shown in this chapter contain common variable fields, which are explained below.

- *MODADDR*=<nnn>
This is the one-, two-, or three-digit number that denotes the slot in the backplane for the module on the node to which this alarm refers.
- *MODTYPE*=<type>
This is the string that denotes the module type.
- *EVENTID*=<node.trunk>
This is the unique identity of the trunk being rerouted in the form of `primary_node.node_trunk_address`.

When Reroutes Are Invoked Manually

Although rerouting is automatically initiated upon trunk failure, it can also be initiated by a node administrative command from a console or as a pass-through command issued from *StarKeeper II* NMS for administrative reasons. Administrators can enter the **route trunk alternate** command for initial testing of the network, or to perform diagnostics on trunks without dropping the calls active on that trunk.

The **route** command is documented in the *Trunk Module Reference*. The message associated with manual reroutes is: 7601 — Manual Reroute Requested Status Message. This message is discussed in the following section.

Manual Reroute Requested Message (7601)

When a reroute of a Session Maintenance trunk is requested by means of the **route trunk** command, a status message appears at the primary node for the trunk being rerouted. It will be followed by a message indicating that the reroute has been completed or has failed. There is no recommended action.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>  
7601 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>  
REPORT STATUS: trunksig: Reroute of session mtce trunk manually requested.
```

SCREEN 5-2. Manual Reroute Requested Status Message

When a Reroute Is Successful

It is important to generate status reports of the network after a trunk has been rerouted successfully. For this reason, the message generated when a reroute is successful can be used as the key to know when to invoke *StarKeeper II* NMS commands for status reports. The message associated with this event is: 7605 — Successful Reroute Report Alarm.

Successful Reroute Report Alarm (7605)

A message appears at the primary node for the trunk being rerouted when either an automatic or manual reroute of a Session Maintenance trunk has succeeded, with all active channel sets rerouted successfully.

A recommended action is to execute a *StarKeeper II* NMS **smstat** command for the primary node to investigate the path of the reroute. The **smstat** command can be executed on a *StarKeeper II* NMS Core System, through the *StarKeeper II* NMS Network Monitor Diagnose screen, and with a Programmer's Interface script.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
** 7605 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT ALARM: trunksig: Session mtce trunk reroute successful.
```

SCREEN 5-3. Successful Reroute Report Alarm

When a Reroute Is Partially Successful

Partially successful reroutes generate messages that can be used to investigate the status of Session Maintenance via the *StarKeeper* II NMS **smstat** command. The message related to this event is: 7105 — Reroute Unsuccessful Report Alarm. This message is discussed in the following section.

Reroute Unsuccessful Report Alarm (7105)

A Report Alarm is issued when a recently initiated reroute of a Session Maintenance trunk has failed either partially or completely. This situation can arise in both manually and automatically invoked reroutes. With a complete failure, all calls are dropped. With a partial failure, calls using channel sets for which no reroute paths can be found are dropped. Channel sets for which no reroute paths are found will remain unusable until the trunk is returned to normal routing with the **route trunk** command.

In the message, the line that indicates the reason for reroute requests failing may appear several times if there were several causes for the reroute failures.

This message appears at the primary node for the trunk being rerouted. A recommended action is to issue a *StarKeeper* II NMS **smstat** command to get the status of the failed reroute. The **smstat** command can be executed on a *StarKeeper* II NMS Core System, through the *StarKeeper* II NMS Network Monitor Diagnose screen, and with a Programmer's Interface script.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
** 7105 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT ALARM: trunksig: Session mtce trunk reroute not successful.
<n> reroute requests sent for <n> channel sets; <n> successful.
<n> reroute requests failed due to <reason>.
Calls are dropped.
```

SCREEN 5-4. Reroute Unsuccessful Report Alarm

Variable fields unique to this message are described below.

■ *<n>*

This is an integer indicating the number of reroute requests sent on behalf of the active channel sets on the failed trunk, the number of active channel sets in the trunk being rerouted, the number of active channel sets which have been successfully rerouted, or the number of reroute requests that failed, respectively. The number of reroute requests sent is greater than or equal to the number of active channel sets in the trunk being rerouted, unless there are not sufficient available standby channel sets or bandwidth on other trunks connected to the primary node.

■ *<reason>*

This is the reason for the failure of one or more requests. The recommended action(s) vary and depend primarily on the reasons given for the reroute failures. The reasons may indicate ways that the network topology or configuration of trunks and channel sets could be changed to ensure the success of future reroutes; see Table 5-2. Output from the **verify trunk** command will indicate if all the reroute requests failed.

TABLE 5-2. Failure Reason Messages

Reason	Explanation	Recommended Action
No Channel Sets	There are no available standby channel sets available on the primary node or on the assisting nodes.	If enough standby channel sets are configured on other trunks, determine if other reroutes were in effect at the same time that would have been using those standby channel sets, or determine if more standby channel sets should be configured to support reroutes between the primary and secondary nodes; see Trunk Parameters in Chapter 4. Determine if forwarding of this request would result in a loop condition.
Insufficient Bandwidth	There is insufficient free bandwidth along trunks with free standby channel sets.	Determine if the bandwidth reduction percentage is adequate for the entire network. Run a simulation and check to see if node tuning parameters need to be adjusted; see Using the Simulator in Chapter 4.
Excessive Hopcount	The reroute request passed over four trunks without reaching the secondary node. It can be normal for this reason to appear; it could be generated for superfluous additional reroute requests.	Check the NRT. Only one-hop neighbors and two-hop neighbors should be listed in the NRT. Consider regenerating the NRTs; see Node Reroute Table Configuration in Chapter 4.
Trunk/Shelf Out of Service on Secondary Node	The trunk to be rerouted is out of service on the secondary node and the reroute cannot be completed.	<ol style="list-style-type: none"> 1. Enter remove trunk to take the failed trunk out of service. 2. Follow procedures in the Troubleshooting chapter of the <i>Trunk Module Reference</i> to determine appropriate action. 3. If necessary, replace the trunk; see the Installation chapter of the <i>Trunk Module Reference</i>. 4. Restore both ends of the trunk. 5. Restore shelf. 6. Enter route trunk to return the trunk to normal routing.
Secondary Node Not in NRT	The secondary node is not listed in the NRT of one of the nodes involved in the reroute attempt.	View or regenerate the NRTs for the reroute nodes using Network Builder.
Bad CS State	A standby channel set was temporarily unable to process a reroute request.	No action.

When Secondary/Assisting Nodes Participate in Reroutes

When secondary and assisting nodes participate in reroutes (when the capability is enabled), they send messages indicating the status of reroute events. These messages may be useful for administration, although they are not as important as reroute invoked or reroute status messages on the primary node. The messages included in this category are:

- 7606 — Secondary Node Assigned Reroute Paths
- 7607 — Assisting Node Participating
- 7608 — Assisting Node Failed to Participate
- 7602 — Already Rerouted
- 7604 — Reroute Is Unnecessary

Secondary Node Assigned Reroute Paths Status Message (7606)

When a reroute of a Session Maintenance trunk has been completed, a message appears at the secondary node of the trunk. The message appears whether or not all of the channel sets have been rerouted.

No action is recommended.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
7606 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT STATUS: trunksig: Secondary node assigned reroute paths.
```

SCREEN 5-5. Secondary Node Assigned Reroute Paths Status Message

Assisting Node Participation Report Alarm (7607)

When a node is participating in the reroute of a Session Maintenance trunk, an alarm message appears at an assisting node for each reroute path. This message appears only if the Assisting Node Status Messages Enabled parameter has been set.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
* 7607 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT ALARM: trunksig: Assisting node participated in a reroute.
Reroute uses trunk <tk1> from node <nd1> and trunk <tk2> to node <nd2>.
```

SCREEN 5-6. Assisting Node Participation Report Alarm

Variable fields unique to this message are described below.

- *MODADDR=<nnn>*
the address of the module that received the reroute request
- *<tk1>*
the trunk along which the reroute request arrived at this assisting node
- *<nd1>*
the name of the node which forwarded the reroute request to the node
- *<tk2>*
the trunk along which this node has forwarded the reroute request
- *<nd2>*
the name of the node to which this node has forwarded the reroute request

Assisting Node Failed to Participate Status Message (7608)

A status message appears when a node has received a request to participate as an assisting node in the reroute of a Session Maintenance trunk but cannot participate. This message appears only if the Assisting Node Status Messages Enabled parameter has been set. The message by itself is no cause for concern; it could be sent because extra reroute paths were requested but not used.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
7608 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT STATUS: trunksig: Assisting node failed to participate in a reroute.
Reason for failure: <reason>
```

SCREEN 5-7. Assisting Node Failed to Participate Status Message

The variable field unique to this message is explained below.

- *<reason>*

The reason for the failure of one or more requests. The recommended action(s) vary and depend primarily on the reasons given for the reroute failures. The reasons may indicate ways that the network topology or configuration of trunks and channel sets could be changed to ensure the success of future reroutes. Valid reasons and an action to take in each case are described in Table 5-2.

Already Rerouted Status Message (7602)

When a Session Maintenance trunk fails after it has already been rerouted by means of the **route trunk** command, the following message appears at the primary node for the trunk being rerouted.

There is no recommended action.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
7602 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT STATUS: trunksig: Session mtce trunk already rerouted.
Failure will not cause further reroute.
```

SCREEN 5-8. Trunk Already Rerouted Status Message

Reroute Unnecessary Status Message (7604)

Sometimes an unnecessary reroute of a Session Maintenance trunk is attempted because the apparent failure of the trunk is in fact a secondary node Switch failure. A switchover to the secondary node redundant Switch module occurs and the reroute is aborted. The trunk remains normally routed, and all calls remain up. This message appears at the primary node for the trunk being rerouted.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
7604 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT STATUS: trunksig: Session mtce reroute aborted: switchover.
```

SCREEN 5-9. Reroute Unnecessary Status Message

Reversion Administration

Reversion administration refers to the process of restoring rerouted channel sets back to the original path—that is, back to the trunk in use before the reroute occurred. This reversion is possible at any time as long as the trunk modules and shelves at either end are in service, but not recommended until after the failed trunk is healthy again. A healthy trunk is one that passes the real-time trunk testing that is performed continually by the node's Control Computer.

Administrators can perform reversion via the **route trunk normal** command from the primary node. There are several ways to execute the command.

- from a node console
- from a *StarKeeper* II NMS Core System console
- from the *StarKeeper* II NMS Network Monitor Diagnose screen
- automatically by a command script that interfaces to *StarKeeper* II NMS Programmer's Interface and is activated when a message that indicates the failed trunk has recovered is recognized

Reversion is covered in the following sections by first discussing the trunk status healthy message, and then the effect and outcome of issuing the **route trunk** command. Messages associated with reversion administration are:

- 7609 — Trunk Status Healthy Message
- 7603 — Reversion Requested Message
- 7611 — Reversion Successful
- 7106 — Reversion Unsuccessful
- 7612 — Assisting Node Participated in a Reversion
- 7615 — Trunk-PQ Drops CIR Calls During Reversion

These messages are discussed in the following sections.

When the Failed Trunk Has Recovered

The trunk status healthy message appears at the primary node of a failed and rerouted Session Maintenance trunk when the trunk has recovered its data transmission integrity and can safely be reverted.

The recommended action is to revert the trunk to normal routing with the **route trunk** command at the primary node.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
* 7609 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT ALARM: trunksig: Session mtce trunk has recovered.
Issue the 'route trunk normal' command.
```

SCREEN 5-10. Trunk Recovery Report Alarm

When a Reversion Is Requested

When reversion of a Session Maintenance trunk to normal routing is requested by means of the **route trunk** command, a message indicating the success or failure of the reversion appears at the primary node for the trunk being reverted.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
7603 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT STATUS: trunksig: Reversion of session mtce trunk requested.
```

SCREEN 5-11. Reversion Requested Status Message

When a Reversion Has Succeeded

When a reversion to normal routing, initiated via the **route trunk** command, has succeeded, the following status message appears at the primary node of a Session Maintenance trunk.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
7611 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT STATUS: trunksig: Reversion of session mtce trunk succeeded.
```

SCREEN 5-12. Reversion Succeeded Status Message

When a Reversion Has Failed

When a recently initiated reversion of a Session Maintenance trunk to normal routing has failed, a Report Alarm appears at the primary node for the trunk being reverted. This type of failure should not happen if the trunk is healthy again and has regained data transmission integrity. Data transport may be interrupted until the trunk is rerouted again (either manually or automatically).

The recommended action is to ensure that the trunk is healthy, and reinvoke the **route trunk** command to route the trunk normally. (If a Trunk-T3 is not fully functional, it cannot be brought back to normal routing with the command.)

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
** 7106 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT ALARM: trunksig: Reversion of session mtce trunk failed.
Data transport may be interrupted.
Rec act: Check trunk's integrity, and route trunk normal again.
```

SCREEN 5-13. Reversion Failure Report Alarm

When an Assisting Node Has Participated in a Reversion

A status message appears at an assisting node for each reroute path when the node is participating in a Session Maintenance trunk reversion. This message appears only if the Assisting Node Status Messages Enabled parameter has been set.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
7612 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT STATUS: trunksig: Assisting node participated in a reversion.
```

SCREEN 5-14. Assisting Node Participated in a Reversion Status Message

The following variable field is unique to Assisting Node Participation Messages:

- *MODADDR=<nnn>*

This is the address of the module that received the reversion request.

When a Trunk-PQ Downloads Prior to Reversion

This alarm applies only to the Trunk-PQ.

The maintenance of committed information rate depends on information in the trunk module; thus, it does not carry over to the reroute trunk. As long as the failed trunk does not download before it is healthy again, the maintenance of committed information rate resumes after reversion to the original path. However, if the Trunk-PQ does download, calls on CIR channel sets are dropped so that they can be reestablished with their CIR information. Since frame relay calls are PDDs, this occurs without administrator intervention.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
7615 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT ALARM: trunksig: Reversion drops calls on CIR channel sets.
Reason: CIR information lost due to download of module.
Rec act: None
```

SCREEN 5-15. When a Trunk-PQ Downloads Prior to Reversion

When the Node Drops Channel Sets

This case falls outside the categories of reroute or reversion administration. A message is generated by the node that has initiated the drop of the channel set. A trunk that has been carrying a rerouted channel set has dropped the channel set. All calls on that channel set will be dropped. The channel set is not rerouted.

```
<YY-MM-DD> <hh:mm:ss> NODE=<xxx...x>
** 7107 MODADDR=<nnn> MODTYPE=<type> EVENTID=<node.trunk>
REPORT ALARM: trunksig: Node drops rerouted channel set.
Reason: <reason>
```

SCREEN 5-16. Node Drops Channel Set Message

The reasons unique to this alarm are explained in Table 5-3.

TABLE 5-3. Reason Messages for Dropped Channel Set

Reason	Explanation
Standby Trunk Failed	The trunk that has been providing the backup service has itself failed and can no longer support any channel sets.
Standby Trunk Out of Service	The trunk that has been providing the backup service has been taken out of service and can no longer support any channel sets.
Initialization Cleanup	The node that has been providing the backup service has been forced to drop the channel set because the node on the other side of the trunk is unaware of the reroute. This may happen if a node is rebooted before its record of the reroute has been saved on disk.
Glare Resolution	This may appear if the nodes at the two ends of a trunk both simultaneously try to use the same channel set for different reroutes. One such attempt will fail, and this message will appear.
Rerouted Trunk Is Deleted	Although a rerouted trunk may be removed from service, it may not be changed or deleted without affecting rerouted calls. If an administrator attempts to delete or change such a trunk, its rerouted channel sets will be dropped with this message.

When PDDs Are Restored

Predefined destination (PDD) calls may require additional attention from administrators when a trunk carrying PDD calls fails and the trunk's active channel sets are automatically rerouted via Session Maintenance.

If the reroute and subsequent reversion of the channel sets carrying those calls is successful, no action is needed.

If channel set(s) that carry PDD calls are not successfully rerouted, PDD behavior is the same as if these PDDs were on trunks without Session Maintenance. Normal alternate routing mechanisms will attempt to re-establish the PDD calls. Typically, the calls will be re-established via call setup alternate routing. When the failed trunk that originally carried the PDD calls is healthy and the successfully rerouted channel sets are reverted to their original path, the calls that were dropped will not be included in the reversion. Those calls (the calls dropped and re-established through call setup alternate routing) will continue to follow the alternate paths provided through alternate routing.

With all trunks working, during off hours, PDD calls can be reinitialized using the **remove or restore host** command without any administration on the trunk. This action will take down all calls to the host, then allow re-establishment of the sessions on the normally engineered paths. Any PDDs rerouted to less desirable paths because of an unsuccessful reroute attempt will be restored to their normal routing.

StarKeeper II NMS Programmer's Interface

The *StarKeeper II* NMS Programmer's Interface (PI) provides a capability to activate user-programmed C language application programs and UNIX shell scripts when the *StarKeeper II* NMS controlling the network receives messages related to Session Maintenance events.

Creating applications that are activated through the Programmer's Interface facility would be useful in most networks with Session Maintenance for the following purposes:

- tracking the status of reroutes invoked by trunk failure(s)
- initiating reversion requests for a trunk that has regained data transmission integrity

The sections that follow provide examples of the Programmer's Interface command (**pi**) used to invoke a user-supplied application program and an example of a useful program that will perform administration for each purpose noted above. In addition, see *StarKeeper II* NMS documentation for more details.

Automatic Request for Reroute Status

The following screen shows the **pi** command and an example of a program that requests the status of Session Maintenance following a reroute event. The program is activated when the following messages are sent:

- 7105 — Unsuccessful Reroute
- 7106 — Reversion Failed
- 7605 — Successful Reroute

The script (program), called **smstat.sh** in the example, is invoked by default for all nodes. It can also be modified so that it is invoked for specific nodes. It must be entered by the *StarKeeper II* NMS administrator following procedures in *StarKeeper II* NMS documentation.

```
# This program monitors alarms and messages for Session Maintenance
# events and issues StarKeeper II NMS commands to gather the status of reroutes.

# Header information.
echo "$NODE_ID Alarm generated." >> $HOME/smlog
echo "Received at `date`" >> $HOME/smlog
echo "$MESSAGE" >> $HOME/smlog
echo "" >> $HOME/smlog

# Issue the StarKeeper smstat command for this primary node.
smstat -n $NODE_ID >> $HOME/smlog

# To invoke this program for unsuccessful reroutes, use the following PI command:
# pi -r smstat.sh -m 7105

# If you wish to obtain status for a failed reversion or successful reroute,
# issue the above PI command with the following alarm numbers:
# 7106 - reversion failed
# 7605 - reroute successful
```

SCREEN 5-17. Sample Script for Route Status

Performing an Automatic Reversion

The following screen shows the **pi** command and an example of a program that requests reversion of a reroute. The program, called **reversion.sh** in the example, is activated when the following message is sent:

- 7609 — Session Maintenance Trunk Has Recovered

```
# This program is initiated by the message reporting that a
# Session Maintenance trunk has returned to a healthy state.
# When this message is received, a command can be automatically issued to
# return the rerouted channel sets to their primary path.

# Header information.
echo "$NODE_ID Alarm generated." >> $HOME/smlog
echo "Received at `date`" >> $HOME/smlog
echo "$MESSAGE" >> $HOME/smlog
echo "" >> $HOME/smlog

# Issue the route trunk command for primary node and the
# now healthy trunk.
route -n $NODE_ID trunk $SLOT1 normal >> $HOME/smlog

# To invoke this program, use the following PI command.
# pi -r reversion.sh 7609
```

SCREEN 5-18. Sample Script for Automatic Reversion

Appendix. Commands Reference

This appendix contains examples of command entry, system prompts, responses, and output that can appear at the node administration console for database information commands in networks using Session Maintenance. This information is provided for reference here because it is not included in the *Node Reference*. A word of caution: using these commands for configuration at the node console is not advised; see Chapter 4, **Configuration, Simulation, and Analysis**. The examples below include user input that, in specific instances, will generate the INFO responses also shown in the same screen.

Information about service state commands related to Session Maintenance are not repeated here because they may be entered at the console *are* included in the *Node Reference*.

The following commands, as they are related to Session Maintenance, are shown and discussed below:

- **enter trunk**
- **change trunk**
- **delete trunk**
- **enter node**
- **change node**

```
CC0> enter
OBJECTS [...trunk...]: trunk
TRUNK MODULE TYPE [64, dds, e3, e3s, hs, pq, sft, swt, t1, t3, t3i, t3s]: swt
MODULE ADDRESS: 18
SESSION MAINTENANCE TRUNK [yes, no: +(no)]: yes
NODE AT REMOTE END OF TRUNK [up to 8 chars]: Redqueen
PRIMARY NODE [yodels, Redqueen: +(Redqueen)]: yodels
COMMENT [up to 60 chars double quoted]:
"SM trunk"
GROUP [up to 8 chars]: trknj
NUMBER OF ACTIVE USER CHANNELS [1-500: +(125)]: +
FRAME TERMINATION LENGTH HPQ/LPQ [16/16, 16/64, 64/64, 64/256,: +(64/256): +
LINE SPEED [2400, 4800, 9600, 19200, 48k, 56k, 64k, 128k, 192k, 256k,
320k, 348k, 448k, 448k, 512k, 576k, 640k, 704k, 768k, 832k, 896k, 960k, 1.024M,
1.088M, 1.152M, 1.216M, 1.28M, 1.344M, 1.40M, 1.472M, 1.536M,
1.544M, 2.048M: +(56k)]: +
EXPECTED BANDWIDTH FOR ACTIVE CHANNEL SETS (PERCENT) [0-100]: 50
TOTAL NUMBER OF ACTIVE CHANNEL SETS [1-4: +(1)]: +

INFO: No channel sets available for standby; total number of standby channel
sets is zero.
FAILURE DECLARATION THRESHOLD (SECONDS) [2-120: +(4)]: +
RECOVERY DECLARATION THRESHOLD (SECONDS) [30-420: +(120)]: +
ENABLE TRUNK MEASUREMENTS FOR THIS TRUNK [yes, no: +(no)]: +
CALL SCREENING PROFILE ID [up to 8 chars, none: +(none)]: callscr1

TRUNK MODULE TYPE [64, dds, e3, e3s, hs, pq, sft, swt, t1, t3, t3i, t3s]: swt
MODULE ADDRESS: 109
SESSION MAINTENANCE TRUNK [yes, no: +(no)]: yes
NODE AT REMOTE END OF TRUNK [up to 8 chars]: Redqueen

INFO: Node yodels is currently assigned as the primary node for this
node pair.
COMMENT [up to 60 chars double quoted]:
"SM trunk"
GROUP [up to 8 chars]: trknj
NUMBER OF ACTIVE USER CHANNELS [1-500: +(125)]: +
LINE SPEED [9600, 19200, 48k, 56k, 64k: +(56k)]: +
EXPECTED BANDWIDTH FOR ACTIVE CHANNEL SETS (PERCENT) [0-100]: 30
TOTAL NUMBER OF ACTIVE CHANNEL SETS [1-4: +(1)]: +
FAILURE DECLARATION THRESHOLD (SECONDS) [2-120: +(4)]: +
RECOVERY DECLARATION THRESHOLD (SECONDS) [30-420: +(120)]: +
ENABLE TRUNK MEASUREMENTS FOR THIS TRUNK [yes, no: +(no)]: +
CALL SCREENING PROFILE ID [up to 8 chars, none: +(none)]: callscr1

CC0>
```

SCREEN A-1. enter trunk Command

```

CC0> enter
OBJECTS [...trunk...]: trunk
TRUNK MODULE TYPE [64, dds, e3, e3s, hs, pq, sft, swt, t1, t3, t3i, t3s]: t3
MODULE ADDRESS: 13
DOWNLOAD SERVER [+ (controller)]: +
SOFTWARE VERSION [+(standard)]: <CR>
SESSION MAINTENANCE TRUNK [yes, no: +(no)]: yes
NODE AT REMOTE END OF TRUNK [up to 8 chars]: emerald
PRIMARY NODE [jewel, emerald: +(emerald)]: <CR>
COMMENT [up to 60 chars double quoted]:
"SMDS SM trunk"
GROUP [up to 8 chars]: trksmdsnj
NUMBER OF ACTIVE USER CHANNELS [0-132: +(32)]: 1250
EXPECTED BANDWIDTH FOR CONNECTIONLESS TRAFFIC (PERCENT) [0-100]: 40
EXPECTED BANDWIDTH FOR ACTIVE CHANNEL SETS (PERCENT) [0-60]: 40
TOTAL NUMBER OF ACTIVE CHANNEL SETS [10-64: +(10)]: 12
TOTAL NUMBER OF STANDBY CHANNEL SETS [0-52: +(0)]: 3
TRUNK WEIGHT [1-16: +(4)]: 2
DQDB BUS INDICATION [a, b: +(a)]: +
THRESHOLD PROFILE ID [1-16, or "default": +(default)]: +
TRUNK-CSU DISTANCE RANGE (FEET) [0-450, 450-900: +(0-450)]: +
ENABLE TRUNK MEASUREMENTS FOR THIS TRUNK [yes, no: +(no)]: +
CALL SCREENING PROFILE ID [up to 8 chars, none: +(none)]: <CR>

TRUNK MODULE TYPE [64, dds, e3, e3s, hs, pq, sft, swt, t1, t3, t3i, t3s]: t3
MODULE ADDRESS: 14
DOWNLOAD SERVER [+ (controller)]: +
SOFTWARE VERSION [+(standard)]: <CR>
SESSION MAINTENANCE TRUNK [yes, no: +(no)]: yes
NODE AT REMOTE END OF TRUNK [up to 8 chars]: emerald

INFO: Node emerald is currently assigned as the primary node for this node
      pair.
COMMENT [up to 60 chars double quoted]:
"SMDS SM trunk"
GROUP [up to 8 chars]: trksmdsnj
NUMBER OF ACTIVE USER CHANNELS [0-8000: +(125)]: 1875
EXPECTED BANDWIDTH FOR CONNECTIONLESS TRAFFIC (PERCENT) [0-100]: 40
EXPECTED BANDWIDTH FOR ACTIVE CHANNEL SETS (PERCENT) [0-60]: 40
TOTAL NUMBER OF ACTIVE CHANNEL SETS [15-64: +(10)]: 16
TOTAL NUMBER OF STANDBY CHANNEL SETS [0-48: +(0)]: 5
TRUNK WEIGHT [1-16: +(4)]: 2
DQDB BUS INDICATION [a, b: +(a)]: +
THRESHOLD PROFILE ID [1-16, or "default": +(default)]: +
TRUNK-CSU DISTANCE RANGE (FEET) [0-450, 450-900: +(0-450)]: +
ENABLE TRUNK MEASUREMENTS FOR THIS TRUNK [yes, no: +(no)]: +
CALL SCREENING PROFILE ID [up to 8 chars, none: +(none)]: <CR>

CC0>

```

SCREEN A-2. enter trunk Command for an SMDS Trunk

```
CC0> enter
OBJECTS [..., trunk,...]: trunk
TRUNK MODULE TYPE [64, dds, e3, e3s, hs, pq, sft, swt, t1, t3, t3i, t3s]: pq
MODULE ADDRESS: 113
DOWNLOAD SERVER [(controller)]: +
SOFTWARE VERSION [(standard)]: +
UPLOAD SERVER [(none)]: +
SESSION MAINTENANCE TRUNK [yes, no: +(no)]: yes
NODE AT REMOTE END OF TRUNK [up to 8 chars]: Redqueen
PRIMARY NODE [HEMLOCK, Redqueen: +(HEMLOCK)]: +
COMMENT [up to 60 chars double quoted]:
"Trunk to Redqueen"
GROUP [up to 8 chars]: trkgrp
TRAFFIC TYPE [cir, non-cir, both: +(non-cir)]: both
NUMBER OF ACTIVE USER CHANNELS [0-500: +(125)]: +
LINE SPEED [56k, 64k, 128k, 192k, 256k, 320k, 384k, 448k, 512k, 576k,
640k, 704k, 768k, 832k, 896k, 960k, 1.024M, 1.088M, 1.152M, 1.216M,
1.280M, 1.344M, 1.408M, 1.472M, 1.536M, 1.544M, 2.048M: +(1.544M)]: +
OPTIMIZATION [low_delay, high_throughput: +(low_delay)]: +
AGGREGATE INFORMATION RATE FOR NON-CIR CHANNELS
[percentage of line speed, 10%-100%; a value in bps, 1200-1544000: +(10%)]: +
MAXIMUM AGGREGATE CIR
[percentage of line speed, 1%-390%; a value in bps, 1200-6021600: +(90%)]: +
EXPECTED BANDWIDTH FOR ACTIVE CHANNEL SETS (PERCENT) [0-100]: 39
TOTAL NUMBER OF ACTIVE NON-CIR CHANNEL SETS [0-3: +(1)]: +
TOTAL NUMBER OF ACTIVE CIR CHANNEL SETS [1-3: +(1)]: +
TOTAL NUMBER OF STANDBY CHANNEL SETS [0-2: +(2)]: +
FAILURE DECLARATION THRESHOLD (SECONDS) [2-120: +(4)]: +
RECOVERY DECLARATION THRESHOLD (SECONDS) [30-420: +(120)]: +
CALL SCREENING PROFILE ID [up to 8 chars, none: +(none)]: +
```

SCREEN A-3. enter trunk Command for a PQ Trunk

```

CC0> change
OBJECTS [...trunk...]: trunk
MODULE ADDRESS: 18
NODE AT REMOTE END OF TRUNK [up to 8 chars: +(Redqueen)]: +

INFO: Changing the primary node will automatically change the primary node
      for all parallel trunks for this node pair.
PRIMARY NODE [yodels, Redqueen: +(yodels)]: +
COMMENT [up to 60 chars double quoted, or none:
"(SM trunk")]:
"Redqueen/yodels pair"
GROUP [up to 8 chars: +(trknj)]: +
NUMBER OF ACTIVE USER CHANNELS [0-506: +(32)]: +
LINE SPEED [9600, 19200, 48k, 56k, 64k: +(56k)]: +
EXPECTED BANDWIDTH FOR ACTIVE CHANNEL SETS (PERCENT) [0-100: +(40)]: +
TOTAL NUMBER OF ACTIVE CHANNEL SETS [4-4: +(4)]: +

INFO: No channel sets available for standby; total number of standby channel
      sets is zero.
FAILURE DECLARATION THRESHOLD (SECONDS) [2-120: +(4)]: +
RECOVERY DECLARATION THRESHOLD (SECONDS) [30-420: +(120)]: +
ENABLE TRUNK MEASUREMENTS FOR THIS TRUNK [yes, no: +(no)]: yes
CALL SCREENING PROFILE ID [up to 8 chars, none: +(callscri1)]: +

MODULE ADDRESS: 19
NODE AT REMOTE END OF TRUNK [up to 8 chars: +(Redqueen)]: +

INFO: Changing the primary node will automatically change the primary node
      for all parallel trunks for this node pair.
PRIMARY NODE [yodels, Redqueen: +(yodels)]: +
COMMENT [up to 60 chars double quoted, or none:
+("SM trunk")]:
"Redqueen/yodels pair"
GROUP [up to 8 chars: +(trknj)]: +
NUMBER OF ACTIVE USER CHANNELS [0-500: +(125)]: +
LINE SPEED [9600, 19200, 48k, 56k, 64k: +(56k)]: +
EXPECTED BANDWIDTH FOR ACTIVE CHANNEL SETS (PERCENT) [0-100: +(40)]: +
TOTAL NUMBER OF ACTIVE CHANNEL SETS [1-4: +(4)]: 1
TOTAL NUMBER OF STANDBY CHANNEL SETS [0-3: +(3)]: +
FAILURE DECLARATION THRESHOLD (SECONDS) [2-120: +(4)]: +
RECOVERY DECLARATION THRESHOLD (SECONDS) [30-420: +(120)]: +
ENABLE TRUNK MEASUREMENTS FOR THIS TRUNK [yes, no: +(no)]: yes
CALL SCREENING PROFILE ID [up to 8 chars, none: +(callscri1)]: +

CC0>
Timeout On Input

```

SCREEN A-4. change trunk Command

```
CC0> delete
OBJECTS [...trunk...]: trunk
MODULE ADDRESS: 18

WARNING: Check network Node Reroute Tables for consistency.

CC0>
```

SCREEN A-5. delete trunk Command

```
CC0> enter
OBJECTS [...node...]: node
.
.
.
.
FINE TUNE SESSION MAINTENANCE PARAMETERS [yes, no: +(no)]: yes
ASSISTING NODE STATUS MESSAGES ENABLED [yes, no: +(no)]: +
ADDITIONAL REROUTE REQUESTS (PERCENT) [0-100: +(50)]: +
REDUCED REROUTE BANDWIDTH (PERCENT) [0-90: +(20)]: +
REROUTE REQUEST WITHDRAWAL TIME (SECONDS) [5-60: +(15)]: +
.
.
.
.

CC0>
Timeout On Input
```

SCREEN A-6. enter node Command

```
CC0> change
OBJECTS [...node...]: node
.
.
.
.
FINE TUNE SESSION MAINTENANCE PARAMETERS [yes, no: +(no)]: yes
ASSISTING NODE STATUS MESSAGES ENABLED [yes, no: +(no)]: yes
ADDITIONAL REROUTE REQUESTS (PERCENT) [0-100: +(50)]: +
REDUCED REROUTE BANDWIDTH (PERCENT) [0-90: +(20)]: +
REROUTE REQUEST WITHDRAWAL TIME (SECONDS) [5-60: +(15)]:30

WARNING: Check network Node Reroute Tables for consistency.
.
.
.
.

CC0>
Timeout On Input
```

SCREEN A-7. change node Command

Index

A

Active calls, 2-3, 2-4
Active channel sets, 4-6
Active Channels/Active Channel Set field in
 Engineering Data Report, 4-17
Active User Channels, 4-5
Additional requests in Simulator, 4-15
Additional reroute requests, 2-14, 4-10, 4-11, 4-14
Administration, 1-5, 5-3
 commands, 2-4, 5-7
 features, 2-15
 steady-state, 5-3
Already rerouted status message, 5-13
Alternate paths, 2-3, 2-10
Analysis, 4-13
 and Network Builder, 2-16, 4-4
 and Simulation task, 4-14
 of held configuration data, 4-3
 of trunk failures, 2-15
Analyze: Session Maintenance Simulation task in
 Network Builder, 4-4
Assisting nodes, 2-5, 2-8, 2-10
 and messages, 5-4
 and status messages, 4-11
 list of, 4-9
 ordering of in NRT, 2-11, 2-14
 participation status message, 4-10, 4-11, 5-11, 5-16
Automatic request for reroute status sample program,
 5-19
Automatic reversion,
 sample program, 5-21
Average hops in Simulator, 4-16

B

Balancing trunks, 3-5
Bandwidth, 1-3, 2-10
 allocation of, 2-10
 and active channel sets, 4-7
 and standby channel sets, 4-8

 determining values for, 4-12
 expected, 4-7, 4-17
 insufficient, 5-10
 reduction, 4-12, 4-15
 reserved, 2-10
 sufficient, 3-7, 4-12
 total utilization of, 2-10

C

change node command, A-6
change trunk command, A-3, A-4
Channel sets, 2-3, 2-10, 5-10
 active, 2-3, 4-6, 4-12
 active and bandwidth, 4-7
 and dropped calls, 4-11
 and groups, 4-6
 and reroutes, 2-10
 and round robin routing, 4-6
 and the Simulator, 4-13
 and utilization, 4-6
 assignment of, 2-8
 dropped, 2-16, 5-8, 5-17
 needed in Simulator, 4-15
 standby, 2-3, 2-8, 4-7, 5-3
 standby and bandwidth, 4-8
 successful in Simulator, 4-15
Configuration, 1-3, 4-3
 and held data, 4-3, 4-5
 and Network Builder, 2-15
 and network topology, 3-6
 and node commands, 4-3
 data and the Simulator, 4-13
 databases, 4-3
 databases and downloading, 2-15
 entering data, 4-3
 from node console, 4-3
 of failure threshold, 4-7
 of node parameters, 4-10
 of standby channel sets, 4-7

- of trunks, 2-3
- philosophy of, 4-3
- process for, 4-3

Configure: Node task in Network Builder, 4-3

Configure: NRT task in Network Builder, 4-4

Configure: Trunk task in Network Builder, 4-3

Control Computer, 1-3, 5-14

D

Database(s),

- and trunks, 2-7
- population of, 4-4

Defaults,

- and node tuning parameters, 4-10
- for configuration, 4-13
- for recovery threshold, 4-7
- for trunk parameters, 4-5

delete trunk command, A-5

Detailed Report, 4-14, 4-16

Device timers,

- and failure declaration, 4-7

E

Editing,

- lists in NRT, 4-9

End node specification parameter, 4-5

Engineering Data Report, 4-6, 4-9, 4-14, 4-16

enter node command, A-6

enter trunk command, A-1, A-2

Events,

- in a simulation, 4-13

Expected bandwidth, 4-7, 4-17

F

Failed trunk recovered message, 5-14

Failed trunks in Simulator, 4-15

Failure declaration threshold, 4-7

Failure detection, 2-14, 5-6

Failure mode, 4-14, 4-15

Figure of merit,

- in NRT, 2-13

G

Glare resolution, 5-18

Groups, 4-6

H

Healthy trunk, 5-14

Held configuration data, 4-3, 4-13

Hop(s), 2-9, 2-10

- and neighbor nodes, 2-11
- definition of, 2-8
- excessive, 5-10
- limits of, 3-4
- to neighbor nodes, 2-12

I

Initialization cleanup, 5-18

Internodal connectivity, 1-3

Interpreting the Simulator Output, 4-14

K

Keep-alive messages, 2-3, 2-14, 4-7, 5-5

M

Manual reroute requested status message, 5-7

Mesh-grid networks, 3-6

Module addresses, in NRT 4-9

Multi-hop alternate paths, 2-3

N

Negotiation,

- for reroutes, 2-14

Network Builder, 1-3, 1-5, 2-15

- and administration, 2-3, 5-3
- and configuration, 2-3, 4-3
- and node tuning parameters, 4-10
- and NRT generation, 2-13, 2-15, 3-7, 4-9
- and trunks, 2-7

Configure: NRT task, 4-9

functionality for configuration, 4-3

Network planning,

- design, 3-3

Network Summary Report, 4-13, 4-14

Node commands, A-1

Node Reroute Table (NRT), 2-3, 2-7, 2-11, 2-12, 5-3, 5-10

- and configuration, 4-3, 4-9
- and neighbor nodes, 2-3
- editing of, 2-13, 4-4
- generation of, 4-4, 4-9, 4-13
- regeneration of, 2-16
- viewing of, 4-4

Node tuning parameters, 2-3, 2-4, 4-4, 4-10

Node(s),

- and connectivity, 4-18
- and dropped channel sets, 5-17
- and reliability options, 2-15
- assisting, 2-8, 2-10, 4-11, 5-4
- busy, 4-10
- failures of, 2-15
- hub, 3-5
- names for, 2-7, 2-9
- neighbor, 2-11, 2-12, 4-9
- primary, 2-7, 3-5, 4-18, 5-4
- reroute, 2-7
- secondary, 2-7, 2-8, 5-4
- unstable, 4-10
- without Session Maintenance, 2-16, 3-3

P

Parallel paths, 2-5

Path Definition field in Detailed Report, 4-16

Pending configuration data, 4-3, 4-13

Performance monitoring, 5-3

Physical trunks,

- and channel sets, 2-3

Predefined destinations (PDDs), 5-18

Primary node, 2-7, 3-5, 4-8, 4-18

- and increasing reroute requests, 4-12
- and messages, 5-4
- and NRT, 2-11
- selection of, 4-8

Programmer Interface, 5-4, 5-14

R

Real-time testing routines, 2-3, 2-4

Reason messages,

- for dropped channel set, 5-18
- for trunk failures, 5-9

Recovery declaration threshold, 4-7

Reduced reroute bandwidth, 4-12

remove trunk command, 5-10

Reporting,

- status, 2-15

Request(s),

- dead in Simulator, 4-16
- field in Detailed Report, 4-16
- rejected in Simulator, 4-15
- sent in Simulator, 4-15

Requirements for Session Maintenance, 1-5

Reroute,

- bandwidth reduction, 4-10, 4-12, 4-14
- Node Connectivity field in Engineering Data Report, 4-18
- nodes, 2-7
- request withdrawal time, 4-10, 4-11
- unnecessary status message, 5-13
- unsuccessful report alarm, 5-8

Rerouted trunk deleted, 5-18

Reroutes,

- and additional requests, 2-4, 2-8
- and administration, 5-5
- and trunk balancing, 3-5
- automatically invoked, 2-3, 5-3, 5-5
- evaluating, 2-15
- manually invoked, 5-3, 5-7
- negotiation of, 2-5, 2-14, 4-6
- process of, 2-14
- requests, 2-3, 2-7, 2-8, 4-11
- selecting trunks for, 2-11
- successful, 5-7
- unsuccessful, 5-8
- viewing success of, 4-13

restore host command, 5-18

Result field in Detailed Report, 4-16

Reversion, 2-14, 5-3, 5-14

- failed message, 5-16
- requested message, 5-15
- sample program for automatic, 5-21

succeeded message, 5-15
route trunk alternate command, 5-7
route trunk normal command, 5-8, 5-10

S

Secondary nodes, 2-7, 2-8
and messages, 5-4
and NRT, 2-11
participation status message, 5-11

Session Maintenance, 1-3

advantages of, 1-3
and active calls, 2-3
and diagnostics, 5-7
and fault tolerance, 1-4
and feature packages, 1-5
and high availability, 1-4
and network services, 2-15
and node capacity, 1-3
and node types, 2-7
and reliability, 1-3
capabilities, 2-15
components, 1-5
concepts, 2-6
Enabled/Disabled parameter, 4-5
events, 2-4
feature description, 2-3
functional description, 2-3
performance of, 3-6
placing into service, 4-4
requirements, 1-5
terminology, 2-6
trunk parameters, 4-6

Shelf,

administration, 3-6
failure, 2-15, 5-5

Simulator, 1-3, 2-15, 4-13

and bottlenecks, 3-6
and evaluating reroute requests, 4-12
and held configuration data, 4-3
and trunk balancing, 3-5
usage, 4-13, 4-14

smstat command, 5-7

smverify command, 2-12

Software, 1-5

Standby channel sets, 4-7

Standby trunk, 5-18

StarKeeper II NMS, 1-3

and messages, 2-14
auto-clearing, 5-4
components, 1-5
Core System, 1-5, 2-15, 3-7
databases,
and Simulator, 4-13
Graphics Workstation, 1-5, 3-7
Programmers Interface, 2-15, 5-19
reporting to, 2-8
scope of control, 3-3, 3-7
suite of, 3-7
Task Manager, 1-5

Status messages, 2-8, 5-3

and initial testing, 4-11
and primary nodes, 4-11
and secondary nodes, 4-11
and tracking reroutes, 4-11
for Session Maintenance, 5-4
from assisting nodes, 4-11

Success percentage in Simulator, 4-16

Successful reroute report alarm, 5-7

Successful reroutes,

and additional requests, 4-11

Switch module, 1-3, 2-4, 2-15, 5-13

Switching back, 2-14, 2-16

T

Threshold,

for trunk failure detection, 2-3

Topology, 2-10, 2-15, 2-16, 3-3

and hierarchical networks, 3-3
and mesh-grid networks, 3-3, 3-6
and primary node, 4-8
and ring networks, 3-3, 3-4
and trunk configurations, 3-6

Total Standby Bandwidth field in Engineering Data

Report, 4-17

Trunk failure,

and detection threshold, 2-3
report alarm, 5-6

-
- Trunk parameters, 4-5
 - Trunk-PQ, 2-4, 4-6, 4-8, 4-17
 - download prior to reversion, 5-17
 - Trunks,
 - and balancing on nodes, 3-5
 - and configuration data, 2-13
 - and connectivity, 3-6
 - and failure detection, 2-4, 2-7
 - and hub nodes, 3-5
 - and interworking, 1-3
 - and module addresses, 2-12
 - and module failure, 5-5
 - and multiple failures, 2-16
 - and network database, 2-7
 - and recovery detection, 2-7
 - and Session Maintenance, 2-9, 3-3
 - and shelf failure, 2-15
 - and unused bandwidth, 2-3
 - as alternate routes, 1-3
 - configuration of, 4-4
 - configuring existing, 4-4
 - fluctuating, 4-7
 - healthy, 2-4, 2-16
 - noisy, 4-7
 - number of, 3-6
 - parallel, 2-3, 2-5, 3-3, 3-4, 4-9, 4-12
 - performance of, 4-6
 - physical, 2-9
 - resources, 3-3
 - selection of, 2-11
 - sharing of,
 - by traffic, 1-3
 - spare, 1-3
 - speeds, 3-7, 4-10
 - standby capacity, 1-3
 - supported by Session Maintenance, 1-3
 - types of, 2-9, 3-6, 4-10
 - utilization of, 3-6, 4-5, 4-7, 4-8
- U**
- User channels, 4-5
- V**
- View button in Simulator, 4-14