UNISYS

DCP Series
TCP–IP Stack
Configuration and Operations Guide

2R2B December 1993

Printed in U S America
Priced Item 7831 5546–110

UNISYS

Product Information Announcement

o New Release o Revision ● Update o New Mail Code

Title:

DCP Series TCP–IP Stack Configuration and Operations Guide Level 2R2B

This Product Information Announcement announces the release and availability of the *DCP Series TCP–IP Stack Configuration and Operations Guide, Level 2R2B (7831 5546–110).* This information provides update pages for the 2R2 release, 7831 5546–100.

The *Unisys DCP Series* TCP–IP Stack program product supports TCP/IP communications across various subnetworks for Telcon software on a Unisys Distributed Communications Processor (DCP).

The pages in this update replace similarly numbered pages in 7831 5546–100.

TCP–IP Stack Level 2R2B includes the following new features:

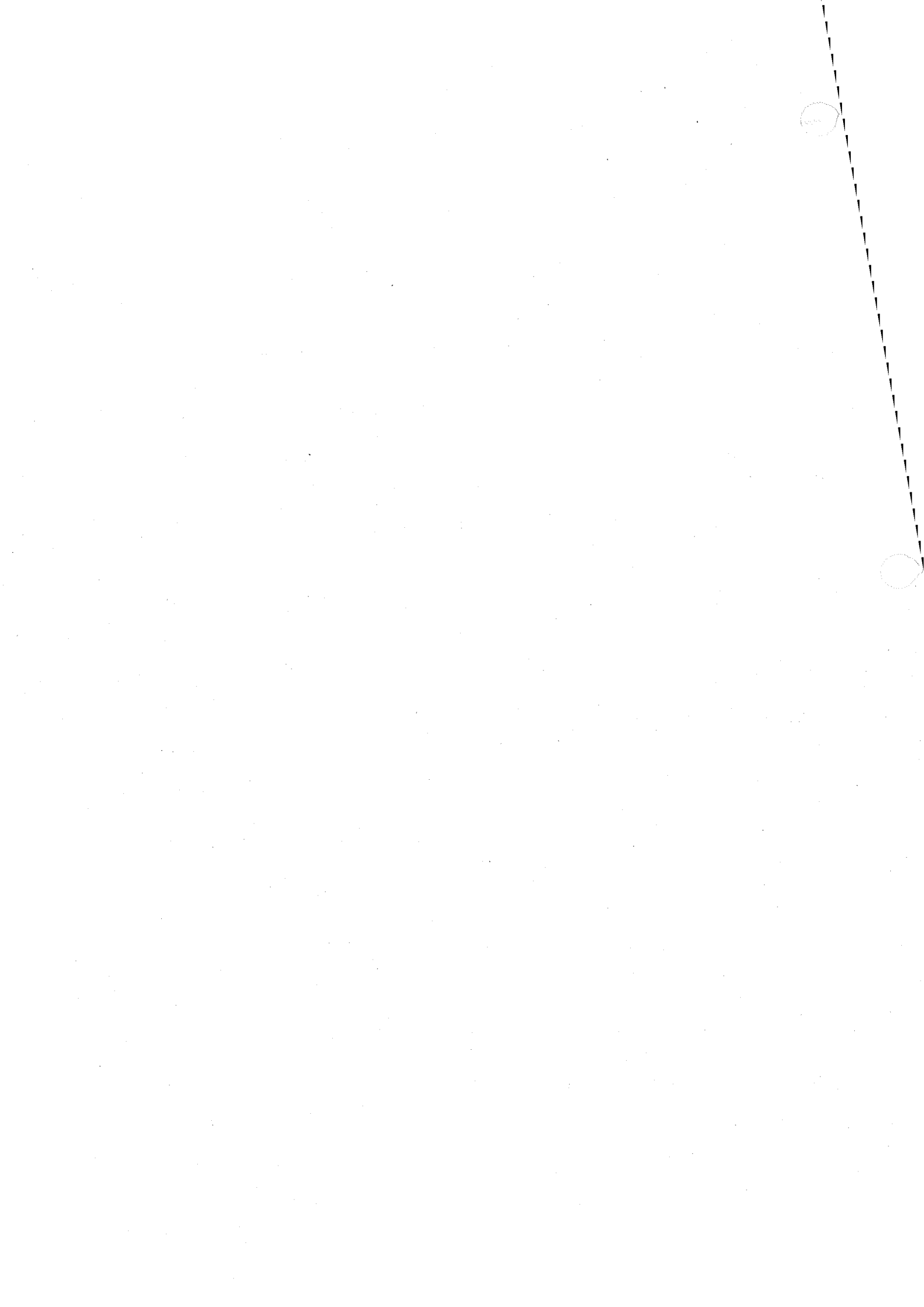
- Supports the 802.5 Token Ring intelligent line module (ILM)
- Supports TN3270 terminal emulators
- Supports DCA session establishment over a TCP/IP network

Note: *The official term for "OS 1100" has been changed to "OS 2200." Books in this library have not yet changed all references of OS 1100 to OS 2200 at this printing.*

To order additional copies of this document:

- United States customers, call Unisys Direct at 1-800-448-1424.
- All other customers, contact your Unisys Sales Office.
- Unisys personnel, use the Electronic Literature Ordering (ELO) system.

Announcement only:	Announcement and attachments:		
MU59, MBZ, MHA, MMZ, MU1Y, MX3, MX6, MX8, MY3, MY5, MY6, MY7, MU1S	AF01	System:	DCP Series
		Release:	Level 2R2B
		Part number:	7831 5546–110



UNISYS

Product Information Announcement

o New Release ● Revision o Update o New Mail Code

Title:

DCP Series TCP-IP Stack Configuration and Operations Guide Level 2R2

This Product Information Announcement announces the release and availability of the *DCP Series TCP-IP Stack Configuration and Operations Guide, Level 2R2 (7831 5546–100).*

The *Unisys DCP Series* TCP-IP Stack program product supports TCP/IP communications across various subnetworks for Telcon software on a Unisys Distributed Communications Processor (DCP).

TCP-IP Stack works with the following DCP Series software, which provides the necessary hardware connectivity:

- LAN Platform for communications over 802.3 and FDDI local area networks (LANs)
- X.25 Packet Switched Communications Software (PSCS) for communications over X.25 packet switched networks, including the Defense Data Network (DDN), public data networks (PDNs), and private packet networks.

TCP-IP Stack supports IP communications across the following networks:

- 802.2, 802.3, and FDDI local area networks (LANs)
- X.25 packet switched networks, including the Defense Data Network (DDN), public data networks (PDNs), and private packet networks
- Unisys DNS networks
- OS 1100-to-DCP host channels

In addition, TCP-IP Stack enables communication between DCA and TCP/IP networks.

TCP-IP Stack Level 2R2 includes the following new features:

- Supports new 802.3 and FDDI ILMs and the 802.3 LM.
- Configuration has been simplified by eliminating the SAP and DLCUNIT configuration statements, as well as several parameters, including DELAY, TIMOUT, TCPTIME, THRUPUT, RELIABLE, and others.
- The terminal type Unisys–TD830–ASCII is now supported.
- Traced messages can be saved continuously using the new circular save file capability.
- Trace parameters can be modified without stopping the trace.

continued

Announcement only:
MU59, MAC, MBZ, MHA, MU1Y, MX3, MX5, MX6, MX8, MY1, MY3, MY5, MY6, MY7, MUS1, MUR3

Announcement and attachments:
AF01

System: DCP Series
Release: Level 2R2
Part number: 7831 5546–100

DCP Series TCP-IP Stack Configuration and Operations Guide Level 2R2

This guide provides the information you need to configure and operate OSITS, including:

- Descriptions of TCP-IP Stack and its features
- Task-oriented configuration information, which provides step-by-step instructions for configuring TCP-IP Stack
- Descriptions of configuration statements
- Configuration examples
- Information for using the TCP-IP Stack NMS commands

This guide is for those persons responsible for configuring a DCP/Telcon system with TCP-IP Stack installed. This guide is a revision and totally replaces earlier TCP-IP Stack documentation.

To order additional copies of this document:

- United States customers, call Unisys Direct at 1-800-448-1424.
- All other customers, contact your Unisys Sales Office.
- Unisys personnel, use the Electronic Literature Ordering (ELO) system.

UNISYS

DCP Series

TCP-IP Stack

Configuration and Operations Guide

Release Level 2R2

March 1993

Printed in U S America

Priced Item

7831 5546–100

Correspondence regarding this publication should be forwarded to Unisys Corporation by addressing remarks to Communication Systems Product Information, Salt Lake City Publications, MS B2C07, 322 North 2200 West, Salt Lake City, UT 84116-2979, U.S.A.

Page Status

Page	Issue
iii through vi	–110
vii	–100
vii through ix	–110
x	–100
xi through xiv	–110
xv	–100
xvi	Blank
xvii	–100
xviii	Blank
1–1	–100
1–2	–110
1–3 through 1–4	–100
1–5	–110
1–6 through 1–7	–100
1–8 through 1–9	–110
1–10	Blank
2–1 through 2–2	–100
2–3 through 2–4	–110
2–5 through 2–7	–100
2–8 through 2–9	–110
2–10	–100
2–11	–110
2–12 through 2–13	–100
2–14	–110
2–15 through 2–23	–100
2–24	–110
2–25 through 2–26	–100
2–27 through 2–28	–110
2–29 through 2–37	–100
2–38 through 2–39	–110
2–40 through 2–49	-100
2–50 through 2–53	–110
2–54	Blank
3–1	–110
3–2	-100
3–3 through 3–7	–110
3-8	–100
3–9	–110
3–10 through 3–12	–100
3–13	-110
3–14	–100
3–15 through 3–17	–110

Page Status

Page	Issue
3-18	-100
3–19	-110
3–20 through 3–24	-100
3–25 through 3–26	–110
4–1	–100
4–2 through 4–3	–110
4–4 through 4–8	-100
4–9 through 4–10	-110
4–11 through 4–14	-100
4–15 through 4–16	-110
4–17 through 4–24	–100
4–25	-110
4–26 through 4–30	-100
4–31 through 4–42	-110
A–1 through A–6	–100
A–7	-110
A–8 through A–24	-100
A–25 through A–26	–110
A–27 through A–28	–100
A-29 through A-30	–110
B–1 through B–23	–100
B–24	Blank
Glossary–1 through 16	–100
Bibliography–1	–110
Bibliography–2	Blank
Parameters Index–1 through 3	–110
Parameters Index–4	Blank
General Index–1 through 8	–110

Page Status

Page	Issue
iii	–100
iv	Blank
v through xv	–100
xvi	Blank
xvii	–100
xviii	Blank
1–1 through 1–8	–100
2–1 through 2–51	–100
2–52	Blank
3–1 through 3–23	–100
4–1 through 4–31	–100
A–1 through A–34	–100
B–1 through B–23	–100
B–24	Blank
Glossary–1 through Glossary–15	–100
Bibliography–1	–100
Bibliography–2	Blank
General Index–1 through General Index–8	–100
Parameters Index–1 through Parameters Index–3	–100
Parameters Index–4	Blank

About This Guide

Purpose

This guide describes how to configure and operate the TCP-IP Stack program product on Distributed Communications Processors (DCPs).

Scope

This guide provides the following:

- A description of level 2R2 and 2R2B enhancements
- A description of product capabilities, consisting of both enhancements and capabilities introduced in earlier releases
- A list of product limitations
- Procedures that explain how to configure various capabilities
- A description of the configuration statements used to configure TCP-IP Stack
- Procedures on how to use product-specific Network Management System (NMS) commands
- Configuration examples that illustrate how typical networks are configured
- Conceptual information about TCP/IP

Audience

There are two audiences for this guide:

- System and network administrators who configure TCP-IP Stack software
- Operators who issue NMS commands

Prerequisites

To use this guide to configure TCP-IP Stack software, you should have Telcon and program product configuration experience or have attended Unisys classes on these subjects. You should also be familiar with the TCP/IP protocols. To use the NMS commands described in this guide, you should be an experienced DCP operator.

How to Use This Guide

Everyone involved with TCP-IP Stack configuration and operations should read Section 1, which provides an overview of the product's capabilities.

If you are responsible for configuration, read the parts of Section 2 that describe procedures you need to implement TCP-IP Stack program products. Use Section 3 for reference when you need a detailed description of a configuration statement. Refer to Appendix A for configuration examples, and Appendix B if you require conceptual information.

If you are responsible for operations, read Section 4 for NMS command formats information. All NMS commands begin on even numbered pages.

Organization

This guide is organized as follows:

Section 1. Introduction

This section provides an introduction to TCP-IP Stack software. It outlines the program product's enhancements for level 2R2 and 2R2B and describes its overall capabilities.

Section 2. Configuring Telcon for the TCP-IP Stack

This section describes how to configure Telcon to run the TCP-IP Stack. Configuration tasks described in this section are complemented by the configuration examples presented in Appendix A.

Section 3. TCP-IP Stack Configuration Statements

This section describes the configuration statements that are used to configure TCP-IP Stack capabilities.

Section 4. Controlling TCP-IP Stack Operations

This section describes NMS commands that are used to control and monitor TCP-IP Stack operation.

Appendix A. Configuration Examples

This appendix presents a variety of TCP-IP Stack configuration examples.

Appendix B. TCP-IP Stack Configuration Concepts

This appendix provides TCP-IP Stack conceptual information that may help you make knowledgeable configuration decisions.

Related Product Information

In addition to this guide, you may need the following Unisys documents to implement TCP-IP Stack software.

DCP Series Telcon Configuration Guide **(7831 5678)**

This guide tells how to configure Telcon software for a data communications network. It also tells you how to reconfigure these software products as your network evolves.

DCP Series Telcon Configuration Reference Manual **(7831 5686)**

This manual provides reference material for configuring data communications networks with Telcon software.

DCP Series Telcon Installation Guide **(7831 5645)**

This guide tells you how to generate, install, and verify Telcon and program products software on an OS 1100 host and its DCPs. Generating and installing involves copying the Telcon software components and related software products from release tapes to mass storage and preparing the software for use with your communications network.

DCP Series Telcon Software Operations Guide
(7831 5785)

This guide is for terminal operators. It explains how to perform daily operational tasks on a terminal within a Telcon network.

DCP Series LAN Platform Configuration and Operations Guide
(7831 5512)

This guide describes how to configure LAN Platform software on a DCP. It includes an overview of the product, hardware and software compatibility, descriptions of the required configuration statements, and examples of typical configurations.

DCP Series X.25 Packet Switched Communications Software (PSCS) Configuration and Operations Guide **(7831 5470)**

This guide describes how to configure X.25 PSCS software on a DCP. It includes an introduction to the product, a task-oriented configuration section, descriptions of the required configuration statements, examples of typical configurations, and information about the packet-switched networks X.25 PSCS supports.

This guide also covers operations, with descriptions of the NMS commands and messages.

DCP Series TCP–IP Stack TELNET User Guide **(7831 5553)**

This guide shows terminal users how to use TELNET to access applications running on hosts connected to TCP/IP networks. To support this activity, this guide also describes how to manage the TELNET environment and how to end communications with hosts.

DCP Series Telcon Message Manual **(7436 0728)**

This manual is a compilation and explanation of the various messages Telcon displays on your screen.

Notation Conventions

The following notation conventions are used in this guide:

Notation	Convention	Example
NMS commands	SMALL CAPS	DISPLAY command
Optional information	[ ]	[DCP=*name*|*netadd*] [NAME1=*name*]
Parameters	UPPERCASE Monofont	OPTIONS=RPOA parameter
Single choice from list	{ }	TYPE={HIMMIL}
Statements	SMALL CAPS	DTETYPE statement
User Entry	***Bold Italic***	***NO/YES***
Variables	*italics*	[DCP=*name*|*netadd*]

Note: *Uppercase letters identify information you must spell exactly as it appears. Commands (such as DISPLAY and SNAP) and parameters (such as DCP, LINE, and TERM) are examples.*

Contents

Section 2. Configuring Telcon for TCP-IP Stack

Section 3. TCP-IP Stack Configuration Statements

Section 4. Controlling TCP-IP Stack Operation

Appendix A. Configuration Examples

Appendix B. TCP-IP Stack Configuration Concepts

Figures

Tables

Section 1
Introduction

The TCP-IP Stack program product supports TCP/IP communications across various subnetworks. This section introduces the TCP-IP Stack program product. It provides the following:

- A description of enhancements for TCP-IP Stack level 2R2
- An overview of the product capabilities
- A list of TCP-IP Stack restrictions

1.1. Enhancements for Level 2R2

This subsection describes the enhancements TCP-IP Stack level 2R2 provides. These include the following:

Component	Feature
IP	Support new intelligent line modules (ILMs)
	Eliminate SAP and DLCUNIT statements
	Provide interface information with message trace
	Enhance bi-directional message trace capability
	Implement LENGTH parameter to shorten traced messages
	Allow trace parameters to be modified while the trace is operational
	Provide local addresses with IP status information
	Support Record Route option
NMS	Add circular save file capability to message trace command (REUSE)
	UPDATE display of SUBNET and IPADR information for LIST commands

Component	Feature
ETNTNET	Supports the Terminal Type Option, Terminal Type Subnegotiation, and the End-of-Record (EOR) option.
	Support Terminal Type UNISYS–TD830–ASCII
PING	Send ICMP Echo requests

1.1.1. Support New ILMs

Internet Protocol (IP) uses configured network and link layer service providers to send datagrams to their destination. The network and link layer service providers can be DCP X.25 Packet-Switched Communication Services (PSCS), Dynamic Network Services (DNS, Telcon's Distributed Communications Architecture backbone), DCP LAN Platform, or a host channel. Support of the new LAN ILMs is added with this release, and coexists with support of the existing DCP LAN Platform.

The following is a description of the ILMs :

LAN Module	Capability
802.3 ILM	This ILM is a follow-on product for the 802.3 line module (LM). Both the 802.3 ILM and the 802.3 LM are supported beginning with release 2R2.
FDDI ILM	This ILM supports the Fiber Distributed Data Interface (FDDI) physical interface standards.

1.1.2. Eliminate SAP and DLCUNIT Statements

You no longer need to use the SAP and DLCUNIT statements when configuring a subnetwork. Without those statements, TCP-IP dynamically attaches to the LAN line module or ILM. Eliminating the need to configure the SAP and DLCUNIT statements is expected to improve the ease with which TCP-IP can be configured. When subnetworks are configured using the SAP and DLCUNIT statements, TCP-IP functions as before. IP generates a CENLOG indicating that the use of SAP and DLCUNIT statements is obsolete.

1.1.3. Provide Interface Information with Message Trace

When tracing messages at the Network Service Provider (NSP) interface, IP specifies the associated interface with the traced message. The information is used when decoding the traced messages.

1.1.4. Enhance Bi-directional Message Trace Capability

When you are conducting a message trace, the SRC and DEST parameters are interpreted as exchangeable when the parameter DIR is set to BOTH. The former convention of tracing messages in both directions when SRC and DEST are identical is retained.

1.1.5. Implement LENGTH Parameter to Shorten Traced Messages

The LENGTH parameter specifies the maximum number of bytes copied for each message. When the message is displayed, the string "MESSAGE DATA WAS TRUNCATED" is appended to the end of the display if the traced message was truncated.

1.1.6. Modify Trace Parameters Without Stopping the Trace

You can enter the command TCP SNAP even for an active trace. The FILE and REUSE parameters cannot be modified for an active trace.

You can activate traces from only one NMS console at a time. If a second NMS console attempts to activate traces when traces are already active, the command is rejected with the message "TRACE ALREADY ACTIVE".

1.1.7. Provide Local Addresses With IP Status Information

The NMS command DISPLAY=IP now provides network number, subnetwork mask, local addresses, and a routing enabled indicator. This enhancement displays only one local address for each physical interface.

This feature does not indicate whether this local address is owned by a trunk statement, is associated with a DNS address in this Telcon node, or is associated with a DNS address in another Telcon node.

1.1.8. Support Record Route Option

IP fully supports the record route option. Starting with release 2R2, IP allows higher layers to request the record route option. IP formerly handled the record route option, updating it when routing datagrams contained the option, and passing the option data to the upper layer program when receiving datagrams containing the option.

1.1.9. Add Circular Save File Capability to Message Trace Command (REUSE)

This feature allows traced messages to be saved continuously, reusing the message trace save files when the last one is full.

The parameter REUSE specifies the number of save files to use while saving traced messages. The file names are generated by truncating the specified name to no more than six characters, and appending a two-digit sequence number to the end of the name, starting with 00 up to the maximum number specified by the REUSE parameter (the parameter value minus one).

When the current save file is full, TCP/IP closes it, and opens the next one. When TCP-IP traces messages without reusing the trace files, it terminates message tracing either when the end of the file is reached, or when the command TCP SNOF=IP is received.

1.1.10. Update Display of SUBNET and IPADR Information for LIST commands

The NMS LIST command displays all fields of the SUBNET and IPADR configuration statements.

1.1.11. Support Terminal Type UNISYS–TD830–ASCII

Server TELNET always requests the terminal type. If server TELNET receives the terminal type UNISYS–TD830 or UNISYS–TD830–ASCII, server TELNET negotiates the end-of-record (EOR) option.

1.1.12. Implement EOR

Implementation of the EOR option means that if the End-of-Record option is set for input, then any EOR sequence (LF,CR/LF,LF/CR,ETC) received, except the TELNET EOR, are ignored. If End-of-Record is set for output, every INT-1 EOR HIC is converted to a TELNET EOR.

1.2. Obsolete Configuration Parameters

In addition to the configuration changes listed in the previous section, the parameters listed in the following table are no longer used. If used with this release of the TCP-IP Stack program product, they may not produce error messages. However, their use may produce error messages in later releases of TCP-IP Stack.

Table 1–1. Obsolete TCP-IP Stack Configuration Parameters

Statement	Parameters
EU	TCPPORT
	DELAY
	THRUPUT
	RELIABLE
	COMPAR
	HREST
	TCC
	PRECEDNS

1.3. TCP-IP Stack Capabilities Overview

TCP-IP also provides additional capabilities. This subsection describes them.

1.3.1. Supported Network Connections

The TCP-IP Stack program product supports IP communications across the following:

- 802.3 and FDDI local area networks (LANs)
- X.25 packet switched networks, including the Defense Data Network (DDN), public data networks (PDNs), and private packet networks

- Unisys DNS networks
- OS 1100-to-DCP host channels

In addition, TCP-IP Stack enables communication between DCA and TCP/IP networks. Figure 1–1 depicts these connections.

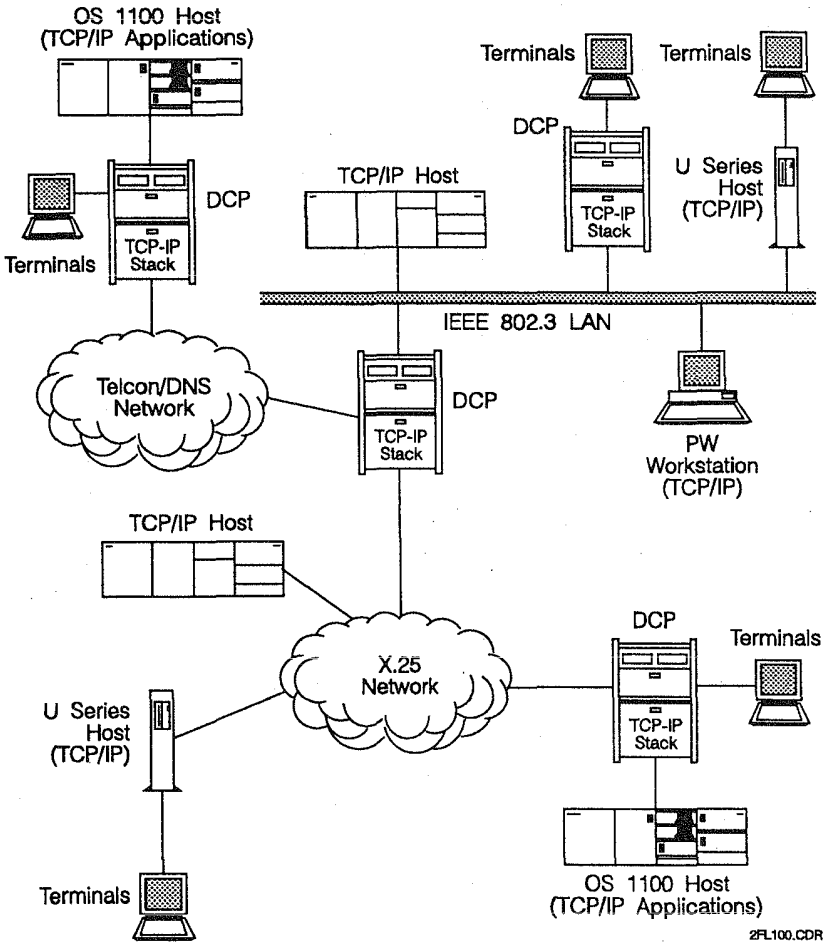


Figure 1–1. TCP-IP Stack Communications

1.3.2. Protocols Implemented

The TCP-IP Stack implements the following protocols:

Protocol	Defined as:	Description
TELNET terminal protocols	MIL-STD 1782 and RFC 854	This protocol is used in the DDN environment. Capabilities include User TELNET, which provides terminals in the DCA network with access to foreign host applications in a TCP/IP network environment, and Server TELNET, which provides terminals in a TCP/IP network environment with access to OS 1100 host applications.
Transmission control protocol (TCP)	MIL-STD 1778 and RFCs 793 and 1122	This protocol provides connections with error detection, positive acknowledgement with retransmission, sequence numbering, multiplexing, and flow control.
Internet protocol (IP)	MIL-STD 1777 and RFCs 791 and 1122	This protocol performs datagram assembly/disassembly, datagram routing, and internet addressing functions. The IP module also provides an interface to lower layer protocols implemented by other program products such as X.25 PSCS and LAN Platform.
Internet control message protocol (ICMP)	RFCs 792 and 1122	ICMP is a sublayer protocol within IP. This implementation generates error messages to be transported to other hosts in response to error conditions.
Address resolution protocol (ARP)	RFCs 826 and 1122	ARP works in combination with IP to route data to LAN-connected hosts. ARP maps internet addresses used by IP to physical addresses used by the LAN subnetwork protocol.
Routing information protocol (RIP)	RFC 1058.	This is one of the protocols used between routers to update their IP routing tables.
User datagram protocol (UDP)	RFCs 768 and 1122	This implementation provides a connectionless datagram transport service to upper layer protocols in the process/application layer. It does not guarantee datagram delivery. TCP-IP Stack uses UDP exclusively to support RIP. UDP provides a data checksum service for data integrity error detection. UDP follows the same port addressing and multiplexing rules as TCP.
Subnetwork access protocol (SNAP)	RFC 1042	This protocol acts as an intermediate protocol between internetwork and data link protocols.

Protocol	Defined as:	Description
Routing information protocol (RIP)	RFC 1058.	This is one of the protocols used between routers to update their IP routing tables.
User datagram protocol (UDP)	RFCs 768 and 1122	This implementation provides a connectionless datagram transport service to upper layer protocols in the process/application layer. It does not guarantee datagram delivery. TCP-IP Stack uses UDP exclusively to support RIP. UDP provides a data checksum service for data integrity error detection. UDP follows the same port addressing and multiplexing rules as TCP.
Domain Name System protocol	RFCs 1034 and 1035	This implementation provides a domain name resolver, allowing you to specify a host by name rather than by Internet address.
Subnetwork access protocol (SNAP)	RFC 1042	This protocol acts as an intermediate protocol between internetwork and data link protocols.

1.3.3. Access to DDN Applications

TCP-IP Stack provides two methods to access the OS 1100 DDN 1100 applications. If host-based TCP/IP is configured on the OS 1100 host, the two products communicate directly, using TCP/IP protocols. Otherwise, host services (a session relay module in the TCP-IP Stack) contains a module that provides a bridge between the DCA protocol stack and the DDN 1100 applications.

Protocol	Description
File transfer protocol (FTP) service	FTP is a file-handling service conforming to MIL-STD 1780. With terminal commands, subscribers can use FTP to store, retrieve, transfer, or delete files on the local host or on a remote host.
Simple mail transfer protocol (SMTP) service	SMTP is an electronic mail handling service conforming to MIL-STD 1781. With terminal commands, subscribers can use SMTP to create, send, and receive mail messages in an environment that includes users on the local host or on remote hosts in the network.
COBOL program interface	This interface for user-written COBOL programs provides program-call access to the FTP and SMTP services. It also provides program-to-program communications with peer processes on the local host or remote hosts in the network.

1.4. TCP-IP Stack Restrictions

1.4.1. Hardware Related

TCP-IP 2R2 does not perform to its fullest ability on DCPs with less than two megabytes of RAM.

1.4.2. Interoperability

No restrictions.

1.4.3. Compatibility

TCP-IP 2R2 is compatible with all software and hardware included in CD5R3 and Open Systems Products Release 4.

TCP-IP Stack configurations for logical link control (LLC) Type 1 and 802.3 media access control (MAC) LAN attachments on the same LAN ILM are mutually exclusive. All peer LAN stations must use the same link layer protocol for TCP/IP communications.

1.5. Enhancements for Level 2R2B

Level 2R2B includes the following enhancements:

- DCA session establishment over TCP/IP

  A new component, TCPTS, implements RFC 1006, allowing you to establish DCA sessions between DCA host systems (such as an OS2200 host or U Series hosts running Information Services) or between terminals and DCA host systems over a TCP/IP network.

- Token ring (802.5) LAN support

  This enhancement enables TCP-IP Stack to operate over an 802.5 LAN, in addition to the existing capability to operate over 802.2 and 802.3 LANs.

- TN3270 protocol implementation

  This enhancement gives you access to applications on OS 2200 or SNA host systems using personal computers attached to a TCP/IP network and running a TN3270 emulator.

- DCP628 support

  TCP-IP Stack can run with both single and dual bus line modules on the DCP628.

Section 2
Configuring Telcon for TCP-IP Stack

This section describes how to configure Telcon to run TCP-IP Stack program product software. This section does not describe how to configure a complete Telcon system. Because Telcon configuration statements apply to a wide range of communications purposes and configurations, this section describes only how to configure the following communications capabilities provided by the TCP-IP Stack software.

- The various LAN attachments available to TCP/IP on the DCP

  This subsection describes TCP-IP Stack configuration requirements for a typical attachment to a LAN LM or an ILM-40 LAN line module (models EN or FD).

- Attachment to an X.25 public data network (PDN) and Defense Data Network (DDN) X.25 subnetwork

  This subsection describes the X.25 Packet-Switched Communications Software (PSCS) program product that works with TCP-IP Stack to support TCP/IP communications over any subnetwork based on CCITT X.25 protocols. You can configure X.25 PSCS to provide proper communications services for either DDN (Internet) subnetworks or for a wide range of X.25 PDNs.

- Configuring a host channel as a TCP/IP subnetwork

  This subsection describes how you can configure TCP-IP Stack to use TCP/IP communications across a host channel between a DCP and an OS 1100 host. TCP-IP Stack interfaces with a host channel by configuring it as a TCP/IP subnetwork type, over which it transmits and receives IP datagrams.

- Configuring IP Gateway nodes and routing functions

  This subsection describes pertinent parameters to include in your TCP-IP Stack configuration.

- Configuring DCA Bridging and TCP-IP stack attachments

  This subsection describes how TCP-IP Stack lets you configure an existing DNS network as a type of TCP/IP subnetwork. This capability lets you use a DNS network as part of a TCP/IP internet.

- Configuring network bridge nodes

  This subsection describes how TCP-IP Stack allows you to configure a DCP to function as a network bridge node between a TCP/IP network and a Telcon network that uses a different network layer routing protocol.

- Configuring host names

  This subsection explains how DCP TCP-IP Stack can associate a name (such as HOST1) with a unique IP address.

Before TCP-IP Stack and supporting program products are installed, define your Telcon network using configuration statements that are unique to each program product in addition to standard and modified Telcon configuration statements.

2.1. Before You Begin

2.1.1. Telcon Configuration Statement Reference Information

Section 3 provides complete reference information on unique TCP-IP Stack configuration statements. For standard or modified Telcon configuration statements, only the parameters related to TCP-IP Stack configurations are explained in this manual. Refer to the following manuals for complete descriptions of Telcon configuration statements and information on creating a Telcon configuration source file:

- *Telcon Configuration Guide* (7831 5678)
- *Telcon Configuration Reference Manual* (7831 5686)

The following manuals provide detailed information on configuration statements and configuration requirements for the program products used with TCP-IP Stack:

- *DCP Series LAN Products Configuration and Operations Guide* (7831 5512)
- *DCP Series X.25 Packet-Switched Communications Software (PSCS) Configuration and Operations Guide* (7831 5470)

2.1.2. DCP/OS Workstations

Each DCP in your network must have an associated DCP/OS workstation. There are two kinds of DCP/OS workstations: direct-connect and virtual.

- Direct-connect workstations are connected to DCP ports owned by DCP/OS. They are not part of the Telcon configuration, and are operable whether Telcon is or is not installed on the DCP.

- Virtual workstations are terminals in the Telcon configuration that are capable of performing most of the functions of a direct-connect workstation, but are not operable unless Telcon is operating.

The *Telcon Configuration Guide* (7831 5678) describes both types of workstations more fully.

You can have an operating direct-connect workstation before Telcon is installed. The workstation must be a UNISCOPE® terminal connected to a medium-speed, single-line line module. The workstation can be on any of the first 32 ports of the DCP.

The DCP/OS boot element, located on the diskette you insert in the DCP disk drive, automatically enables a direct-connect workstation. The DCP/OS boot element scans the ports until it finds a terminal that satisfies the criteria for a workstation. Alternatively, you can use the DCP/OS utility MONFIG to specify the workstation port or ports by making entries in the DCP physical device table (PDT). The *DCP/OS Operations Reference Manual* (7831 5702) tells how to use MONFIG.

2.2. Telcon Statements that Define TCP-IP Stack

When you add TCP-IP Stack to a Telcon configuration, use the following configuration statements to define TCP-IP Stack communications capabilities.

2.2.1. TCP-IP Stack Configuration Statements

The following configuration statements are used to configure TCP-IP Stack software:

ADDRESS

Description
Can be used to associate a configuration name with an IP address. The name of the statement can then be used in other statements as a value for a destination parameter.

This statement does not create entries in the local name directory. See the NSS and IPADR statements for information on TCP-IP Stack name/address association.

EU

Description
Identifies TCP-IP Stack as an end user. This statement is required to allow TCP-IP stack to complete initialization.

IPADR

Description
Assigns IP addresses and associates them with DCA endpoints allowing TCP-IP Stack to function as a network bridge node between TCP/IP and DCA networks.

Note: This statement can also be used to create host name directory entries.

NSM (optional)

Description
Identifies entries in the host name directory and maps an IP address to a locally specified host name.

NSS (optional)

Description
Defines the operational characteristics of the host name directory. Required if you are using the domain name system resolver.

SUBNET

Description
Defines the characteristics of subnetworks that support TCP/IP communications.

XEU

Description
Configures the following:

- access to the Telnet user application
- access to OS2200 DDN applications
- access to DCA applications over a TCP/IP transport

The IPADR, NSM, NSS, and SUBNET statements are unique to the TCP-IP Stack program product. The ADDRESS, EU and XEU statements are Telcon configuration statements that specify parameters used with TCP-IP Stack configurations.

2.2.2. Configuration Statements for LAN Attachments

The following configuration statements are required to configure LAN Platform software to support TCP-IP Stack:

LCLASS

Description
Specifies the line protocol handler for LAN intelligent line modules.

LINE

Description
STA specifies the physical LAN station address (subnetwork point of attachment, or SNPA) for the LAN line module or ILM 40 line module.

DCLUNIT (optional)

Description
HLETYPE specifies attachments for IP or ARP. On LLC (802.2) networks ARP does not require a separate attachment.

Starting with release 2R2, this statement is optional.

SAP (optional)

Description
RSHLE refers to the DLCUNIT statement name. If this is a MAC LAN attachment, the parameter ILMIF specifies the interface type ILMMAC, the parameter SAPOPT specifies the TCP-IP Stack interface option IPARP, and the parameter LSAP specifies either IPTYPE or ARPTYPE, depending on which attachment is being specified.

Starting with release 2R2, this statement is optional.

2.2.3. Configuration Statements for X.25 Attachments

The following configuration statements are required to configure X.25 PSCS software to support TCP-IP Stack:

LCLASS **Description** Specifies the line protocol handler for X.25 line modules (X25PKT).
LINE **Description** Identifies the specific X.25 line supporting the attachment.
PDNGRP **Description** Identifies packet-level protocol (PLP) and interface options for the TCP/IP X.25 attachment.
X25DEF **Description** Identifies unique characteristics of the attached PDN or DDN X.25 network.

The PDNGRP and X25DEF statements are unique to the X.25 PSCS program product. The LCLASS and LINE statements are Telcon configuration statements that specify parameters required for PSCS configurations.

2.2.4. Configuration Statements for Host Channel Attachments

The following Telcon configuration statements are required to configure TCP/IP communications over a channel connection to an OS 1100 host:

CHANNEL **Description** Defines a channel between an OS 1100 host and the local DCP.
DCATS **Description** Required for the host channel in TS/TN configurations.
TRUNK **Description** Required to configure the host channel to use DNS network layer protocol (the host is configured as a DNS node).

TCP-IP will not initialize the host channel as the only user of the channel. Therefore, even if only TCP-IP traffic is expected on the channel, either a DCATS host termination system or a DNS host trunk must be configured to initialize the channel.

2.3. Configuring TCP-IP Stack Attachments to LAN Subnetworks

The LAN attachments work with TCP-IP Stack to support TCP/IP communications over LAN subnetworks. TCP-IP Stack can use one of the following LAN attachments:

- 802.2 logical link control (LLC) type 1
- 802.3 media access control (MAC)
- 802.5 token ring
- FDDI

TCP-IP Stack implements the subnetwork access protocol (SNAP) to provide an OSI conformable attachment or an LLC type 1. Alternatively, internet protocol (IP) can interface directly with the MAC link layer service.

Because of different interface requirements, each of these LAN link layer services has different configuration requirements. You must configure TCP-IP Stack to use the same link layer service as the other TCP/IP stations on the LAN.

2.3.1. LAN Attachment

To configure TCP-IP Stack for a LAN attachment, include the following configuration statements in your configuration file:

Statement	Description	Required Parameters	Additional Information
PRCSR	Identifies the DCP as a processor to other Telcon entities.		
NETADR or DCPTS	Identifies this DCP as a DNS node. Identifies a DCP termination system on this DCP.		The NETADR statement assigns a DNS node number to the DCP. The DCPTS statement associates the DCP termination system with a related PRCSR statement.
LCLASS	Defines the LAN line class.	LPH=ILML, or =(ILML,'ENET$') =(ILML,'FDDI$') =(ILML,'TRNG$')	LCLASS defines the line protocol handler (LPH) for the LAN line module.

Statement	Description	Required Parameters	Additional Information
LINE	Identifies the specific LAN line supporting the attachment.	PRCSR	Name of the related PRCSR statement.
		CLASS	Name of the LAN related LCLASS statement.
		ADR	PPID of the LAN line module.
		STA	Six-octet (12 hexadecimal digit) LAN station address of the line module.
EU	Defines TCP-IP Stack as an end user.	PRCSR	Name of the related PRCSR statement.
		TYPE=TCPIP	TCP-IP Stack does not initialize properly without a TCPIP end user statement.
SUBNET	Defines the LAN as a TCP/IP subnetwork.	PRCSR	Name of the related PRCSR statement.
		TYPE=LANLLC TYPE=LAN TYPE=IPFDDI TYPE=IPTR	TYPE defines the type of LAN configured as a TCP/IP subnetwork.
		IPNETID	The network number (the IP address or subnetwork number) of this LAN subnetwork.
		LINE	Name of the LAN-related LINE statement. This identifies the referenced line as the physical interface to the LAN subnetwork.
IPADR	Assigns an IP address to the DCP LAN attachment.	PRCSR	Name of the related PRCSR statement.
		DCAEP	Name of the NETADR or DCPTS statement that identifies this DCP.
		IPADDR1 or IPADDR2	The IP address of this LAN attachment.

Example 1

The following example configures TCP-IP stack for LAN/LLC using the LAN platform.

```
** Processor Definition
*
DCP1      NETADR     NA=1
PRC1      PRCSR      NA=1
*
** LAN-Related Statements
*
LANLLC    LCLASS     LPH=ILML
LNP10B    LINE       PRCSR=PRC1,CLASS=LANLLC,ADR=X'0B',;
                     STA=X'08000B0C360B'
*
*
** TCP-IP Stack Statements
*
EUTCPIP   EU         PRCSR=PRC1,TYPE=TCPIP
SNLLCP1   SUBNET     PRCSR=PRC1,TYPE=LANLLC,LINE=LNP10B,;
                     IPNETID=(208,17,198)
IPP1H1    IPADR      PRCSR=PRC1,DCAEP=DCP1,;
                     IPADDR1=(208,17,198,217,LOCAL)
```

Example 2

The following example configures TCP-IP Stack for a basic MAC LAN attachment using the LAN line module.

```
** Processor Definition
*
HOST1     NETADR     NA=1
PRC1      PRCSR      NA=1
*
** LAN-Related Statements
*
LANLLC    LCLASS     LPH=ILML
LNP10B    LINE       PRCSR=PRC1,CLASS=LANLLC,ADR=X'0B',;
                     STA=X'08000B0C360B'
*
** TCP-IP Stack Statements
*
EUTCPIP   EU         PRCSR=PRC1,TYPE=TCPIP
SNLLCP1   SUBNET     PRCSR=PRC1,TYPE=LAN,LINE=LNP10B,;
                     IPNETID=(208,17,198)
IPP1H1    IPADR      PRCSR=PRC1,DCAEP=HOST1,;
                     IPADDR1=(208,17,198,217,LOCAL)
```

Example 3

The following example configures TCP-IP Stack for LAN LLC, using the LAN platform.

```
LANLC     LCLASS     LPH=ILML
LANLN     LINE       ...,CLASS=LANLC,...
LANSN     SUBNET     ...,TYPE=LANLLC,...
```

Example 4

The following example configures TCP-IP Stack for LAN, using the ILM40-EN line module.

```
LANLC       LCLASS      LPH=(ILML,'ENET$')
LANLN       LINE        ...,CLASS=LANLC,...
LANSN       SUBNET      ...,TYPE=LAN,...
```

Example 5

The following example configures TCP-IP Stack for LAN LLC, using the ILM40-EN line module.

```
LANLC       LCLASS      LPH=(ILML,'ENET$')
LANLN       LINE        ...,CLASS=LANLC,...
LANSN       SUBNET      ...,TYPE=LANLLC,...
```

Example 6

The following example configures TCP-IP Stack for FDDI, using the ILM40-FD line module.

```
FDDILC      LCLASS      LPH=(ILML,'FDDI$')
FDDILN      LINE        ...,CLASS=FDDILC,...
FDDISN      SUBNET      ...,TYPE=IPFDDI,...
```

Example 7

The following example configures TCP-IP Stack for a token ring LAN, using the ILM40-TR line module.

```
LANTR       LCLASS      LPH=(ILML,'TRNG$')
LANLN       LINE        ...,CLASS=LANTR,...
TRSN        SUBNET      ...,TYPE=IPTR,...
```

2.4. Configuring TCP-IP Stack Attachments to X.25 Subnetworks

The X.25 implementation used on the Internet conforms to standards developed for the U. S. Department of Defense (DoD) communications requirements. Although based on CCITT X.25 standards, it includes some capabilities that are generally not implemented in X.25 PDNs, and therefore presents some unique X.25 PSCS and TCP-IP Stack configuration requirements. This document refers to the Internet X.25 implementation as DDN X.25.

The X.25 implementations used on public data networks generally represent a subset of the communications capabilities defined by CCITT X.25 standards. Although many PDNs implement different subsets of X.25 capabilities, they do not usually include proprietary capabilities such as those found in DDN X.25. This document refers to public data network X.25 implementations as generic X.25.

Refer to the *X.25 Packet-Switched Communications Software (PSCS) Configuration and Operations Guide* (7831 5470) for detailed information about configuring X.25 subnetwork attachments.

2.4.1. Generic X.25 PDN Attachment

To configure TCP-IP Stack for a generic X.25 PDN attachment, include the following configuration statements in your configuration file:

Statement	Description	Required Parameters	Additional Information
PRCSR	Identifies the DCP as a processor to other Telcon entities.		
NETADR or DCPTS	Identifies this DCP as a DNS node. Identifies a DCP termination system on this DCP.		The NETADR statement assigns a DNS node number to the DCP. The DCPTS statement associates the DCP termination system with a related PRCSR statement.
X25DEF	Identifies characteristics of the specific PDN to which this DCP is attached.	NETWORK	Name of the PDN (from the list of PDN labels defined for PSCS
LCLASS	Defines the X.25 line class.	LPH=X25PKT	X25PKT defines the line protocol handler (LPH) for X.25 lines.
PDNGRP	Identifies X.25 packet-level protocol (PLP) attributes and other interface options associated with this attachment.	PDNGRP	Name of the related PRCSR statement.
		X25DEF	Name of the related X25DEF statement.
		VCGRP	Range of logical channel numbers assigned as virtual circuits (VCs), and the type of virtual circuit used for this attachment.
LINE	Identifies the specific X.25 line supporting the attachment.	PDNGRP	Name of the related PDNGRP statement.
		CLASS	Name of the X.25-related LCLASS statement.
		ADR	PPID of the X.25 line module.

Statement	Description	Required Parameters	Additional Information
EU	Defines TCP-IP Stack as an end user.	PRCSR	Name of the related PRCSR statement.
		TYPE=TCPIP	TCP-IP Stack will not initialize properly without a TCPIP end user statement.
SUBNET	Defines the X.25 PDN as a TCP/IP subnetwork.	PRCSR	Name of the related PRCSR statement.
		TYPE=IPPDN	IPPDN defines the X.25 PDN as a TCP/IP subnetwork.
		IPNETID	The network number (the IP address) of this X.25 subnetwork.
		LINE or PDNGRP	Name of the X.25-related LINE statement (for single-link attachments only). Name of the related PDNGRP statement (for multilink or single-link attachments). This identifies the referenced LINE or PDNGRP as the physical interface to the X.25 subnetwork. If the X.25 attachment supports multilink procedures, you must specify only the PDNGRP parameter. You must not specify individual lines of a multilink attachment. The PDNGRP parameter works for either attachment type (see 2.4.3).

Statement	Description	Required Parameters	Additional Information
IPADR	Assigns an IP address to the DCP X.25 attachment.	PRCSR	Name of the related PRCSR statement.
		DCAEP	Name of the NETADR or DCPTS statement that identifies this DCP.
		IPADDR1 or IPADDR2	The IP address of this X.25 attachment (must be flagged LOCAL).
IPADR	Specifies an IP address for each remote TCP/IP host with which you want to communicate across the X.25 network and pairs that address to a DTE address.	PRCSR	Name of the related PRCSR statement.
		IPADDR1 or IPADDR2	The IP address of the remote TCP/IP host (must not be flagged LOCAL).
		DTEADR	The assigned DTE address of the remote host in the X.25 PDN (must be supplied by PDN administrator). The DTE address is a quoted string of decimal numbers, up to 15 digits long. You must use a separate IPADR statement to configure the unique IP address and DTE address pair for each host you want to reach in a generic X.25 network (see 2.4.3).

Example

This example configures an attachment to an X.25 network, network number 17. The DCP's address is 17.0.0.5, and a TCP-IP host with address 17.0.0.6 is reachable by DTE address 21601004647.

```
** Processor Definition
*
DCP1        NETADR      NA=1
PRC1        PRCSR       NA=1
*
** X.25 PDN Definition
*
X25DFTE1    X25DEF      NETWORK=TELENET
PGPP1TE1    PDNGRP      PRCSR=PRC1,VCGRP=(20,30),;
                        X25DEF=X25DFTE1
X2596       LCLASS      LPH=X25PKT
LNP113      LINE        PDNGRP=PGP1TE1,CLASS=X2596,ADR=13
*
** TCP-IP Stack Definition
*
EUTCP       EU          PRCSR=PRC1,TYPE=TCPIP
SNPDNP1     SUBNET      PRCSR=PRC1,TYPE=IPPDN,;
                        PDNGRP=PGPP1TE1,IPNETID=(17,0,0,0)
IPP1H1      IPADR       PRCSR=PRC1,DCAEP=DCP1,;
                        IPADDR1=(17,0,0,5,LOCAL),;
IPP1H2      IPADR       PRCSR=PRC1,;
                        IPADDR1=(17,0,0,6),;
                        DTEADR='21601004647'
```

2.4.2. DDN X.25 (Internet) Attachment

To configure TCP-IP Stack for a DDN X.25 attachment, include the following configuration statements in your configuration file:

Statement	Description	Required Parameters	Additional Information
PRCSR	Identifies the DCP as a processor to other Telcon entities.		
NETADR or DCPTS	Identifies this DCP as a DNS node. Identifies a DCP termination system on this DCP.		The NETADR statement assigns a DNS node number to the DCP. The DCPTS statement associates the DCP termination system with a related PRCSR statement.
X25DEF	Identifies characteristics of the specific DDN to which this DCP is attached.	NETWORK	This specifically defines the characteristics of DDN X.25 subnetworks in Internet.
LCLASS	Defines the X.25 line class.	LPH=X25PKT	X25PK defines the line protocol handler (LPH) for the X.25 lines.
PDNGRP	Identifies X.25 packet-level protocol (PLP) attributes and other interface options associated with this attachment.	PDNGRP	Name of the related PRCSR statement.
		X25DEF	Name of the related X25DEF statement.
		VCGRP	Range of logical channel numbers assigned as virtual circuits (VCs), and the type of virtual circuit used for this attachment.
		DTEADR	The assigned DTE address of the local attachment to the X.25 DDN (must be supplied by DDN administrator). The DTE address is a quoted string of decimal numbers, up to 15 digits long.

Statement	Description	Required Parameters	Additional Information
LINE	Identifies the specific X.25 line supporting the attachment.	PDNGRP	Name of the related PDNGRP statement.
		CLASS	Name of the X.25-related LCLASS statement.
		ADR	PPID of the X.25 line module.
EU	Defines TCP-IP Stack as an end user.	PRCSR	Name of the related PRCSR statement.
		TYPE=TCPIP	TCP-IP Stack will not initialize properly without a TCPIP end user statement.
IPADR	Assigns an IP address to the DCP X.25 attachment.	PRCSR	Name of the related PRCSR statement.
		DCAEP	Name of the NETADR or DCPTS statement that identifies this DCP.
		IPADDR1 or IPADDR2	The IP address of this X.25 attachment (must be flagged LOCAL).

Statement	Description	Required Parameters	Additional Information
SUBNET	Defines the X.25 DDN as a TCP/IP subnetwork.	PRCSR	Name of the related PRCSR statement.
		TYPE=DDN	DDN defines the TCP/IP subnetwork type as DDN X.25. Use the label DDN only if connecting to an actual DoD DDN X.25 network. These X.25 networks support a specific DTE addressing scheme that allows an address conversion algorithm to turn an IP address into a DTE address. While this algorithm can be implemented on any private network, it is not compatible with most X.25 networks DTE addressing schemes, nor is it compatible with CCITT recommendation X.121. In addition to the addressing, the DDN X.25 networks use a set of proprietary X.25 facilities. Unless the non-DDN X.25 network can ignore them, these facilities may cause problems and prevent completing connections across the network.
		IPNETID	The network number (the IP address) of this X.25 subnetwork.
		PDNGRP	This identifies the referenced PDNGRP as the physical interface to the X.25 subnetwork. The PDNGRP parameter works for either attachment type (see Section 2.4.3).

The following terminology is used exclusively for the DDN address:

Term	Format
Network	NN
Host or Trunk	HH
PSN	II
Class A	NN,HH,00,II
Class B	NN,NN,HH,II

Example

This example illustrates configurations where an X.25 link connects to a DDN network. The TCP/IP network number is 26, the host number is 4, and the PSN trunk is 5.

```
** Processor Definition
*
DCP1        NETADR      NA=1
PRC1        PRCSR       NA=1
*
** X.25 DDN Definition:
*
X25DFDDN    X25DEF      NETWORK=DDNX25
PGPP1DDN    PDNGRP      PRCSR=PRC1,VCGRP=(31,50),;
                        X25DEF=X25DFDDN,;
                        DTEADR='000000050400'
X2596       LCLASS      LPH=X25PKT
LNP123      LINE        PDNGRP=PGP1DDN,CLASS=X2596,ADR=23
*
*
** TCP-IP Stack Definition:
*
EUTCP       EU          PRCSR=PRC1,TYPE=TCPIP
SNP1DDN     SUBNET      PRCSR=PRC1,TYPE=DDN,PDNGRP=PGPP1DDN,;
                        IPNETID=26
IPP1DDN     IPADR       PRCSR=PRC1,DCAEP=DCP1,;
                        IPADDR=(26,4,0,5,LOCAL)
```

2.4.3. Configuring Unique X.25 Capabilities

When you configure an X.25 attachment, you must make a number of configuration choices that depend on the network type and its specific capabilities. The SUBNET and IPADR statements provide some optional parameters to configure unique X.25 capabilities.

2.4.3.1. Configuring IP-to-DTE Address Pairings

For generic X.25 attachments, you must include an additional IPADR statement for each remote TCP/IP host reachable (across the PDN) through the attachment. Each IPADR statement that refers to a remote host must include an IPADDR parameter (not flagged LOCAL) and a DTEADR parameter to associate the DTE address of a specific remote host with a unique IP address. You can use either IP address parameter (IPADDR1 or IPADDR2).

When you configure an IPADR statement that uses both IP address parameters, the DTE address of the remote host pairs with the IP address that is **not** flagged LOCAL.

If **neither** IP address is flagged LOCAL, the DTE address pairs with the IP address specified by parameter IPADDR1.

Example

This example configures two IP address-DTE address pairs. The first pair is configured on IPADR1 to identify a specific remote TCP/IP host. This would, for example, allow terminals to use the TCP-IP Stack User TELNET application to connect to that host across the X.25 network.

The second pair is configured on IPADR2, which configures TRUNK1 to bridge the X.25 network. The IPADDR2 and DTEADR parameters provide the addressing information for the DCP at the remote end of the trunk.

```
IPADR1    IPADR     PRCSR=PRCSR1,IPADDR1=(34,24,0,125),;
                    DTEADR='18010007736'
IPADR2    IPADR     PRCSR=PRCSR1,DCAEP=TRUNK1,;
                    IPADDR1=(34,45,0,32,LOCAL),;
                    IPADDR2=(34,17,0,81),;
                    DTEADR='18010002483'
```

2.4.3.2. Configuring X.25 Single-link and Multilink Attachments

The X.25 PSCS program product can implement two types of physical attachments to an X.25 network:

- Single-link attachments are configured as one line per DTE address (one LINE statement per PDNGRP statement).
- Multilink attachments are configured as two or more lines per DTE address (multiple LINE statements per PDNGRP statement).

You must configure a SUBNET statement to identify the X.25 link or links that TCP-IP Stack uses to attach to the X.25 network. All LINE statements, whether for single-link or multilink configurations, must refer to a PDNGRP statement. Therefore, you should use the PDNGRP parameter on the SUBNET statement to specify all X.25 attachments.

2.4.3.3. Configuring the Number of Virtual Circuits Per Connection

When making a connection to another host across an X.25 network, PSCS establishes a virtual circuit (VC) route across the network. Each VC has a bandwidth limited by the packet size and window size allowed by the network. IP may try to increase the bandwidth of a given X.25 connection by requesting additional VCs when PSCS signals that data is backing up on an existing VC.

The drawback of this method is that a host can be limited in the number of VCs it can support simultaneously, either because of software limitations, or because of its subscribed connection to the X.25 network. Establishing additional VCs may use resources that should remain available for other connections.

You can limit the number of VCs used per connection by specifying a value for the IPMAXVC parameter on the SUBNET statement. If you do not include the IPMAXVC parameter it defaults to zero, indicating that no limit is set.

Example

The following example sets the maximum number of virtual circuits per TCP/IP connection to 3.

```
SNPDN1     SUBNET     ...,IPMAXVC=3,...
```

2.5. Configuring a Host Channel as a TCP/IP Subnetwork

Although the channel configuration has the same basic requirements as a DNS channel, the related TCP-IP Stack statements and the matching CMS 1100 configuration enable TCP/IP communications over the channel without dependency on DNS protocols. TCP-IP Stack treats the host channel as an entire TCP/IP network.

CMS 1100 software on the host must be configured to use TCP/IP communications over the host channel. CMS 1100 level 7R3 (SB4R5) or a later release is required to configure TCP/IP communications.

Note: *This configuration does not apply to LAN attached hosts. When an OS 1100 host is connected to a LAN through a Host LAN Controller (HLC), TCP-IP Stack communicates with the host as with any other TCP/IP host connected to the LAN.*

To configure a host channel as a TCP/IP subnetwork, include the following network definition statements in your configuration file:

Statement	Description	Required Parameters	Additional Information
PRCSR	Identifies this processor as a DNS node to other Telcon entities.	NA=*n*	The variable *n* represents the DNS network address or node number you assign to this node. Each node in a Telcon DNS network must have a unique address.
NETADR (optional)	Assigns a name (label) to a DNS network address reachable through this attachment, such as the DCP itself, a channel-attached host, or a remote host in the Telcon DNS network.	NA=*n*	The variable *n* represents any valid DNS hierarchical network address for this Telcon DNS network. It can be a subdomain, super cluster, simple cluster, or node address. For DNS configurations, multiple TRUNK and XEU statements can reference the same NETADR name. The NETADR allows you to label an often-referenced DNS address at a single point in your configuration, then use the label for all references.

Statement	Description	Required Parameters	Additional Information
TRUNK	Defines the attachment to the host channel subnetwork. The host channel must be configured as a logical trunk.	PRCSR	The name of the related PRCSR statement for this trunk. For a trunk configured over a host channel, you specify only the name of the attached processor.
		LSUTYPE=DNS	
CHANNEL	Configures the physical channel connection between the DCP and the host.	PPID	The DCP port processor number to which the channel is connected.
		TRUNK	The name of the related TRUNK statement for this channel.
EU	Defines TCP-IP Stack as an end user.	PRCSR	Name of the related PRCSR statement.
		TYPE=TCPIP	The TCP-IP Stack will not initialize properly without a TCPIP end user statement.
SUBNET	Defines the channel as a TCP/IP subnetwork.	PRCSR	Name of the related PRCSR statement for this DCP.
		TYPE=IPCHAN	Identifies the host channel as a TCP/IP subnetwork type.
		IPNETID	The IP network number assigned to the host channel subnetwork. The host channel subnetwork has a unique network number assigned by the TCP/IP network authority.
		CHANNEL	The name of the related CHANNEL statement for this host channel.

Statement	Description	Required Parameters	Additional Information
IPADR	Assigns an IP address to the DCP's host channel attachment.	PRCSR	Name of the related PRCSR statement for this DCP.
		DCAEP	The name of the NETADR statement that assigns a DNS network address to this DCP.
		IPADDR1	The complete IP address assigned to the host channel attachment. You must include the LOCAL flag on this parameter.

Example

This example shows a trunk connected to a CMS 1100. The CMS 1100 is running its own version of TCP/IP. The network number for the channel is 173.53.130. The address of the DCP is 172.53.130.42.

The TRUNK statement sets up a DNS connection over the CHANNEL. Alternatively, a TSTN SESSN can be configured over the CHANNEL. TCP-IP Stack cannot initialize the CHANNEL by itself. Either TRUNK or SESSN must be used.

```
** Host and processor definitions
*
DCP1        NETADR     NA=1
PRC1        PRCSR      NA=1
*
** DNS host channel definition
*
TKCP1H1     TRUNK      PRCSR=PRC1,LSUTYPE=DNS
CHANH1      CHANNEL    PPID=X'0E',TRUNK=TKCP1H1
*
** TCP-IP Stack configuration
*
EUTCP       EU         PRCSR=PRC1,TYPE=TCPIP
SNCHAP1     SUBNET     PRCSR=PRC1,TYPE=IPCHAN,;
                       IPNETID=(172,53,130),CHANNEL=CHANH1
IPP1H1      IPADR      PRCSR=PRC1,DCAEP=DCP1,;
                       IPADDR1=(172,53,130,42,LOCAL)
```

2.6. Configuring IP Gateway Nodes and Routing Functions

TCP-IP Stack allows you to configure a DCP to perform IP routing functions within a TCP/IP network. With TCP-IP Stack, you can configure the following IP gateway capabilities on a DCP:

- IP routing
- Subnet routing
- Configured IP routes
- Routing Information Protocol (RIP)
- IP broadcast addresses
- Autonomous system numbers

TCP-IP Stack IP gateway capabilities are configured with SUBNET statement parameters. Host names are configured with IPADR statement parameters.

2.6.1. Enabling IP Routing

The IPROUTER parameter on the SUBNET statement enables TCP-IP Stack IP routing on a specific subnetwork. TCP-IP Stack can operate as an IP gateway between any TCP/IP subnetwork attachments that have IP routing enabled. It can route messages received through one attachment to another attachment, based on IP addresses in the message datagrams and on the routing information it maintains internally.

The IPROUTER parameter is optional. To enable IP routing on a particular subnetwork, you must explicitly declare IPROUTER=YES on the related SUBNET statement. Otherwise, IP routing is disabled for that particular subnetwork.

Example

```
SULAN1      SUBNET    ...,IPROUTER=YES,...
SULAN2      SUBNET    ...,IPROUTER=YES,...
```

2.6.2. Configuring Subnet Routing

You can configure TCP-IP Stack to support subnet routing in local networks. Subnet routing allows an IP gateway to route datagrams to a local network (group of subnetwork attachments) which is identified to the rest of the TCP/IP network by a single network number. The local gateway node differentiates destinations in the local network by means of a subnet mask, which permits a nonstandard interpretation of part of the host number. For TCP-IP Stack subnet routing configurations, you can use one of two methods to specify a subnet mask:

- TCP-IP Stack can automatically calculate an appropriate subnet mask.
- You can explicitly configure a subnet mask.

2.6.2.1. Calculated Subnet Mask

For class A or class B networks, TCP-IP Stack can calculate a subnet mask that uses a one-byte portion of the host number. Each subnetwork attachment you configure requires an `IPNETID` parameter on the related SUBNET statement to define its network number. For a standard class B network number, you would normally define the first two bytes of the IP address; for example, `IPNETID=(132,47,0,0)`.

For each subnetwork in a subnet routing configuration, include the locally-defined subnet number in the most significant one-byte portion of the host number; for example, `IPNETID=(132,47,2,0)`. TCP-IP Stack recognizes the extra byte as a subnet extension to a class B address. From the `IPNETID` information, TCP-IP Stack automatically calculates the appropriate subnet mask value for all subnetworks in the configuration.

This method applies only to class A and class B networks. For class C networks, you must use the alternative method of explicitly configuring a subnet mask value.

2.6.2.2. Configured Subnet Mask

To explicitly configure a subnet mask value, you must include a SUBNMASK parameter on each SUBNET statement in your local network. This method applies if you chose to use less than a whole byte of the host number to assign local network numbers (for example, in a class C network). The following guidelines apply to configured subnet masks:

- The mask value must be two bits, minimum. A one-bit mask value is not allowed because it interferes with proper interpretation of IP broadcast addresses.
- Incoming IP addresses with "all ones" at the mask bit locations represent IP broadcast addresses to hosts on all subnetworks associated with the mask value. (Earlier implementations of the TCP/IP protocol suite may use a value of "all zeroes".)
- IP addresses with "all zeroes" at the mask bit locations are interpreted as "this subnetwork." IP routers do not route datagrams to "all zeroes" addresses.
- The subnet mask value represents 2^n-2 subnetworks, where n is the number of bits you use for the mask.

Example

Your local class C network consists of two subnetworks. The assigned network number is 197.36.241. A class C address allows only one byte for host numbers, which defines up to 254 individual destinations.

To create two subnetworks, you can divide this byte (8 bits) further, into three bits for the subnetwork number, leaving five bits for the host number. This way you have allowed for up to six subnetworks, with up to 30 hosts on each subnetwork. You assign subnetwork numbers 1 and 2 to your first two subnetworks. An example of the network topology is shown in Figure 2-1.

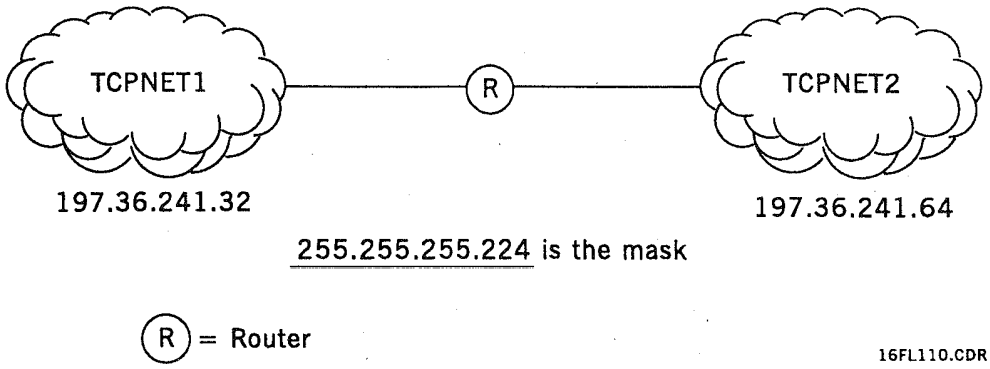


Figure 2-1. Subnetwork Topology

The SUBNET statements specify an IPNETID of (197,36,241,32) for subnetwork 1, and (197,36,24,64) for subnetwork 2. Each SUBNET statement specifies a SUBNMASK parameter of (255,255,255,224).

2.6.3. Configuring IP Routes

With TCP-IP Stack, you can configure specific routes to destinations in a TCP/IP network.

2.6.3.1. Configuring IP Routes to Other Networks

You must include a unique SUBNET statement for each configured route that TCP-IP Stack can use to send data to a remote destination. TCP-IP Stack can maintain more than one route to the destination network, each specified using a SUBNET statement, or acquired through RIP exchanges.

To identify a route through a gateway to a detached network, you must define the destination network number with the IPNETID parameter. You must also include an IPGATEWY parameter to identify the complete IP address of the gateway and its distance (in hops) from the destination network. These parameters form a route that TCP-IP Stack uses when sending datagrams to a destination on the network identified by this SUBNET statement.

If multiple routes are available, TCP-IP Stack selects the route that represents the shortest distance, using the distance subparameter on the IPGATEWY parameter. If two gateways exist at the same distance from the destination network, the distance parameter can be used to set priorities for the routes.

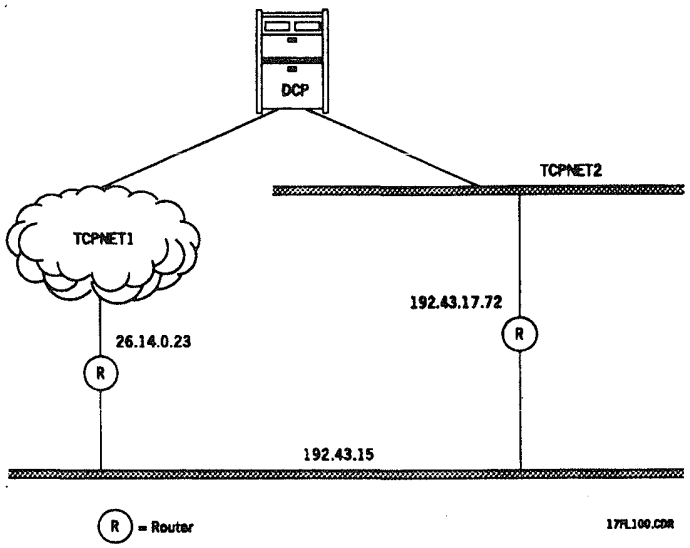


Figure 2–2. Routes to Default Gateways

Example

TCP-IP Stack can reach network 192.43.15 through at least two gateways, one at 26.14.0.23, and one at 192.43.17.72. The first gateway provides a path through subnetwork 192.43.17, while the second gateway defines a path through subnet 26.

```
TCPNET1    SUBNET    ...,IPNETID=26,...
TCPNET2    SUBNET    ...,IPNETID=(192,43,17),...
TCPGW1     SUBNET    ...,IPNETID=(192,43,15),IPGATEWY=(1,192,43,17,72),...
TCPGW2     SUBNET    ...,IPNETID=(192,43,15),IPGATEWY=(2,26,14,0,23),...
```

2.6.3.2. Configuring Routes to Default Gateways

You can identify default gateways using the label DEFLTGWY on the IPNETID parameter. Default gateway configurations do not use the SUBNMASK parameter.

When you specify the DEFLTGWY parameter, you must also include a corresponding IPGATEWY parameter. The IPGATEWY parameter specifies the distance (in hops) between the Telcon TCP-IP node and the default gateway, and the IP address of the default gateway.

2.6.4. Configuring Routing Information Protocol (RIP)

TCP-IP Stack supports an internal gateway protocol, called Routing Information Protocol (RIP), a widely-used de facto standard, described in RFC 1058. RIP can be activated on many directly connected networks. If there are no other RIP gateways on a network, RIP introduces additional overhead without adding any value.

Note: *As of release 2R2, RIP is supported on SUBNETs of TYPE=LAN, LANLLC, IPTR, IPFDDI, and IPDNS.*

To configure RIP for a specific subnetwork attachment, include the RIP parameter on the related SUBNET statement. If multiple SUBNET statements for the same network specify the RIP parameter, only the first instance is used by the software.

Format

```
name      SUBNET    ...,RIP=(activity,cost,timeout,count,delay,port),...
```

2.6.4.1. Activating RIP

RIP is activated by specifying YES as the first sub-parameter. RIP can be deactivated by specifying NO as the first sub-parameter. If no value is specified, the first sub-parameter default is NO.

If TCP-IP Stack in the DCP is physically connected to only one network, or routing is turned off for this network, then RIP can still be activated, allowing it to acquire routing information from other RIP gateways. To keep RIP from generating unneeded traffic on this network, specify SILENT as the first sub-parameter.

Format

```
name      SUBNET    ...,RIP=({NO|YES|SILENT},,,,,),...
```

2.6.4.2. Specifying the Cost to Route

The cost metric is specified by the second sub-parameter. It should reflect the relative cost (in hops) of routing through this network. This cost is added to all routes acquired from other RIP gateways on this network to calculate the cost of routing through this network relative to routes discovered on other networks.

The cost metric can range from one to 14. Be careful when using this metric, since it reduces the maximum range of operation for RIP. The default is one.

Format

```
name      SUBNET    ...,RIP=(,n,,,,),...
```

2.6.4.3. Specifying the Routing Update Timeout

When gateways crash it is difficult to determine their status, unless a way to track their activity is implemented. RIP does this by setting a timer on a route. When the timer expires a specific number of times, RIP assumes that the gateway from which the route was received is no longer active and the route is unreachable.

To prevent the timer from expiring, RIP transmits its routing information periodically. The routing update resets the timer. The interval for updates is specified by the third sub-parameter. It defaults to 30 seconds.

Format

```
name      SUBNET    ...,RIP=(,,n,,,),...
```

2.6.4.4. Specifying the Route Timeout Count

The fourth sub-parameter is used in conjunction with the routing update timeout value. It specifies the number of times that the routing update timer can expire before a route becomes unreachable. It defaults to six.

Format

```
name      SUBNET    ...,RIP=(,,,n,,),...
```

2.6.4.5. Specifying the Update Delay Time

The fifth subparameter specifies the time in seconds that RIP waits before transmitting information about lost routes. It defaults to 300.

Format

```
name      SUBNET    ...,RIP=(,,,,n,),...
```

2.6.5. Configuring the IP Broadcast

Only the LAN, IPFDDI, and LANLLC subnetwork types currently support broadcasting. A broadcast on a LAN always uses the same destination station address (X'FFFFFFFFFFFF'). However, the message being broadcast specifies its own IP broadcast address.

The IP broadcast address was originally specified to be all zeros: 0.0.0.0 – also written as {0,0}. The current standard IP broadcast address is all ones: 255.255.255.255 – also written as {-1,-1}. Not all implementors adopted the new standard, so some implementations still use a broadcast address of {0,0}.

TCP-IP Stack recognizes both broadcast address types in incoming datagrams. It transmits only the new type (all ones) unless you specify the old type (all zeros) on the IPBRDCST parameter.

Format

```
name      SUBNET    ...,IPBRDCST=0,...
```

2.6.6. Configuring Autonomous System Numbers

The autonomous system number is a device used to prevent routing information local to one autonomous system from propagating into other autonomous systems. Among other things, this helps to keep routing tables down to a manageable size.

The autonomous system number cannot be used to split a subnetworked network into multiple autonomous systems.

If the DCP is part of more than one autonomous system, then the autonomous system number must be specified for each network. Otherwise it is assumed to be one. The value is set using the AUTONUM parameter.

Format

```
name      SUBNET    ...,AUTONUM=n,...
```

The autonomous system number can range in value from 1 to 65,535.

2.7. Configuring Telcon as a TCP-IP Subnetwork

TCP-IP Stack works with standard Telcon software to support TCP/IP communications over Unisys DCA networks that use dynamic network services (DNS) protocols. DNS protocols provide connectionless network layer (internet layer) routing services for local area and wide area networks that normally carry DCA communications.

The DNS network functions as an intervening subnetwork between TCP-IP Stack nodes that are attached to other types of TCP/IP subnetworks. DNS protocols can route IP datagrams across any subnetwork type configured to support DNS routing within the network.

You can also enable RIP within the DNS network to distribute configuration information to use the dynamic neighbor discovery feature. The dynamic neighbor discovery feature maps IP addresses to the DNS addresses of TCP-IP Stack nodes in the network, and lets the TCP-IP Stack nodes exchange RIP routing information.

Note: *TCP-IP Stack release 2R2 and higher does not support more than one SUBNET statement of this type per Telcon node.*

2.7.1. Telcon DNS Subnetworks

DNS conducts all node-to-node routing within the Telcon DNS network, but TCP-IP Stack is required on any node that exchanges IP datagrams with other TCP/IP subnetwork types. For example, TCP-IP Stack is required on a node that routes IP datagrams between the Telcon DNS network and a DDN X.25 attachment. However, TCP-IP Stack is not required on a node that routes IP datagrams only to other nodes in the Telcon DNS network. Only Telcon nodes configured to use DNS can relay IP datagrams across a DNS network.

For a DNS trunk attachment, include the following configuration statements in your configuration file:

Statement	Description	Required Parameters	Additional Information
PRCSR	Identifies this processor as a DNS node to other Telcon entities.	NA=*n*	The variable *n* represents the DNS network address or node number you assign to this node. Each node in a Telcon DNS network must have a unique address.
	If you want to associate a DNSINFO statement with this node, include the following parameter:	DNSINFO	Name of the related DNSINFO statement.
NETADR (optional)	Assigns a name (label) to a DNS network address reachable through this attachment, such as the DCP itself, a channel-attached host, or a remote host in the Telcon DNS network.	NA=*n*	The variable *n* represents any valid DNS hierarchical network address for this Telcon DNS network. It can be a subdomain, super cluster, simple cluster, or node address. For DNS configurations, multiple TRUNK and XEU statements can reference the same NETADR name. The NETADR allows you to label an often-referenced DNS address at a single point in your configuration, then use the label for all references.
LCLASS	Define the line class for DNS trunk attachments. This statement applies to the attachments at both ends of the trunk.	LPH=UDLCL	UDLCL defines the line protocol handler for the UDLC lines that interconnect Telcon DNS nodes.

Statement	Description	Required Parameters	Additional Information
TRUNK	Defines the attachment to the DNS subnetwork.	PRCSR	The names of the related PRCSR statements for this trunk. For a typical DNS trunk using UDLC lines, you must specify the names of both processors (local and remote) connected by this trunk. If you want to configure an attachment to a LAN emulated UDLC trunk, refer to the *LAN Platform Configuration and Operations Guide* for detailed information.
		LSUTYPE=DNS	Identifies DNS network layer protocol as the link service user for this trunk. The label DNS implies the use of connection-oriented UDLC data link service over this trunk.
LINE	The TRUNK and CLASS statements identify the specific line supporting the trunk attachment on the DCP.	TRUNK	Name of the related TRUNK statement (the parent facility of this line). Both LINE statements refer to the same TRUNK statement.
		CLASS	Name of the related LCLASS statement.
		ADR	PPID of the line module controlling this line.

Statement	Description	Required Parameters	Additional Information
STATION	Two statements, each of which defines UDLC station attributes for one of the DCPs connected by this trunk.	LINE	The name of the related line statement (the parent facility of this UDLC station).
		RSHLE	The name of the related TRUNK statement. In this case, the trunk is the Telcon higher-level entity associated with each UDLC station. Both STATION statements refer to the same TRUNK statement.
		LSA	The local station address you assign to the local end of the trunk.
		RSA	The remote station address you assign to the remote end of the trunk. Both the LSA and RSA parameters are required for DNS trunks using UDLC lines. These parameters have a reciprocal relationship, the LSA for the local DCP is the RSA for the remote DCP.
EU	Define TCP-IP Stack as an end user.	PRCSR	Name of the related PRCSR statement.
		TYPE=TCPIP	The TCP-IP Stack will not initialize properly without a TCPIP end user statement.
SUBNET	To define Telcon DNS as a TCP/IP subnetwork.	PRCSR	Name of the related PRCSR statement for this DCP.
		TYPE=IPDNS	IPDNS defines the DNS network as a TCP/IP subnetwork type.
		IPNETID	The IP network number assigned to this Telcon DNS network. This is a unique number assigned by the TCP/IP network administrator.

Statement	Description	Required Parameters	Additional Information
IPADR	Assigns an IP address to the DNS subnetwork.	PRCSR	Name of the related PRCSR statement.
		DCAEP	Name of the NETADR statement that identifies this DCP as a DNS node.
		IPADDR1 or IPADDR2	The IP address of this DCP, which identifies it to the attached TCP/IP network. (must be flagged LOCAL).
IPADR (optional)	Additional statements to identify other TCP-IP Stack nodes in the DNS network as RIP neighbor nodes. In your local configuration file, you need to configure only a small subset of the RIP neighbor nodes to enable dynamic nieghbor discovery. If each TCP-IP Stack node in the network has at least one other RIP neighbor identified in its associated configuration file, all TCP-IP STACK nodes can automatically exchange RIP information.	PRCSR	Name of the related PRCSR statement.
		NA	The DNS address of the RIP neighbor (as specified on the related PRCSR or NETADR statement)
		IPADDR1 or IPADDR2	The assigned IP address of the RIP neighbor only for dynamic neighbor discover. You can use either IPADDR parameter to identify an RIP neighbor. You can optionally assign a name to each RIP neighbor, using the NAME1 or NAME2 parameters.

Example

```
** Processor 1 Definition
*
DCP1        NETADR      NA=1
PRC1        PRCSR       NA=1
*
** Processor 2 Definition
*
DCP2        NETADR      NA=2
PRC2        PRCSR       NA=2
*
*
** TRUNK/LINE/STATION Definitions
*
UDLC192     LCLASS      LPH=UDLCL,SPEED=19200
TKDP1P2     TRUNK       PRCSR=(PRC1,PRC2),LSUTYPE=DNS
LNP11C      LINE        TRUNK=TKDP1P2,CLASS=UDLC192,ADR1=X'1C',ADR2=X'1D'
STDP1P2     STATION     LINE=LNP11C,RSHLE=TKDP1P2,
*
** TCP-IP Stack Definition
*
EUTCP1      EU          PRCSR=PRC1,TYPE=TCPIP
EUTCP2      EU          PRCSR=PRC2,TYPE=TCPIP
SNP1DNS     SUBNET      PRCSR=PRC1,TYPE=IPDNS,;
                        IPNETID=(212,254,32)
SNP2DNS     SUBNET      PRCSR=PRC2,TYPE=IPDNS,;
                        IPNETID=(212,254,32)
IPP1H1      IPADR       PRCSR=PRC1,DCAEP=DCP1,;
                        IPADDR1=(212,254,32,5,LOCAL)
IPP2H2      IPADR       PRCSR=PRC2,DCAEP=DCP2,;
                        IPADDR1=(212,254,32,6,LOCAL)
*
** RIP Neighbor Definitions
*
IPP1RIP     IPADR       PRCSR=PRC1,NA=2,;
                        IPADDR1=(212,254,32,6,RIPNBR)
IPP2RIP     IPADR       PRCSR=PRC2,NA=1,;
                        IPADDR1=(212,254,32,5,RIPNBR)
```

2.6.4. Configuring Dynamic Neighbor Discovery

The Telcon network can include several nodes running TCP-IP Stack. DCP TCP-IP Stack must know the IP address and associated Telcon network address of each of these nodes.

Note: *If you are not configuring Telcon as a TCP/IP network (TYPE=IPDNS) then you can ignore this section.*

On some networks (IEEE 802 and FDDI networks) mechanisms exist to interview all connected hosts at once, eliciting a reply about the location of a host owning a specific IP address. However, most networks, including Telcon, do not have such a mechanism.

TCP-IP Stack implements a technique called dynamic neighbor discovery. This method allows all TCP-IP Stack nodes in the Telcon network to discover all other TCP-IP Stack nodes, while configuring only a small subset of TCP-IP Stack nodes.

2.7.2.1. Configuring RIP Neighbor Addresses

You can identify other Telcon TCP-IP nodes as RIP gateways by including the RIPNBR flag on a related IPADR statement. You must configure RIPNBR gateways only if you use routing information protocol across a DNS network (subnetwork type IPDNS) that does not support a broadcast mechanism.

Format

```
name        IPADR      ...,IPADDR1=(nn,nn,nn,nn,RIPNBR),...
```

or

```
name        IPADR      ...,IPADDR2=(nn,nn,nn,nn,RIPNBR),...
```

Either IP address parameters can be used. The address must be flagged RIPNBR (RIP neighbor).

Example

This example configures two RIP neighbors in a DNS network.

```
PRCSR1      PRCSR      ...
DCP1        NETADR      NA=4
DCP2        NETADR      NA=5
   .
   .
   .
IPADR1      IPADR      PRCSR=PRCSR1,IPADDR1=(213,247,101,23,RIPNBR),;
                       NA=4,NAME1='ROUTER-1'
IPADR2      IPADR      PRCSR=PRCSR1,IPADDR2=(213,247,101,13,RIPNBR),;
                       NA=5,NAME2='ROUTER-2'
```

2.7.2.2. Assigning the Telcon DNS Node Address

Format

```
name      IPADR      ...,NA=(sd,sc,sic,nn),...
```

This parameter specifies the Telcon network address using the same format as that used on other Telcon configuration statements that use the NA parameter.

Example 1

This example, shown in Figure 2–3, assumes that there are at least four Telcon TCP-IP nodes, assigned IP addresses 210.15.1.21, 210.15.1.22, 210.15.1.23, and 210.15.1.24, respectively. The Telcon nodes at IP addresses 210.15.1.21, 210.15.1.22, and 210.15.1.23 are configured to use only the node at IP address 210.15.1.24 as a central RIP neighbor. The central node at 210.15.1.24 is configured to use only 210.15.1.23 as an RIP neighbor.

This scheme allows all Telcon TCP-IP nodes in the network to eventually discover each other and exchange RIP information. If you added nodes to this network, you could configure the additional nodes to use either 210.15.1.24 or 210.15.1.23 as a central RIP neighbor. These two nodes are mutually configured as RIP neighbors and directly exchange RIP information about the rest of the network.

Note: *RIP must be turned on in the subnetwork statement.*

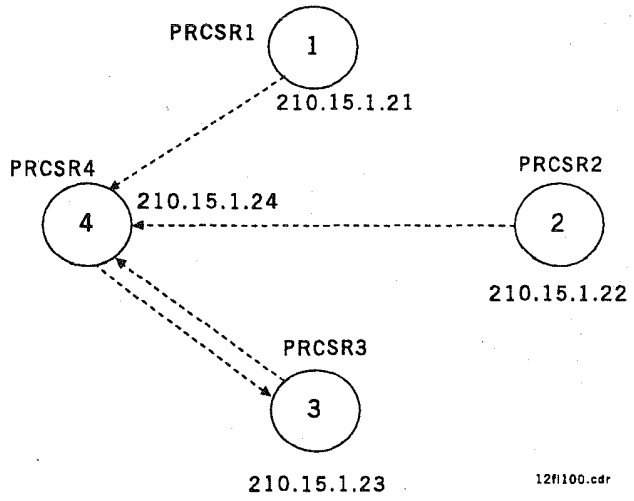


Figure 2–3. Relationship Between RIP Neighbors

```
PRCSR1    PRCSR     ...
PRCSR2    PRCSR     ...
PRCSR3    PRCSR     ...,NA=3,...
PRCSR4    PRCSR     ...,NA=4,...
    .
    .
    .
IPADR1    IPADR     PRCSR=PRCSR1,IPADDR1=(210,15,1,21,LOCAL),...
IPADR2    IPADR     PRCSR=PRCSR1,IPADDR1=(210,15,1,24,RIPNBR),NA=4
*
IPADR3    IPADR     PRCSR=PRCSR2,IPADDR1=(210,15,1,22,LOCAL),...
IPADR4    IPADR     PRCSR=PRCSR2,IPADDR1=(210,15,1,24,RIPNBR),NA=4
*
IPADR5    IPADR     PRCSR=PRCSR3,IPADDR1=(210,15,1,23,LOCAL),...
IPADR6    IPADR     PRCSR=PRCSR3,IPADDR1=(210,15,1,24,RIPNBR),NA=4
*
IPADR7    IPADR     PRCSR=PRCSR4,IPADDR1=(210,15,1,24,LOCAL),...
IPADR8    IPADR     PRCSR=PRCSR4,IPADDR1=(210,15,1,23,RIPNBR),NA=3
```

Example 2

This method for configuring all other TCP/IP DNS nodes assumes the DNS SUBNET statement specifies RIP=NO, and that *n* nodes require *n-1* IPADR statements that specify the NA parameter for IP address pairing.

This method also assumes that all routes across the DNS TCP-IP subnetwork are configured.

```
PRCSR1    PRCSR     ...,NA=101,...
PRCSR2    PRCSR     ...,NA=102,...
PRCSR3    PRCSR     ...,NA=103,...
PRCSR4    PRCSR     ...,NA=104,...
    .
    .
    .
IPADR11    IPADR     PRCSR=PRCSR1,IPADDR1=(210,15,1,21,LOCAL),...
IPADR12    IPADR     PRCSR=PRCSR1,IPADDR1=(210,15,1,22),NA=102
IPADR13    IPADR     PRCSR=PRCSR1,IPADDR1=(210,15,1,23),NA=103
IPADR14    IPADR     PRCSR=PRCSR1,IPADDR1=(210,15,1,24),NA=104
*
IPADR21    IPADR     PRCSR=PRCSR2,IPADDR1=(210,15,1,22,LOCAL),...
IPADR22    IPADR     PRCSR=PRCSR2,IPADDR1=(210,15,1,21),NA=101
IPADR23    IPADR     PRCSR=PRCSR2,IPADDR1=(210,15,1,23),NA=102
IPADR24    IPADR     PRCSR=PRCSR2,IPADDR1=(210,15,1,24),NA=104
*
IPADR31    IPADR     PRCSR=PRCSR3,IPADDR1=(210,15,1,23,LOCAL),...
IPADR34    IPADR     PRCSR=PRCSR3,IPADDR1=(210,15,1,21),NA=101
IPADR34    IPADR     PRCSR=PRCSR3,IPADDR1=(210,15,1,22),NA=102
IPADR34    IPADR     PRCSR=PRCSR3,IPADDR1=(210,15,1,24),NA=104
*
IPADR41    IPADR     PRCSR=PRCSR4,IPADDR1=(210,15,1,24,LOCAL),...
IPADR42    IPADR     PRCSR=PRCSR4,IPADDR1=(210,15,1,21),NA=101
IPADR43    IPADR     PRCSR=PRCSR4,IPADDR1=(210,15,1,22),NA=102
IPADR44    IPADR     PRCSR=PRCSR4,IPADDR1=(210,15,1,23),NA=103
```

2.8. Configuring Network Bridge Nodes

A bridge node lets you use DCA communications protocols across a TCP/IP network to interconnect parts of a Telcon network. It also lets you connect a Telcon network to a TCP/IP network, providing access to hosts running TCP/IP applications in the Telcon network. You can configure the following connections:

- Connecting trunks across TCP/IP networks
- Connecting to DCA across TCP/IP networks
- Connecting to hosts running TCP/IP (DDN 1100) applications in a Telcon network
- Configuring DCPs as TELNET terminal concentrators

2.8.1. Connecting Trunks Across TCP/IP Networks

You can configure a Telcon trunk to use a TCP/IP network as the transmission medium for DCA communications.

> **Caution**
>
> NEVER bridge a DNS subnetwork with a TCP-IP Trunk. This can cause circular routing and numerous other problems.

2.8.1.1. Connecting Trunks over TCP/IP Networks using only IP

This method allows either RTC or DNS trunks to be bridged across a TCP/IP network. You must configure a unique trunk to bridge the TCP/IP network, using only the network transmission medium. You cannot configure the trunk to include other transmission media, such as UDLC lines. The trunk itself is considered a DCA endpoint (DCAEP) for this type of configuration.

Format

```
name        IPADR       ...,DCAEP=trunk,...
```

There are two ends to the trunk. Each end must be assigned a unique IP address. The side associated with the PRCSR with which the IPADR configuration statement is associated must be flagged LOCAL. The other side must not be flagged LOCAL.

Format

```
name        IPADR       ...,IPADDR1=(nn,nn,nn,nn,LOCAL),;
                        IPADDR2=(mm,mm,mm,mm),...
```

or

```
name        IPADR       ...,IPADDR1=(nn,nn,nn,nn),;
                        IPADDR2=(mm,mm,mm,mm,LOCAL),...
```

Examples

```
PRCSR1      PRCSR       ...
PRCSR2      PRCSR       ...
    .
    .
    .
TRUNK1      TRUNK       PRCSR=(PRCSR1,PRCSR2),...
                        ...
IPADR1      IPADR       PRCSR=PRCSR1,DCAEP=TRUNK1,;
                        IPADDR1=(57,23,0,54,LOCAL),;
                        IPADDR2=(57,31,0,117)
IPADR2      IPADR       PRCSR=PRCSR2,DCAEP=TRUNK1,;
                        IPADDR1=(57,23,0,54),;
                        IPADDR2=(57,31,0,117,LOCAL)
```

An alternative method configures the two processors separately:

Configuration 1

```
PRCSR1      PRCSR       ...
    .
    .
    .
TRUNK1      TRUNK       PRCSR=PRCSR1,...
    .
    .
    .
IPADR1      IPADR       PRCSR=PRCSR1,DCAEP=TRUNK1,;
                        IPADDR1=(57,23,0,54,LOCAL),;
                        IPADDR2=(57,31,0,117)
```

Configuration 2

```
PRCSR2      PRCSR       ...
    .
    .
    .
TRUNK1      TRUNK       PRCSR=PRCSR2,...
    .
    .
    .
IPADR1      IPADR       PRCSR=PRCSR2,DCAEP=TRUNK1,;
                        IPADDR1=(57,23,0,54),;
                        IPADDR2=(57,31,0,117,LOCAL)
```

2.8.1.2. Connecting Trunks Across TCP/IP Networks Using TCP and IP

This method allows DNS trunks to be bridged across a TCP/IP network. You must configure a unique trunk to bridge the TCP/IP network, using only the network transmission medium. You cannot configure the trunk to include other transmission media, such as UDLC lines. The trunk itself is considered a DCA endpoint (DCAEP) for this type of configuration.

Use this method to bridge trunks over networks that can lose a lot of data (IP routers connected to X.25 networks are one example). Also use this method to bridge trunks that connect IS-5000/6000 or IS-PC end systems.

Format

```
name    IPADR       ...,DCAEP=TRUNK,...
```

There are two ends to the trunk. Each end must be assigned a unique IP address. The side belonging to the PRCSR parameter with which the IPADR configuration statement is associated, must be flagged LOCAL. The other side must **not** be flagged LOCAL. Both sides must have the TCP Port (265) specified.

Format

```
name    IPADR   ...,IPADDR1=(nn,nn,nn,nn,LOCAL,265),;
                IPADDR2=(nn,nn,nn,nn,,265),...
```

or

```
name    IPADR   ...,IPADDR1=(nn,nn,nn,nn,265),;
                IPADDR2=(nn,nn,nn,nn,LOCAL,265)
```

Examples

```
PRCSR1      PRCSR       ...
PRCSR2      PRCSR       ...
TRUNK1      TRUNK       PRCSR=(PRCSR1,PRCSR2),...
    .
    .
    .
IPADR1      IPADR       PRCSR=PRCSR1,DCAEP=TRUNK1,;
                        IPADDR1=(57,23,0,56,LOCAL,265),;
                        IPADDR2=(57,31,0,117,265)
IPADR2      IPADR       PRCSR=PRCSR2,DCAEP=TRUNK1,;
                        IPADDR1=(57,23,0,56,,265),;
                        IPADDR2=(57,31,0,117,LOCAL,265)
```

The alternative method would configure the two processors separately:

Configuration 1

```
PRCSR1     PRCSR      ...
    .
    .
    .
TRUNK1     TRUNK      PRCSR=PRCSR1,...
    .
    .
    .
IPADR1     IPADR      PRCSR=PRCSR1,DCAEP=TRUNK1,;
                      IPADDR1=(57,23,0,56,LOCAL,265),;
                      IPADDR2=(57,31,0,117,265)
```

Configuration 2

```
PRCSR1     PRCSR      ...
    .
    .
    .
TRUNK1     TRUNK      PRCSR=PRCSR2
    .
    .
    .
IPADR1     IPADR      PRCSR=PRCSR2,DCAEP=TRUNK1,;
                      IPADDR1=(57,23,0,56,,265),;
                      IPADDR2=(57,31,0,117,LOCAL,265)
```

2.8.2. Connecting to DCA Across TCP/IP Networks

A DCA termination system can reside in any hardware executing the appropriate software. Another DCP running Telcon, a PC running IS-PC, or a UNIX system running IS-5000/6000 are DCA termination systems. When the DCA software is configured to use TCP/IP communications, it can connect to the DCP over a TCP/IP network as a DCA termination system.

Any DCA termination system can connect to the DCP without using TCP/IP, as long as the transmission medium connects to the DCP directly. When a DCA termination system must communicate across the DDN, or any other series of networks connected by TCP/IP routers, it must use TCP/IP to complete the connection.

The configuration statements required to connect to a DCA termination system include the DCATS statement which identifies the termination system, and the SESSN statements that assign network sessions connecting the termination system to other points in the Telcon network. You must match the network session numbers defined in the Telcon configuration to the network session numbers defined in the DCA termination system's software.

2.8.2.1. Defining the DCA Endpoint

The DCA endpoint for bridging to a DCA termination system is the DCATS statement that identifies the DCA termination system.

Format

```
name        IPADR       ...,DCAEP=dcats,...
```

2.8.2.2. Assigning IP Addresses

There are two ends to the connection. One end is associated with the DCA termination system to which the connection is being established. The other end is associated with the DCP running the TCP-IP Stack software. This address must be flagged LOCAL.

Note: *The local address is usually identical to the address originally specified for the DCP.*

Format

```
name        IPADR       ...,IPADDR1=(nn,nn,nn,nn,LOCAL),;
                        IPADDR2=(mm,mm,mm,mm),...
```

or

```
name        IPADR       ...,IPADDR1=(nn,nn,nn,nn),;
                        ...,IPADDR2=(mm,mm,mm,mm,LOCAL),...
```

Example

```
PRCSR1      PRCSR       ...
   .
   .
   .
DCATS1      DCATS       PRCSR=PRCSR1,...
   .
   .
   .
IPADR1      IPADR       PRCSR=PRCSR1,DCAEP=DCATS1,;
                        IPADDR1=(193,17,242,3,LOCAL),;
                        IPADDR2=(193,32,11,61)
```

2.8.2.3. Changing TCP Port Numbers

Port numbers are used by TCP to separate communications between two TCP/IP hosts into various protocols. The default value used for bridging DCA termination systems across TCP/IP is 264.

Other protocols communicating with either the DCP or the DCA termination system can already be using port number 264. In that case, you must choose a new port number.

Format

```
name        IPADR       ...,IPADDR1=(nn,nn,nn,nn,LOCAL,port1),;
                        IPADDR2=(mm,mm,mm,mm,,port2),...
```

or

```
name        IPADR       ...,IPADDR1=(nn,nn,nn,nn,,port1),;
                        IPADDR2=(mm,mm,mm,mm,LOCAL,port2),...
```

Your network administrator must know the various protocols being run. TCP-IP Stack uses various non-standard port numbers, as do protocols running with the IS software on PC or UNIX systems.

Standard assigned port numbers are listed in RFC 1060 (as of March 1990) or an RFC that can have superseded RFC 1060. When changing port numbers, your network administrator must be aware of all port numbers being used by the TCP/IP hosts.

Example

```
PRCSR1      PRCSR       ...
   .
   .
   .
DCATS1      DCATS       PRCSR=PRCSR1,...
   .
   .
   .
IPADR1      IPADR       PRCSR=PRCSR1,DCAEP=DCATS1,;
                        IPADDR1=(23,45,117,43,LOCAL,150),;
                        IPADDR2=(63,255,34,41,,150)
```

2.8.3. Connecting to Hosts Running TCP/IP (DDN 1100) Applications in a Telcon Network

TCP-IP Stack can receive messages from any host, directed at any other host. TCP-IP Stack determines which messages are passed to applications by comparing the destination address of incoming datagrams with a list of local host addresses. The list is built using the IPADR statement.

Local hosts include the DCP being configured to run TCP-IP Stack, or any OS 1100 hosts running DDN 1100. Each local host must be assigned a unique IP address.

2.8.3.1. Defining the DCA Endpoint

The DCAEP parameter specifies the name of a network definition statement that identifies the Telcon node (DCA endpoint) to be associated with a given IP address. One of several configuration statements can be used to establish the association within a Telcon network.

- A DCPTS statement identifies the DCP being configured to run TCP-IP Stack itself. Specify a DCPTS name if terminals attached to the DCP use User TELNET to connect to other TCP/IP hosts.
- A DCATS statement identifies a DCA termination system connected to the DCP. Specify a DCATS name if a channel attached host or a remote host in the Telcon network will use Server TELNET to connect to other TCP/IP hosts.
- An XEU statement identifies an application residing on a DCA termination system or a DNS node somewhere in the DCA network. Specify an XEU name if there is a DDN 1100 application running on a host that is not connected directly to this DCP.
- A NETADR statement identifies any DNS node. It can be used in any of the above IPADDR statements.

2.8.3.2. Assigning IP Addresses

Format

```
name        IPADR       ...,IPADDR1=(nn,nn,nn,nn,LOCAL),;
                        ...,DCAEP=dca-endpoint
```

Either parameter IPADDR1 or IPADDR2 can be used; there is no preference. The first four sub-parameters specify the four bytes of the IP address, one byte per parameter. The last sub-parameter is the flag LOCAL, which marks the address as the IP address of the local host.

Example

This example configures two IP addresses for the DCP. Data directed to the IP address specified on the first IPADR statement is passed to the Telcon network node identified by the NETADR statement. Data directed to the other IP address is passed to the Telcon network node where XEU1 is located.

When data is received by TCP-IP Stack from either of the two network nodes, the respective IP address is used as the source when passing the data to another TCP/IP host.

```
PRCSR1      PRCSR       ...
NETADR1     NETADR      ...
XEU1        XEU         ...
   .
   .
   .
IPADR1      IPADR       PRCSR=PRCSR1,DCAEP=NETADR1,;
                        IPADDR1=(23,45,0,12,LOCAL)
IPADR2      IPADR       PRCSR=PRCSR1,DCAEP=XEU1,;
                        IPADDR1=(23,67,0,12,LOCAL)
```

2.8.4. Configuring DCPs as TELNET Terminal Concentrators

The User TELNET application in TCP-IP Stack converts data received from a TCP/IP host to a format appropriate to the terminal that made the connection. For communications with TCP/IP hosts across a TCP/IP network, User TELNET interprets all messages as using Network Virtual Terminal (NVT) protocol. For communications with TCP/IP applications on DCA hosts in a Telcon network, User TELNET can interpret a combination of NVT and DCA terminal protocol elements.

You can configure User TELNET by adding an XEU statement to your configuration that defines the path to TELNET. The value for the DESTSSU parameter must be the name of the TCP–IP Stack EU statement.

Example

```
USRTEL    XEU       TS=DCPTS1,DESTTSU='DTPX',DESTSSU='EUTCP'
```

2.9. Associating Host Names with IP Addresses

TCP–IP Stack supports the Domain Name System, an official Internet protocol specified in RFC 1034 and RFC 1035. TCP–IP Stack implements the resolver portion of that protocol. This enhancement allows you to specify a destination with a name rather than an Internet address. This name is used when making TELNET connections, specifying the name on the CONNECT command instead of specifying the IP address.

You can associate names (like HOST1) with Internet addresses in the TCP–IP Stack configuration or TCP–IP Stack can query a configured Domain Name System Server for the address. TCP–IP Stack keeps configured names and addresses permanently in a cache file. Addresses obtained from the Domain Name System Server are timed and deleted when the time expires.

Note: *If you do not want to assign names to TCP/IP hosts, or if you know that the TELNET user side of DCP TCP–IP Stack will never be used, the information in this subsection is not applicable.*

To associate names and addresses, use the following procedure.

1. On the NSS statement, identify the cache files to hold the host names and Internet addresses with the parameter HSTNAMES.

   Example

```
TCPEU     EU        PRCSR=PRC1,TYPE=TCPIP,HSTNAMES='TCP*CACHE.'
```

2. Identify the Domain Name System Server with the SERVER value on the IPADDR1 parameter of the IPADR statement. The purpose of the server is to supply TCP–IP Stack with Internet addresses for unconfigured host names. You can configure as many servers as you need.

   Example

```
SERVER    IPADR     PRCSR=PRC1,IPADDR1=(192,60,224,1,SERVER),;
                    NAME1='DALLAS.HOST.UNIX.COM'
```

3. Associate names and IP addresses using the IPADDR and NAME parameters on the IPADR statement. These names and addresses are kept permanently in the cache file. This example configures four host names and Internet addresses.

   Example

```
IPADR1    IPADR     PRCSR=PRC1,IPADDR1=(96,2,0,143),;
                    NAME1='DALLAS1'
IPADR2    IPADR     PRCSR=PRC1,IPADDR2=(193,43,254,250),;
                    NAME2='AUSTIN1'
IPADR3    IPADR     PRCSR=PRC1,IPADDR1=(12,147,32,1),;
                    NAME1='SLC1',;
                    IPADDR2=(12,147,2,2),;
                    NAME2='SLC2'
```

2.10. Configuring DCA Sessions over TCP/IP

With TCP–IP Stack Release 2R2B, you can configure DCA sessions to use the TCP/IP transport. This means that terminals connected to a TCP/IP network can access DCA applications on an OS 2200 or IS 6000 host system. Applications on DCA hosts can also communicate across a TCP/IP network.

To do this, use the following procedure:

1. Using the ADDRESS statement, assign IP addresses to each DCA system that will use the TCP/IP transport.

   Example

   ```
   HOST1      ADDRESS      IPADR=192.60.223.28
   ```

2. Use the name of the ADDRESS statement as the value for the DS parameter in the XEU statements defining DCA applications.

   Example

   This example configures access to demand on an OS 1100 host system over TCP/IP.

   ```
   DEMH1      XEU          ADDRESS=HOST1,DESTTSU='RSDCSU'
   ```

Note: *This is not the same as connecting DCA termination systems over TCP/IP or running DCA trunks over TCP/IP.*

2.11. Configuring the TN3270 Emulator

You can use a TN3270 emulator on a personal computer to access applications on OS 2200 and SNA host systems through the SNA/net program product on the DCP. You must have SNA/net installed on at least one DCP in the network, not necessarily the DCP to which the TN3270 is connected. The TN3270 must be connected to a DCP running TCP–IP Stack.

In the TN3270 emulator, you must configure the IP address of the DCP running TCP–IP Stack.

Example

This example configures the TN3270 handler. PLU257 is the name of a POOL statement in an SNA/net configuration. To see this example in the context of a complete SNA/net configuration, refer to the *SNA/net Configuration Guide* (7831 5629). In this example, both TCP–IP Stack and SNA/net are running on the same DCP.

```
PRC1        PRCSR       NA=11,DSPL=ALL,;
                        LOGL=ALL
TCPDTPX     EU          PRCSR=PRC1,TYPE=TCPIP
TN3270A     XEU         DS=PRC1,DESTSSU='PLG257',DESTTSU='SNACONN',;
                        DPP=TN3270
```

Section 3
TCP-IP Stack Configuration Statements

This section describes the following configuration statements:

Statement	Description
EU	A Telcon statement modified to define TCP-IP Stack as an end user program.
IPADR	A TCP-IP Stack statement that assigns internet protocol (IP) addresses to DCA end points in Telcon or TCP/IP networks.
NSM	A TCP-IP Stack statement that specifies entries in the host name directory.
NSS	A TCP-IP Stack statement that defines the characteristics of the host name table.
SUBNET	A TCP-IP Stack statement that defines the TCP/IP subnetworks used by the DCP.
XEU	A Telcon statement that configures DCA sessions over TCP/IP and identifies the handler for the TN3270 emulator.

Note: *The information on the EU and XEU statements provided in this section describes parameters that are pertinent to TCP-IP Stack configurations. Refer to the Telcon Configuration Reference Manual (7831 5686) for a complete description of these statements.*

Configuration Statements Syntax

Configuration statements consist of a name field, an operation field, and a parameter field. The name and operation fields are always required for configuration statements; the parameter field is usually required. Defaults are in place wherever possible for parameters that are used often.

The syntax of a configuration statement is as follows:

```
name        operation   parameter,parameter,...
```

Parameters

name

Description

The name field begins in column 1. A name must start with an alphabetic character and can contain additional alphabetic (A through Z) or numeric (0 through 9) characters. Names can be referenced in subsequent statements for identification purposes; therefore, names should be meaningful for clarity in other statements.

Certain names are reserved. Do not specify them in the name field of a statement. The configuration processor flags any attempt to use a reserved word with the diagnostic message:

```
<reserved word> MAY NOT BE REDEFINED
```

Words reserved for standard Telcon are listed in the *Telcon Configuration Reference Manual* (7831 5686).

operation

Description

The operation field, separated from the name field by one or more blanks, specifies the kind of configuration statement. The operation field specifies a communications facility such as a communications line or communications processor.

Code the operation field as an alphabetic character string that exactly matches one of the configuration statement types described in the following subsections.

parameter

Description

Separate parameter fields, if present, from the operation field by one or more blanks. The parameter fields consist of keywords and their associated values. Keywords are shown in UPPERCASE font in this section and should be entered exactly as shown. Keywords are separated from values with an equals sign (=). Separate multiple parameters with commas and with no intervening blanks.

EU – Defining End-User Programs

The EU statement is a standard Telcon statement that has been modified for the TCP-IP Stack to define the following:

- The TCP-IP Stack as an end user on a DCP
- The maximum number of times TCP attempts to transmit a message
- A TELNET sentinel character
- A time-to-live value for IP datagrams
- The TCP keep-alive mechanism
- The size of the IP routing table a DCP uses
- The name of the cache file for host names and IP addresses

Each Telcon running the TCP-IP Stack requires one EU statement. Two formats for the EU statement are shown here. The parameters differ if you are pairing TCP–IP Stack with Telcon 9R2 or Telcon 9R3.

Format 1. Use with Telcon 9R2

```
name      EU      TYPE=TCPIP,;
                  [HSTNAMES=filename],;
                  [KEEPALIV={NO | YES}],;
                  [MAXTRY=n],;
                  [NPPORT=mxprt],;
                  [ROUTSIZE=entries],;
                  [SENTINEL='char'],;
                  [TCPTIME=time3],;
                  [TIMOUT=time1],;
                  [TMTOLIV=time2]
```

Format 2. Use with Telcon 9R3

```
name      EU      TYPE=TCPIP,;
                  [KEEPALIV={NO | YES}],;
                  [MAXTRY=n],;
                  [NPPORT=mxprt],;
                  [ROUTSIZE=entries],;
                  [SENTINEL='char'],;
                  [TCPTIME=time3],;
                  [TIMOUT=time1],;
                  [TMTOLIV=time2]
```

Required Parameters:

TYPE=TCPIP

Description
Identifies the TCP-IP Stack as an end user.

Optional Parameters

HSTNAMES=*filename*

Range
0 to 37 characters

Default
No file name

Description
The name of the cache file to contain host names and their associated IP addresses. Use this parameter when you want to use the Domain Name Server with Telcon 9R2. With Telcon 9R3, you configure the file name on the *NSS* statement.

KEEPALIV=NO | YES

Default
NO

Description
Specifies the TCP keep-alive mechanism. When you specify KEEPALIV=YES, TCP periodically checks the other end of the connection for activity. If the other end does not respond within about six minutes, TCP ends the session. This capability is useful when a user ends a session with an OS 1100 application without following the proper procedures, such as simply turning the terminal off. Without keep-alive configured, the application can not know the session has ended. Consequently, it can not let the user sign on again until the OS 1100 or the DCP is rebooted.

Configuring keep-alive can cause problems, however. If, for example, a file transfer is interrupted when an IP router is temporarily disabled, the router must come back into service before the keep-alive mechanism expires, or the file transfer must start over from the beginning.

MAXTRY=*n*

Range
1 to 65,535

Default
7

Description
Specifies the maximum number of times TCP attempts to transmit a message.

NPPORT=*mxprt*

Range
1 to 1,350

Default
300

Description
Specifies the number of TCP port identifiers reserved for passive ports.

ROUTSIZE=*entries*

Default
0

Description
Specifies the maximum number of routing table entries. A value of zero indicates that the table has no size limit. TCP-IP Stack discards received routes that are in excess of the specified table size.

SENTINEL='*char*'

Range
Any symbolic ASCII character, except a question mark (?).

Default
%

Description
Specifies the Telcon sentinel character. TELNET users use the Telcon or TOMFsentinel to inform the TCP-IP Stack that what follows is a TELNET command, not data. Although users type a two- character sequence, the single character here is used twice at the beginning of a TELNET command. Choose a character that is not likely to appear in data. This character cannot be a question mark (?), and it must be different than the Telcon sentinel character.

TCPTIME=*time3*

Range
1 to 120 seconds

Default
120 seconds

Description
The TCP disconnect quiet time. It specifies the wait time before a connection is completely closed after the peer TCP processes a connection close sequence.

TIMOUT=*time1*

Range
1 to 250 seconds

Default
120 seconds

Description
The initial TCP timer value. It specifies the wait time in seconds between retransmissions. This value is automatically recomputed by TCP during the lifetime of a connection.

TMTOLIV=*time2*

Range
1 to 255 seconds

Default
255 seconds

Description
The time-to-live value for IP datagrams, in gateway hops. This value is carried in the datagram and is decremented by one second each time the datagram passes through a gateway. If the value reaches zero, the datagram is invalidated and is discarded. The time-to-live mechanism prevents datagrams from continually passing through the network.

Example

```
EUITCP    EU        PRCSR=PRC1,TYPE=TCPIP,MAXTRY=5,;
                    TMTOLIV=40,SENTINEL=':'
```

The example defines the following:

- TCP-IP Stack as an end user
- A retransmission count of 5
- Time-to-live value of 40 gateway hops
- A colon as the TELNET sentinel character

Online Configuration Differences

None.

IPADR – Assigning IP Addresses

The IPADR statement was created especially for TCP-IP Stack. It performs the following functions:

- Defines for the TCP-IP Stack the internet addresses for the following:
  - DCA end points within a Telcon network, enabling TCP-IP Stack to function as a network bridge by routing communications between that end point and other hosts and terminals located in a TCP/IP network
  - TCP/IP devices across a public data network (PDN)
- Enables TCP port number changes
- Defines a DCP's RIP neighbor, enabling RIP functionality within the DCA network
- Associates the DTE address of a TCP/IP device with its IP address
- Associates IP addresses and host names

Each TCP-IP Stack configuration must include at least one IPADR statement and can include many, depending on the communications you want enabled.

Format

```
name    IPADR   PRCSR=prcsr,[DCAEP=dcaname],;
                IPADDR1=(adr1,adr2,adr3,adr4[,[flag][,port]]),;
                [IPADDR2=(adr1,adr2,adr3,adr4[,[flag][,port]],;)
                [{ DTEADR='dteaddress'|                        }  ]
                [{ NA={nn|(sic,nn)|(sc,sic,nn)|(sd,sc,sic,nn)} ,;]
                [NAME1=name][,NAME2=name]
```

Parameters

PRCSR=*prcsr*

Description
Specifies the PRCSR statement that defines a DCP running the TCP-IP Stack.

DCAEP=*dcaname*

Description
Specifies the name of the DCATS, DCPTS, TRUNK, XEU, or NETADR statement that defines the DCA end point associated with this IPADR statement.

To define the relationship for TS/TN configurations, reference a DCATS, DCPTS, TRUNK, or XEU statement. For DNS configurations, reference a TRUNK, XEU, or NETADR statement. This parameter is required when either IPADDR parameter is flagged LOCAL.

If both IPADDR1 and IPADDR2 are specified, you are configuring a DCA enpoint acrosss a TCP/IP network, and one of the addresses must be flagged LOCAL. In this case, the value for the DCAEP parameter can be the name of either a TRUNK or DCATS statement. A TRUNK or DCATS statement that is used as the value of the DCAEP parameter cannot also be referenced by a STATION or SAP statement.

IPADDR*n*=(*adr1,adr2,adr3,adr4* [,[flag][,*port*]])

Default

264 for *port* (DCATS-to-DCATS pairing only)

Range

1 to 65,535 for *port*

Description

IPADDR1 and IPADDR2 associate IP addresses and DCA end points. They also associate IP addresses and DTE addresses for TCP/IP hosts attached to PDNs. In most instances, use either IPADDR1 or IPADDR2, not both. You can use them interchangeably, however. Use both only when you define a DCA endpoint across a TCP/IP network. Specify the IP address segments as decimal numbers ranging in value from 0 to 255.

flag Identifies the kind of address specified with the *addrn* parameter. You may use one of the following flags:

LOCAL Identifies the IP address of the local DCP or another DCA end point within the Telcon network. The LOCAL flag refers to the DCP referenced on the PRCSR parameter and to the DCA facility referenced on the DCAEP parameter. If the DCA end point resides in this DCP, the IP address specified on the IPADDR*n* parameter is assigned to this end point and this DCP.

If the DCA facility defined on the DCAEP parameter is not located on this DCP, the LOCAL label assigns the IP address to the DCP identified on the PRCSR parameter and associates the DCA facility with the IP address.

If the DCA end point is a DCP trunk, with one DCP located across the TCP/IP network, two IPADR statements are required to assign an IP address to each end of the trunk. On one statement, one end is flagged LOCAL and on the other statement, the other end is flagged LOCAL. See Example 3 at the end of the IPADR statement discussion for information on configuring trunks.

RIPNBR Identifies the IP address of an RIP neighbor node that uses routing information protocol. An RIP neighbor address cannot be flagged LOCAL. This parameter is used only to define RIP neighbors on DCA networks.

SERVER Identifies the address of the Domain Name System name server.

port Port number is set to 265 to run ES-IS trunks over TCP-IP stack. In all other cases it is best to leave this parameter unspecified.

DTEADR=*dteaddress*

Description

The DTE address associated with the remote IP address specified on this statement. It applies to the IPADDR that is not flagged LOCAL. This parameter is required for generic X.25 networks that do not support the IP-to-X.121 address translation algorithm used in the DDN. The format is a quoted string of up to 15 decimal digits. This parameter should be used only when an associated SUBNET statement specifies TYPE=IPPDN.

NA={ *nn* | (*sic,nn*) | (*sc,sic,nn*) | (*sd,sc,sic,nn*) }

Description

The DNS node address of a DCA end point across a TCP/IP network. Use this parameter only for TCP-IP hosts that are not flagged LOCAL. This parameter is required for DNS subnetworks to enable the TCP-IP Stack to discover other active TCP-IP nodes in the DNS subnetwork. The address format conforms to DNS addressing conventions and is identical to the NA parameter used on the NETADR statement.

NAME1=*name*

Description

Specifies a name for the host defined by the IPADDR1 parameter. This name becomes an entry in the local host name directory. If you use this parameter, you do not need to use the NSM configuration statements.

NAME2=*name*

Description

Specifies a name for the host defined by the IPADDR2 parameter. This name becomes an entry in the local host name directory. If you use this parameter, you do not need to use the NSM configuration statements.

Example 1

This example shows an IPADR statement defining a local host, that is, a host in the DCA network. The host could be an OS 1100 running DDN 1100, a DCP functioning as a TELNET terminal concentrator, or some other DCA system running TCP/IP software.

```
IPA1      IPADR      PRCSR=PRC1,DCAEP=H1100,IPADDR1=(36,143,0,95,LOCAL)
```

Example 2

This example defines a U Series or Personal Workstation[2] host running Information Services (IS) software located across the TCP/IP network.

- The DCAEP parameter specifies a DCATS statement that uniquely identifies the remote host. This DCATS statement cannot be referenced by a STATION or SAP statement.
- IPADDR1 is flagged LOCAL and specifies the IP address of the local DCP.
- IPADDR2 specifies the IP address of the remote host.

```
IPA2      IPADR      PRCSR=PRC1,DCAEP=REMOTE1,;
                     IPADDR1=(36,143,0,100,LOCAL),;
                     IPADDR2=(36,143,0,90)
```

Example 3

This example defines a trunk connection to U Series or PW[2]/IS End System (ES trunk).

- The DCAEP parameter specifies a TRUNK statement defining the DNS trunk to the U Series or PW[2] end system. This TRUNK cannot be referenced by a STATION or a SAP statement.
- IPADDR1 is flagged LOCAL and specifies the IP address of the DCP.
- IPADDR2 is the IP address for the U Series or PW[2] host.
- The port number for an ES-IS trunk, which connects a DCP to an IS system, must be specified as 265 for both ends of the trunk.

```
IPA3      IPADR      PRCSR=PRC1,DCAEP=TRUNKIS,;
                     IPADDR1=(129,221,2,91,LOCAL,265),;
                     IPADDR2=(129,221,2,112,,265)
```

Example 4

Example 4 defines a DNS or RTC trunk between the local DCP and a remote DCP across the TCP/IP network. Each IPADR statement refers to the same TRUNK statement, and each defines one end of the trunk.

```
PRCSR1    PRCSR      ...
PRCSR2    PRCSR      ...
   .
   .
   .
TRUNK1    TRUNK      PRCSR=(PRCSR1,PRCSR2),...
   .
   .
   .
IPADR1    IPADR      PRCSR=PRCSR1,DCAEP=TRUNK1,;
                     IPADDR1=(57,23,0,54,LOCAL),;
                     IPADDR2=(57,31,0,117)
IPADR2    IPADR      PRCSR=PRCSR2,DCAEP=TRUNK1,;
                     IPADDR1=(57,23,0,54),;
                     IPADDR2=(57,31,0,117,LOCAL)
```

Example 5

Example 5 illustrates how to specify a DTE address for a TCP/IP host connected to a generic X.25 PDN.

```
IPA4      IPADR      PRCSR=PRCSR1,;
                     IPADDR1=(77,13,1,44),;
                     DTEADR='31108010009672'
```

Example 6

Example 6 defines the DCP running TCP-IP Stack as a terminal concentrator (to support TELNET terminal access to TCP/IP destinations). It specifies the following:

- A DCPTS statement on the DCAEP parameter, which uniquely identifies the DCP
- One IPADDR parameter, flagged LOCAL, to assign an IP address to the DCPTS

```
IPA5      IPADR    PRCSR=PRCSR1,DCAEP=DCPTS1,IPADDR1=(36,143,0,100,LOCAL)
```

Online Configuration Differences

None.

NSM – Naming Entries in the Host Name Directory

The NSM statement specifies entries in the host name directory. This statement maps internet addresses to locally specified names.

Format

```
name      NSM       NSS=nss,;
                    NSID='name',;
                    IPADDR=(adr1,adr2,adr3,adr4)
```

NSS=*nss*

Description
Identifies the NSS statement with which this configuration statement is associated

NSID='*name*'

Description
Names the ASCII host. The name is 24 characters maximum and is of the format *label1.label2.....labeln*, enclosed in single quotation marks. Each label must begin with an alpha character and can consist of the characters A through Z and hyphen (-). Upper and lower case are the same.

IPADDR=(*adr1, adr2, adr3, adr4*)

Description
Specifies the internet host addresses associated with the specified host name

Example

The example specifies a parent NSS statement (NSPRCR1), a name for the remote host (REMOTEHOSTA), and an IP address.

```
RNAME     NSM       NSS=NSPRC1,NSID='REMOTEHOSTA',;
                    IPADDR=(168,2,25,3)
```

Online Configuration Differences

None.

NSS – Defining the Characteristics of the Host Name Directory

The NSS statement defines the characteristics of the domain name system resolver. This provides a directory for name-to-address mapping of host names and host Internet addresses.

Two formats for the NSS statement are shown here. The parameters differ if you are pairing TCP–IP Stack with Telcon 9R2 or Telcon 9R3.

Format 1. Use with Telcon 9R2

```
name      NSS       PRCSR=prcsr,;
                    [CCT=cnt1],;
                    [LIV=t],;
                    [NADR=cnt3],;
                    [UCT=cnt2]
```

Format 2. Use with Telcon 9R3

```
name      NSS       PRCSR=prcsr,;
                    [CCT=cnt1],;
                    [HSTFSIZ=size],;
                    [HSTNAM1='filename1'],;
                    [HSTNAM2='filename2'],;
                    [LIV=t],;
                    [NADR=cnt3],;
                    [TIMEOUT=timeout],;
                    [UCT=cnt2]
```

Parameters

PRCSR=*prcsr*

Description
Identifies the PRCSR for which this directory is being specified.

CCT=*cnt1*

Default
25

Range
0 to 255

Description
Identifies the number of entries to reserve in the core cache table.

HSTFSIZ=*size*

Default
100 blocks

Range
4 to 1000 blocks

Description
The size of each of the two files that the Domain Name System uses to store host names and their associated addresses.

HSTNAM1='*filename1*'

Default
DNS*HSTFILE1

Range
1 to 37 characters

Description
The name of one of the cache files to contain host names and their associated IP addresses. The Domain Name System uses two files. HSTNAM1 and HSTNAM2 cannot have the same name.

HSTNAME2='*filename2*'

Default
DNS*HSTFIL2

Range
1 to 37 characters

Description
The name of one of the cache files to contain host names and their associated IP addresses. The Domain Name System uses two files. HSTNAM1 and HSTNAM2 cannot have the same name.

LIV=*t*

Default
48

Range
0 to 65,535

Description
Identifies the time-to-live value in hours for the local cache entries.

NADR=*cnt3*

Default
3

Range
0 to 255

Description
Identifies the maximum number of host addresses to allow for each destination name.

TIMEOUT=*timeout*

Default
3 seconds

Range
1 to 100 seconds

Description
The amount of time the Domain Name System will wait for a response before retransmitting a request for an address.

UCT=*cnt2*

Default
25

Range
0 to 255

Description
Identifies the number of entries in the user control table.

Example

This example specifies PRC1 as the local processor for which the directory is built. In most cases, the defaults provide sufficient functionality and it is unnecessary to configure the other parameters.

```
NSPRC1    NSS         PRCSR=PRC1
```

Online Configuration Differences

None.

SUBNET – Defining TCP/IP Network Connections and Static Routes

The SUBNET statement defines TCP/IP subnetworks. Each statement defines a particular subnetwork, which is either directly attached to the DCP or reachable through a gateway.

- When used with the CHANNEL parameter, this statement allows IP routing over a host channel to TCP-IP/1100.
- When used with the PDNGRP parameter, this statement connects TCP-IP to an X.25 network.
- When used with the LINE parameter, this statement connects TCP-IP to LANs.
- When used with the IPGATEWY parameter, this statement defines fixed routes to other subnetworks.

Notes:

1. *TCP–IP Stack does not support logical subnetworking. Do not configure SUBNET statements with different IPNETID parameters and the same CHANNEL, LINE, PDNGRP, or IPGATEWY parameter.*
2. *TCP–IP Stack does not support network bridging. Do not configure multiple SUBNET statements with identical IPNETID parameters and different values for the TYPE parameter.*
3. *Multiple SUBNET statements with identical IPNETID and TYPE parameters are assumed to connect to the same physical network.*

Format

```
name     SUBNET    PRCSR=prcsr,;
                   IPNETID={(adr1,adr2,adr3,adr4) | DEFLTGWY},;
                   TYPE=subntype,;
                   {CHANNEL=channel[,CHANNEL=channel]... |
                   {LINE=line[,LINE=line]... |
                   {PDNGRP=pdngrp[,PDNGRP=pdngrp]... |
                   {IPGATEWY=(hopcnt,adr1,adr2,adr3,adr4),;}
                   [AUTONUM=n],;
                   [DGSIZE=bytes],;
                   [IPBRDCST={1|0}],;
                   [IPMAXVC=n],;
                   [IPROUTER={NO|YES}],;
                   [RIP=(activity,cost,timeout,count,delay,port)],;
                   [SECURITY=security],;
                   [SUBNMASK=(adr1,adr2,adr3,adr4)]
```

Parameters

PRCSR=*prcsr*

Description
Specifies a PRCSR statement name that defines the DCP attached to this subnetwork

IPNETID=(*adr1,adr2,adr3,adr4*) | DEFLTGWY

Description

adr1,adr2,adr3,adr4 Specifies a network number (NETID), which along with the host number (HOSTID), makes up a complete internet address. All internet addresses are 32-bit values specified as four decimal numbers. How you specify this value, however, depends on whether this SUBNET statement defines subnetwork routing. For networks that do not require this type of routing, follow the rules described in Table 3-1.

To define subnetwork routing, the rules described in Table 3-1 apply as well. However, you now use some of the bytes to which you assigned zeros for an extended network number. For example, class A networks still require the first byte – the NETID portion – to specify an address between 1 and 126. You can use part of the remaining three bytes to specify additional NETID digits. The parts you use for this purpose must be defined on the SUBNMASK parameter of this SUBNET statement.

DEFLTGWY Specifies that any networks or subnetworks not specifically configured can be reached through the default gateway defined by the IPGATEWY parameter. Use the DEFLTGWY label in place of an IP address specification.

Table 3–1. IPNETID Rules for Networks that Do Not Use Subnetworking

Net Type	1st Byte Value	2nd Byte Value	3rd Byte Value	4th Byte Value
A	1-126	0	0	0
reserved	127	any	any	any
B	128-191	1-255	0	0
C	192-223	1-255	1-255	0
reserved	224-255	any	any	any

Note: *Using reserved values will cause unpredictable results in the operation of TCP-IP Stack.*

Parameters

TYPE=*subntype*

Description

Specifies the subnetwork type, identified by one of the following labels:

DDN	specifies the DDN X.25 subnetwork.
IPCHAN	specifies a host channel (to an OS 1100 host) as an IP subnetwork.
IPDNS	specifies a Telcon DNS subnetwork. TCP-IP supports only one SUBNET of TYPE=IPDNS in any one Telcon node.
IPFDDI	specifies a fiber distributed data interface (FDDI).
IPPDN	specifies an X.25 PDN (generic X.25) subnetwork. If you specify IPPDN, you must also configure all required DTE addresses on associated IPADR statements.
IPTR	specifies a token ring (IEEE 802.5) subnetwork.
LAN	specifies a LAN subnetwork (for 802.3 MAC attachments only).
LANLLC	specifies a LAN subnetwork (for 802.2 LLC type 1 attachments only). This configures the subnetwork access protocol (SNAP) interface to LLC type 1.

Note: *The TYPE parameter is not required when the IPGATEWY parameter is used.*

CHANNEL=*channel*

Description

Specifies the name of a CHANNEL statement identifying the channel subnetwork you want to configure. The TYPE parameter must be set (up to eight channels can be configured on one SUBNET) to IPCHAN. The IPCHAN subnetwork type is valid only for OS 1100 hosts configured to use TCP/IP protocols over a channel connection to a DCP.

IPGATEWY=(*hopcnt,adr1,adr2,adr3,adr4*)

Range
0 to 255 for *hopcnt*

0 to 255 decimal, for each *addr*

Description
Specifies the hopcount and address of a gateway through which this subnetwork can be reached. This parameter implies that the subnetwork is not directly attached, but can be reached through the specified gateway. Use this parameter with the IPNETID parameter when defining a default gateway. Do not use the LINE, PDNGRP, or CHANNEL parameter if you use this parameter.

hopcnt	Specifies the number of hops from the gateway to the destination subnetwork. Use this parameter to prioritize the gateway when more than one gateway is configured.
adr1,...,adr4	The IP address of the gateway, formatted as four decimal values.

LINE=*line*

Description
Specifies the label of a LINE statement if the DCP is directly connected to the specified subnetwork. Use this parameter for LAN connections. You can specify this parameter up to eight times if the DCP has multiple connections to the LAN. This is mutually exclusive with PDNGRP, CHANNEL, or IPGATEWY.

PDNGRP=*pdngrp*

Description
Specifies the label of a PDNGRP statement identifying an attachment to either a generic X.25 or DDN X.25 subnetwork. You can specify up to eight additional PDNGRP parameters when multiple PDN groups are connected to the same subnetwork. This is mutually exclusive with LINE, CHANNEL, or IPGATEWY.

AUTONUM=*n*

Default
0

Description
Specifies the autonomous system number of the subnetwork, assigned by the network administrator.

DGSIZE=*bytes*

Default
576

Range
1 to 65,535

Description
Specifies the maximum number of bytes in an IP datagram on this subnetwork

IPBRDCST=1|0

Default
1

Description
Specifies the IP broadcast address format used on this subnetwork.

1 = 255.255.255.255
0 = 0.0.0.0

IPMAXVC=*n*

Description
Specifies the maximum number of virtual circuits IP can use for a given connection over an X.25 subnetwork

IPROUTER=YES | NO

Default
NO

Description
Specifies whether TCP-IP Stack operates as an IP router on this interface

RIP=(*activity,cost,timeout,count,delay,port*)

Note: *As of Release 2R3, RIP is supported for SUBNETs of TYPE=LAN, LANLLC, IPFDDI, and IPDNS.*

Default

NO for *activity*
1 for *cost*
30 for *timeout*
5 for *count*
300 for *delay*
520 for *port*

Range

1 to 15 for *cost*
1 to 255 for *timeout*
1 to 255 for *count*
1 to 255 for *delay*
0 to 65,535 for *port*

Description

Allows you to change defaults associated with RIP activity. Do not change the values for these parameters unless you know what the results will be.

activity Specifies RIP activity with the following labels:

YES	RIP active (acquires and propagates routing information)
NO	RIP inactive
SILENT	acquires, but does not propagate routing information on this subnetwork

cost Specifies the cost in gateway hops to route across this subnetwork

timeout Specifies the time in seconds between routing updates transmitted by RIP

count Specifies the number of timeouts that must expire before RIP marks a routing table entry as being unreachable, if no routing updates are received for that entry.

delay Specifies the time in seconds that RIP waits before propagating information about lost routes

port Specifies the UDP port that RIP uses on this processor

Note: *Some ports within this range are reserved for other protocols, so be extremely careful if you change this value.*

SECURITY=*security*

Default
UNCL

Description
Specifies the security level as a decimal value ranging from 0 to 15, or one of the following labels:

UNCL	0	PROG	4
CONF	1	REST	5
EFTO	2	SECR	6
MMMM	3	TOPS	7

Specifying security parameters (other than the default) when they are not actually required can cause network problems. Your network administrator can provide you with a list of required parameters

SUBNMASK=*(adr1,adr2,adr3,adr4)*

Description
Specifies the subnet address mask. TCP-IP Stack uses this mask to determine what portion of the HOSTID is to be interpreted as part of the network number (NETID).

If you do not specify a subnet mask, TCP-IP Stack generates a default mask appropriate to the network class.

Example 1

This example configures a locally attached DDN X.25 subnetwork.

```
DDN1      SUBNET     PRCSR=PRC1,IPNETID=(4),;
                     TYPE=DDN,PDNGRP=XLM1
```

Example 2

This example configures a remote DDN X.25 subnetwork that can be reached through a gateway at address 3.0.0.4 in one hop.

```
DDN2      SUBNET     PRCSR=PRC1,IPNETID=(4),;
                     IPGATEWY=(1,3,0,0,4)
```

Online Configuration Differences

None.

XEU – Defining External End-User Programs

The Telcon XEU statement defines the path to applications that are external to Telcon, whether in the DCP or on a host system. When used with TCP–IP Stack, the XEU statement does the following:

- It allows the destination to be an IP address, which enables DCA sessions to use the TCP/IP transport.
- It defines the path to the handler for the TN3270 terminal handler.

Format

```
name   XEU    [DS=ds | IPADDR=adr1,adr2,adr3,adr4],;
               [DESTSSU='poolname | luname'],;
               [DESTTSU='SNACONN'],;
               [DPP=TN3270]
```

Required Parameters

name

Description
A name you supply for an application.

Optional Parameters

DS=*ds*

Values
A configured name for a NETADR, PRCSR, or ADDRESS statement.

Description
If you are configuring DCA sessions over TCP/IP, use the name of an ADDRESS statement. The ADDRESS statement should contain the IP address of the destination where the application is located.

If you are configuring a TN3270 emulation, use the name of the PRCSR statement defining the DCP running SNA/net. At least one DCP in the network must be running the SNA/net program product, but TCP–IP and SNA/net do not need to coexist in the same DCP.

IPPADDR=*adr1,adr2,adr3,adr4*

Default
None

Range
0 to 255 for each segment

Description
The IP address of the host where the application is located.

DESTSSU='*poolname* | *luname*'

Default
The name of this XEU statement

Description
The destination session service user. When used to define the TN3270 handler, the value for this parameter is either the name of an SNA/net POOL or LU statement. The POOL or LU statement must be defined on the DCP running SNA/net. See the *SNA/net Configuration Guide* (7831 5686) for information on configuring SNA/net.

DESTTSU='SNACONN'

Default
DTPX

Description
The destination transport service user. You must configure the value SNACONN when you are configuring the TN3270 handler. It is used for connections through the SNA/net EPU2 component.

DPP=TN3270

Default
INT1

Description
The data presentation protocol. You must specify the value TN3270 when you are configuring the TN3270 handler.

Section 4
Controlling TCP-IP Stack Operation

Controlling TCP-IP Stack Operation is part of the overall Telcon network operation and management facility, Network Management Services (NMS). This section describes NMS commands that provide the diagnostic and management functions of the TCP-IP Stack. They are presented in a format similar to that of NMS commands in the *Telcon Operations Reference Manual* (7831 5728).

Summary of NMS Commands for TCP-IP Stack

The following are TCP-IP Stack NMS commands:

Command	Description
DISPLAY=ARP	Displays LAN IP-to-MAC address mapping
DISPLAY=IP	Displays IP status
DISPLAY=RIPNBR	Displays RIP neighbors
DISPLAY=ROUTE	Displays IP routing tables
DISPLAY = SAT	Displays source address table
DISPLAY=TCP	Displays active TCP connections (sockets)
HELP	Displays Help text
KILL=ARP	Deletes LAN IP-to-MAC address mapping, forcing an ARP request
KILL=RIPNBR	Deletes an RIP neighbor
KILL=TCP	Terminates a specific TCP connection
MODIFY=ROUTE	Modifies a specific entry in the IP routing table

Command	Description
PING=nnn.nnn.nnn.nnn	Sends ICMP Echo Request
SNAP=IP	Turns on IP traces and opens a trace file
SNAP=TCPTB	Turns on traces in the transport bridge and opens a trace file
SNAP=TCPTS	Turns on traces in the transport service and opens a trace file
SNOF=IP	Turns off IP traces and closes a trace file
SNOF=TCPTB	Turns off transport bridge traces and closes a trace file
SNOF=TCPTS	Turns of transport service traces and closes a trace file

Note: *To specify that these NMS commands apply to TCP-IP Stack, you must enter the prefix TCP, followed by a space, before each command.*

DISPLAY=ARP – Display ARP Address Mapping

Function

Displays the physical address-to-IP address mapping that TCP-IP Stack (ARP) maintains. Typically, it is used to determine if the TCP-IP Stack is communicating with or attempting to communicate with another TCP/IP host. If you use the command without the DEST parameter, the mapping for every address that ARP maintains is displayed.

Format

```
TCP    DISPLAY=ARP[,DEST={nn.nn.nn.nn|(adr1,adr2,adr3,adr4)}]
```

Parameters

DISPLAY=ARP

Description
Specifies ARP related address mapping information for LAN configurations

DEST=*nn.nn.nn.nn* | (*adr1,adr2,adr3,adr4*)

Default
XX.XX.XX.XX

Description
Optionally specifies address mapping information about a specific destination IP address. You can use one of two address formats to specify the destination address. The *nn.nn.nn.nn* format is similar to the standard Internet address format. The *(adr1,adr2,adr3,adr4)* format is identical to that used on TCP-IP Stack configuration statements in configuration files and online configuration. Each address segment (*nn* or *adr*) represents a value ranging from 0 to 255 (decimal). You can specify address values in decimal, hexadecimal, or binary notation, and use 'wildcard' characters to match or mask values as follows:

Notation	Prefix	Example for decimal 63	Wildcard Example for any byte
Decimal	none	63	XX
Hexadecimal	0X	0X3F (=63)	0XXX (X=any 4 bits)
Binary	0B	0B111111 (=63)	0BBBBBBBB (B=any 1 bit)

Note: *Decimal notation allows only the XX wildcard designation*

Example 1

```
TCP DISPLAY=ARP
```

Displays all ARP address mapping for this Telcon node.

Example 2

```
TCP DISPLAY=ARP,DEST=(63,XX,XX,XX)
```

Displays all ARP address mapping for all IP addresses on network number 63.

DISPLAY=IP – Display IP Status

Function

Lets you collect and display IP run-time statistics. You can reset IP–related statistical counters to begin data collection at a known time. The IP counters have a range of 0 to $(2^{32}-1)$= 4,294,967,295 events. Items counted include the following:

- Inbound segments and datagrams
- Outbound segments and datagrams
- Inbound and outbound fragmented datagrams
- Datagram errors and timeouts

You can collect and display statistics for a specific network service provider (NSP) interface or for all IP communications handled through the local processor.

Format

```
TCP  DISPLAY=IP[,SUBNET={subnet|ALL}][,RESET={YES|NO}]
```

Parameters

DISPLAY=IP

Description
Collects and displays IP related run-time statistical information.

SUBNET=*subnet*|ALL

Default
ALL

Description

subnet	Specifies a network service provider (NSP) interface for which information will be displayed. Identify the NSP by specifying either the network number from the IPNETID parameter of a SUBNET statement or the name of a SUBNET statement.
ALL	Specifies all NSPs (default).

RESET=YES ! NO

Description

YES	Resets all statistical counters associated with the specified IP address or NSP interface. If you do not specify the DEST or SUBNET parameters, only counters associated with the Telcon node from which the command was issued are reset.
NO	Does not reset statistical counters

Example

```
TCP  DISPLAY=IP,RESET=YES
```

Resets all counters.

DISPLAY=RIPNBR – Display RIP Neighbors

Function

Displays information on all known RIP neighbors on the Telcon DNS network. This information includes the neighbor's IP address and the subnetwork with which the neighbor is associated.

Format

```
TCP DISPLAY=RIPNBR[,SUBNET={nn.nn.nn.nn|(adr1,adr2,adr3,adr4 )}]
```

Parameters

DISPLAY=RIPNBR

Description
This is the command to display information on RIP neighbors

SUBNET=*nn.nn.nn.nn.* | *(adr1,adr2,adr3,adr4)*

Default
XX.XX.XX.XX

Description
Specifies the subnet on which the RIP neighbors are located. Use the value specified on the IPNETID parameter of a SUBNET statement to identify the subnet.

Notation	Prefix	Example for decimal 63	Wildcard Example for any byte
Decimal	none	63	XX
Hexadecimal	0X	0X3F (=63)	0XXX (X=any 4 bits)
Binary	0B	0B111111 (=63)	0BBBBBBBBB (B=any 1 bit)

Note: *Decimal notation allows only the XX wildcard designation.*

Example

```
TCP  DISPLAY=RIPNBR,SUBNET=124.54.0.34
```

Gathers information on the RIP neighbor with network number 124.54.0.34.

DISPLAY=ROUTE – Display IP Routing Tables

Function

Displays IP routing table information for either a specified destination IP address or for all IP addresses known to TCP-IP Stack.

Format

```
TCP  DISPLAY=ROUTE[,DEST={nn.nn.nn.nn|(adr1,adr2,adr3,adr4 )}]
```

Parameters

DISPLAY=ROUTE

Description
Displays IP routing table information

DEST=*nn.nn.nn.nn* | *(adr1,adr2,adr3,adr4)*

Default
XX.XX.XX.XX (specifies all IP addresses)

Description
Specifies a destination IP address for which routing table information is displayed. You can use one of two address formats to specify the destination address. The *nn.nn.nn.nn* format is similar to the the standard Internet address format. The *(adr1,adr2,adr3,adr4)* format is identical to that used on TCP-IP Stack configuration statements in configuration files and online configuration. Each address segment (*nn* or *adr*) represents a value ranging from 0 to 255 (decimal). You can specify address values in decimal, hexadecimal, or binary notation and use 'wildcard' characters to match or mask values as follows:

Notation	Prefix	Example for Decimal 63	Wildcard Example for Any Byte
Decimal	none	63	XX
Hexadecimal	0X	0X3F (=63)	0XXX (X=any 4 bits)
Binary	0B	0B111111 (=63)	0BBBBBBBBB (B=any 1 bit)

Note: *Decimal notation allows only the XX wildcard designation*

Example

```
TCP  DISPLAY=ROUTE
```

Displays routing table information for all destination IP addresses known to TCP-IP Stack on the local processor.

Note: *The routing table display can be very large and may be truncated. Use the DEST parameter to display routing information for specific entries.*

DISPLAY=SAT – Display Source Address Table

Function

Allows the display of all addresses configured for TCP-IP Stack. This includes addresses assigned to DCA TSs or DNS nodes that do not reside in the same Telcon node as the TCP-IP Stack, and for which TCP-IP Stack provides a DCA-to-TCP/IP gateway.

Format

```
TCP DISPLAY=SAT
```

Example

```
>TCP DISPLAY=SAT
>
>07/12/92 14:42:52 ***  COMMAND ACCEPTED ***
>07/12/92 14:42:52 *** TCP-IP SOURCE ADDRESS ***
>
>SOURCE ADDRESS           DCA ADDRESS
>129.221.2.91             1.1.1.2
>129.221.2.91             13,2
>129.221.2.92             1.1.1.3
>129.221.2.92             13,12
>129.221.2.93             1.1.1.4
>129.221.2.93             13,27
>***  FUNCTION COMPLETE  ***
>_
```

The example shows several IP addresses in use by the DCP. All are paired with DNS addresses (for example: 129.221.2.91 through 1.1.1.2) and processor TRUNK statement identifications (for example: 129.221.2.91 through 13,2). Depending on the TCP-IP configuration and the use of DNS versus TS/TN, a DCP can show the IP addresses paired with either DNS or TS/TN addresses, or both.

DISPLAY=TCP – Display Active TCP Connections

Function

Displays the status of all TCP connections. You can gather this status based on the source or destination IP address, on both, or on a source or destination TCP port number. To uniquely identify a single TCP connection, specify all the parameters.

Format

```
TCP    DISPLAY-TCP [ [,SRC-{nn.nn.nn.nn | (adr1,adr2,adr3,adr4)}] ]
                   [ [,DEST-{nn.nn.nn.nn | (adr1,adr2,adr3,adr4)}] ]
                   [,SRCPORT- xxxx ][,DESTPORT- xxxx ]
```

Parameters

DISPLAY=TCP

Description
Displays TCP connection information

SRC=*nn.nn.nn.nn*|(*adr1,adr2,adr3,adr4*)

Default
xx.xx.xx.xx (all destinations)

Description
Specifies a source IP address for which TCP connection information will be displayed. Refer to DISPLAY=ARP for information on the use of wildcard characters.

DEST=*nn.nn.nn.nn* | (*adr1,adr2,adr3,adr4*)

Default
xx.xx.xx.xx (all destinations)

Description

Specifies a destination IP address for the TCP connection information to be displayed.

Notation	Prefix	Example for decimal 63	Wildcard Example for any byte
Decimal	none	63	XX
Hexadecimal	0X	0X3F (=63)	0XXX (X=any 4 bits)
Binary	0B	0B111111 (=63)	0BBBBBBBB (B=any 1 bit)

Note: *Decimal notation allows only the XX wildcard designation.*

SRCPORT=*xxxx*

Default
0 (all ports)

Description
Specifies a source TCP port number

DESTPORT=*xxxx*

Default
0 (all ports)

Description
Specifies a destination TCP port number

Example

```
TCP DISPLAY=TCP,SRC=124.54.0.34,DEST=124.54.0.36
```

Displays the active TCP connections between the source and destination IP addresses specified. The following is a list of TCP connection statement abbreviations:

**	Closed	TW	Time Waiting
W1	FIN wait1	LI	Listen
W2	FIN wait2	CL	Closing
SS	Syn sent	CW	Close wait
SR	Syn received	LA	Last Ack
ES	Established		

HELP – Displays Online Help Text

Function

- Leaving off the first parameter displays a list of accepted TCP-IP commands that can be used as values for cmd, in addition to general help information regarding the NMS commands supported by TCP-IP Stack.
- Specifying one of the values for cmd provides a list of accepted types that the command can be used with, in addition to general help regarding the TCP-IP Stack NMS command identified by cmd.

This information is followed by additional screens of help text for each individual type specified.

Format

```
TCP HELP[ cmd[ type]]
```

Parameter

cmd **Description** The name of the command, such as DISPLAY or KILL.
type **Description** The TCP/IP facility you want the command to work with, such as ARP or ROUTE.

Example

```
TCP  HELP  DISPLAY
```

Description
This command displays several pages of information regarding DISPLAY commands, beginning with a list of facilities that you can display information about.

KILL=ARP – Delete ARP Address Mapping

Function

Deletes ARP addresses from the cache. Normally, these addresses are updated when the ARP timer expires. Use the KILL=ARP command to update addresses before the timer expires.

Format

```
TCP  KILL=ARP[,DEST={nn.nn.nn.nn|(adr1,adr2,adr3,adr4)}]
```

Parameters

KILL=ARP

Description
This is the command to delete ARP addresses

DEST=*nn.nn.nn.nn* | (*adr1,adr2,adr3,adr4*)

Default
xx.xx.xx.x (all destinations)

Description
Specifies a destination IP address to delete. If you do not use the DEST parameter, all ARP addresses are deleted.

Notation	Prefix	Example for decimal 63	Wildcard Example for any byte
Decimal	none	63	XX
Hexadecimal	0X	0X3F (=63)	0XXX (X=any 4 bits)
Binary	0B	0B111111 (=63)	0BBBBBBBBB (B=any 1 bit)

Note: *Decimal notation allows only the XX wildcard designation.*

Example

```
TCP KILL ARP DEST=124.54.0.3
```

Deletes ARP addresses from the cache at the IP address specified.

KILL=RIPNBR – Remove an RIP Neighbor

Function

Removes discovered RIP neighbors. Use this command on DNS networks only.

Format

```
TCP KILL=RIPNBR,ADR={nn.nn.nn.nn|(adr1,adr2,adr3,adr4)}
```

Parameters

KILL=*RIPNBR*

Description
This is the command to remove an RIP neighbor from the neighbor list

ADR=*nn.nn.nn.nn* | *(adr1,adr2,adr3,adr4)*

Default
xx.xx.xx.xx (all destinations)

Description
Specifies the address of the RIP neighbor to be removed.

Notation	Prefix	Example for decimal 63	Wildcard Example for any byte
Decimal	none	63	XX
Hexadecimal	0X	0X3F (=63)	0XXX (X=any 4 bits)
Binary	0B	0B111111 (=63)	0BBBBBBBBB (B=any 1 bit)

Note: *Decimal notation allows only the XX wildcard designation.*

Example

```
TCP KILL RIBNBR ADR=124.54.0.3
```

Deletes an RIP neighbor from the neighbor list.

KILL=TCP – Terminate a TCP Connection

Function

Terminates TCP connections. Use it to free a suspended TCP connection when no keep-alive mechanism is configured. You must use all the parameters listed to uniquely identify a TCP connection.

Format

```
TCP    KILL=TCP,SRC={nn.nn.nn.nn|(adr1,adr2,adr3,adr4)}
              ,DEST={nn.nn.nn.nn|(adr1,adr2,adr3,adr4)}
              ,SRCPORT=xxxx,DESTPORT=xxxx
```

Parameters

KILL=TCP

Description
This is the command to terminate a TCP connection

SRC=*nn.nn.nn.nn* | *(adr1,adr2,adr3,adr4)*

Description
Specifies the source IP address of a TCP connection to terminate. This parameter does not allow the use of wildcard characters.

DEST=*nn.nn.nn.nn* | *(adr1,adr2,adr3,adr4)*

Description
Specifies the destination IP address of a TCP connection to terminate. This parameter does not allow wildcard characters.

SRCPORT=*xxxx*

Description
Specifies the source TCP port number of a TCP connection to terminate

DESTPORT=*xxxx*

Description
Specifies a destination TCP port number of a TCP connection to terminate

Example

```
TCP KILL=TCP,SRC=124.54.0.34,DEST=124.54.0.36,SRCPORT=264,DESTPORT=264
```

Terminates a TCP connection between the IP addresses and ports shown.

MODIFY=ROUTE - Modify an IP Routing Table Entry

Function

Changes entries in the routing table. The command requires the DEST parameter.

Format

```
TCP    MODIFY=ROUTE,DEST={nn.nn.nn.nn|(adr1,adr2,adr3,adr4 )}
                        [,GATEWAY={nn.nn.nn.nn|(adr1,adr2,adr3,adr4 )}]
                        [,COST={nn|INFINITE}][,LOCK={YES|NO}]
```

Parameters

MODIFY=ROUTE

Description
This is the command to modify an entry in the routing table.

DEST=*nn.nn.nn.nn* | *(adr1,adr2,adr3,adr4)*

Description
This is the subnetwork number of the destination subnetwork. This parameter does not allow wildcard characters.

GATEWAY=*nn.nn.nn.nn* | *(adr1,adr2,adr3,adr4)*

Description
This is the IP address of the first router on the path to the destination subnetwork. This parameter does not allow wildcard characters.

COST=*nn* | INFINITE

Default
1

Description
This is the number of hops to the destination. Entering INFINITE causes the route to be deleted.

LOCK=YES | NO

Description
Specifies whether RIP is to update the route.

Example

```
TCP MODIFY=ROUTE,DEST=124.54.0.36,GATEWAY=120.45.9.54,COST=3
```

Changes the route to the host at IP address 124.54.0.36. It specifies the first gateway to the destination subnetwork and the number (3) of gateway hops to the destination subnetwork.

PING – Sends ICMP Echo Request

Function

This command sends an ICMP echo request to another TCP-IP host. This allows the operator or network administrator to collect information about connectivity.

Format

```
TCP    PING=nnn.nnn.nnn.nnn[,TIMEOUT=nn]
                            [,REPEAT=nnn]
                            [,LENGTH=nnn]
                            [,RECORD={YES|NO}]
```

Parameters

PING=*nnn.nnn.nnn.nnn*

Description
The address of the TCP/IP host that is to be pinged. This parameter is required.

TIMEOUT=*nn*

Default
2 seconds

Range
1-99 seconds

Description
Specifies the amount of time that TCP-IP will wait for a response from the TCP/IP host being pinged.

REPEAT=*nnn*

Default
1

Range
1-99

Description
Specifies the number of attempts that are to be made for this PING command. Successive attempts are made following every response, or following every timeout.

LENGTH=*nnn*

Default
100 bytes

Range
40-576 bytes

Description
Specifies the length of the packet to be sent. It is useful when checking maximum transmission unit sizes for routers and bridges.

RECORD=YES | NO

Default
NO

Description
Requests TCP-IP record and report the route taken for successful PING attempts.

SNAP=IP – Turn On IP Traces

Function

Turns on message tracing in the internet component.

Format

```
TCP      SNAP-IP[,DIR-{IN|OUT|BOTH}][,IF-{NSP|ULP|BOTH}]
              [,FAC-{line|pdngrp|channel|ALL}]
              [,SRC-{nn.nn.nn.nn|(adr1,adr2,adr3,adr4)}]
              [,DEST-{nn.nn.nn.nn|(adr1,adr2,adr3,adr4)}]
              [,PID-{ICMP|TCP|UDP|xxxx|ALL}]
              [,FILE-filename][,PARSE-{IP|NONE}]
              [,REUSE-n]
              [,LENGTH-length]
```

Parameters

SNAP=IP

Description
This is the command that turns on message tracing for the IP software module.

DIR=IN | OUT | BOTH

Default
BOTH

Description
Specifies the direction in which messages are to be traced

IN	Specifies that only messages being sent from a TCP/IP network are to be traced
OUT	Specifies that only messages to a TCP/IP network are to be traced
BOTH	Specifies that all messages coming from or being sent to a TCP/IP network are to be traced

IF=NSP | ULP | BOTH

Default
BOTH

Description
Specifies the interface where messages are to be traced

NSP	Specifies the network service provider, which means messages are to be traced on the IP side of the interface
ULP	Specifies the upper layer protocol, which means messages are to be traced on the TCP or user datagram protocol (UDP) sides of the interface
BOTH	Specifies that messages are to be traced on both sides of the interface

FAC=*line* | *pdngrp* | *channel* | ALL

Default
ALL

Description
Specifies the facility that is to be traced. If you do not use this parameter, all facilities are traced.

line Specifies a LINE statement that defines the line to trace

pdngrp Specifies a PDNGRP statement that defines the PDN line or lines to trace. Normally, there is a one-to-one relationship between a LINE and a PDNGRP. If you have multilink configured, however, several LINEs form a PDNGRP. Therefore, use this parameter only with multilink.

channel Specifies a CHANNEL statement that defines a host channel connection to trace

ALL Specifies tracing for all facilities.

SRC=*nn.nn.nn.nn* | (*adr1,adr2,adr3,adr4*)

Default
XX.XX.XX.XX (all addresses)

Description
This is the source IP address. Datagrams with this address are traced. If the DIR parameter specifies BOTH, then SRC and DEST are exchangeable.

Notation	Prefix	Example for decimal 63	Wildcard Example for any byte
Decimal	none	63	XX
Hexadecimal	0X	0X3F (=63)	0XXX (X=any 4 bits)
Binary	0B	0B111111 (=63)	0BBBBBBBBB (B=any 1 bit)

Note: *Decimal notation allows only the XX wildcard designation.*

DEST=*nn.nn.nn.nn* | (*adr1,adr2,adr3,adr4*)

Default
XX.XX.XX.XX.XX (all addresses)

Description
This is the destination IP address. Datagrams with this address are traced. If the DIR parameter specifies BOTH then SRC and DEST are exchangeable. Refer to SRC for information on the use of wildcard characters.

PID=ICMP | TCP | UDP | *xxxx* | ALL

Default
ALL

Description
Limits tracing to datagrams of a specific ULP, which can be specified by name or number.

ICMP	Specifies tracing for the ICMP component
TCP	Specifies tracing for the TCP component
UDP	Specifies tracing for the UDP component
xxxx	Specifies tracing for all messages on a port, which is identified by a PID parameter on a LINE statement
ALL	Specifies tracing for all ULPs. This is the default.

FILE=*filename*

Description
Specifies the name of a file to which tracing information is sent. If the filename is not catalogued, then TCP-IP Stack catalogues the file automatically.

This parameter can not be changed unless all tracing is terminated by the SNOF command.

PARSE=IP | NONE

Default
NONE

Description
Specifies if output to the screen should be presented by the protocol.

IP	Specifies that tracing information on IP should be presented separately on the terminal screen
NONE	Specifies that tracing information should not be presented separately by the protocol.

REUSE=*n*

Default
1

Description
Specifies the number of trace files to open while tracing messages. This value is either 1, which indicates that trace files are not to be reused, or a value ranging from 2 to 99, which causes TCP/IP to open a maximum of that many files and reuse the first file when the last file is full. When REUSE is greater than 1, the specified filename is truncated to no more than six characters and a two-digit sequence number is added to the end of the filename. If REUSE is specified, and the filename is not specified, then the filename is set to TRACEFIL.

This parameter can not be changed unless all tracing is first terminated by the SNOF command.

LENGTH=*length*

Default
0 (zero)

Description
This parameter is used to limit the length of the traced messages. If the length of the message is greater than the specified length, then the traced message is truncated to the specified length. The default of 0 specifies no length limit.

Example

```
TCP SNAP=IP SRC=24.5.5.165
```

Turns tracing on for datagrams originating at or sent to the IP address specified.

SNAP=TCPTB – Turn On Transport Bridge Traces

Function

Turns on message tracing in the transport bridge component.

Format

```
TCP        SNAP=TCPTB[,DIR={IN|OUT|BOTH}][,IF={TSU|TCP|BOTH}]
               [,FILE=filename][,REUSE=n]
               [,LENGTH=length]
```

Parameters

SNAP=TCPTB

Description
Turns on message tracing for the TCP bridge software module. The TCPTB allows bridging between the DCA transport service (DTPX) and the TCP transport service.

DIR=IN | OUT | BOTH

Default
BOTH

Description
Specifies the direction in which messages are to be traced

IN	Specifies that only messages sent from a TCP/IP network are to be traced
OUT	Specifies that only messages sent to a TCP/IP network are to be traced
BOTH	Specifies that all messages sent from or to a TCP/IP network are to be traced

IF=TSU | TCP | BOTH

Default
BOTH

Description
Specifies the interface where messages are to be traced

TSU	Specifies the transport service user, which means messages are to be traced on the Telcon side of the interface
TCP	Specifies the transport control protocol, which means messages are to be traced on the TCP side of the interface
BOTH	Specifies that messages are to be traced on both sides of the interface

FILE=*filename*

Description
Specifies the name of a file to which tracing information is sent. If the filename is not catalogued, then TCP-IP Stack catalogues the file automatically.

If REUSE is specified, and the filename is not specified, then the filename is set to TRACEFIL.

This parameter cannot be changed unless all tracing is terminated by the SNOF command.

REUSE=*n*

Default
1

Range
1 to 99

Description
Specifies the number of trace files to open while tracing messages.

1 indicates that trace files are not to be reused

2 - 99 causes TCP-IP to open a maximum of that many files and reuse the first file when the last file is full. The specified filename is truncated to no more than six characters and a two-digit sequence number is added to the end of the filename.

This parameter can not be changed unless all tracing is first terminated by the SNOF command.

LENGTH=*length*

Default
0 (zero)

Description
This parameter is used to limit the length of the traced messages. If the length of the message is greater than the specified length, then the traced message is truncated to the specified length. The default of 0 specifies no length limit.

SNAP=TCPTS – Turn on Transport Service Traces

Function

Turns on message tracing in the TCP transport service (TCP TS) component.

Format

```
TCP       SNAP=TCPTS[,DIR={IN|OUT|BOTH}][,IF={TSU|TCP|BOTH}]
              [,SRC=source TSAP-ID]
              [,DEST=destination TSAP-ID]
              [,FILE=filename][,REUSE=n]
              [,LENGTH=length]
```

Parameters

SNAP=TCPTS

Description
Turns on message tracing for the TCP TS software module. The TCP TS provides an interface to users, allowing them to use the TCP transport.

DIR=IN | OUT | BOTH

Default
BOTH

Description
Specifies the direction in which messages are to be traced

IN	Specifies that only messages sent from a TCP/IP network are to be traced
OUT	Specifies that only messages sent to a TCP/IP network are to be traced
BOTH	Specifies that all messages sent from or to a TCP/IP network are to be traced

IF=TSU | TCP | BOTH

Default
BOTH

Description
Specifies the interface where messages are to be traced

TSU	Specifies the transport service user, which means messages are to be traced on the user side of the interface
TCP	Specifies the transport control protocol, which means messages are to be traced on the TCP side of the interface
BOTH	Specifies that messages are to be traced on both sides of the interface

SRC=*source TSAP-ID*

Description

The ID of the local transport service access point.

TSAP IDS are registered with the TCP transport service by each transport user. TSAP IDS that are received by TCP TS and have not been registered are passed to the TCP transport bridge where they are treated as XEU names.

If the DIR parameter specifies BOTH, then SRC and DEST are exchangeable.

The format of a TSAP ID can be an 'ASCII string' or a x'hexadecimal string' up to 20 bytes long. The hexadecimal string must be an even number of characters because two hexadecimal characters specify one byte.

Telcon registers the following values with the TCP TS:

ID	USER
'DTPXbbbb'	DTPX
'NMSbbbbb'	NMS
'INTbbbbb'	Interactive
'RBbbbbbb'	Remote batch
'HOSTMSbb'	MSAP protocol
'DX'	DDP bridge. DX is the prefix. The suffix is the name of a NETADR, DCATS, or XEU statement name. The value DX*name* must be configured in the CMS 1100 TRANSPORT statement as the value for the TRANSPORT-USER parameter.

Note: *'b' signifies a blank space. Blank spaces must be specified in full.*

DEST=*destination TSAP-ID*

Description

The destination transport service access point ID.

TSAP IDs are registered with the TCP TS by each user of the transport. TSAP IDs that are received by TCP-TS and have not been registered are passed to the TCP transport bridge where they are treated as XEU names.

If the DIR parameter specifies BOTH,then SRC and DEST are exchangeable.

The format of a TSAP ID can be an 'ASCII string' or a x'hexadecimal string' up to 20 bytes long. The hexadecimal string must be an even number of characters, because two hexadecimal characters specify one byte.

Telcon registers the following TSAP-IDs with TCP–IP Stack.

ID	User
'DTPXbbbb'	DTPX
'NMSbbbbb'	NMS
'INTbbbbb'	Interactive
'RBbbbbbb'	Remote batch
'HOSTMSbb'	MSAP protocol
'DX'	The DDP bridge. DX is the user prefix. The suffix is the name of a NETADR, DCATS, or XEU statement name. The value DX*name* must be configured in the CMS 1100 TRANSPORT statement as the value for the TRANSPORT-USER parameter.

Note: *'b' signifies a blank space. Blank spaces must be specified in full.*

FILE=*filename*

Description

Specifies the name of a file to which tracing information is sent. If the filename is not catalogued, then TCP-IP Stack catalogues the file automatically.

If REUSE is specified, and the filename is not specified, then the filename is set to TRACEFIL.

This parameter cannot be changed unless all tracing is terminated by the SNOF command.

REUSE=*n*

Default
1

Range
1 TO 99

Description
Specifies the number of trace files to open while tracing messages.

1	Indicates that trace files are not to be reused
2 - 99	Causes TCP-IP to open a maximum of that many files and reuse the first file when the last file is full. The specified filename is truncated to no more than six characters and a two-digit sequence number is added to the end of the filename.

This parameter cannot be changed unless all tracing is first terminated by the SNOF command.

LENGTH=*length*

Default
0 (zero)

Description
Limits the length of the traced messages. If the length of the message is greater than the specified length, then the traced message is truncated to the specified length. The default of 0 specifies no length limit.

Example

The following example traces all INT-1 (interactive) sessions through TCP TS.

```
TCP SNAP=TCPTS,SRC='INT      '
```

SNOF=IP – Turn Off IP Traces

Function

Turns off the IP message tracing started by the SNAP=IP command. If there are no other active TCP/IP traces, it also closes an open trace file.

Format

```
TCP SNOF=IP
```

Parameters

SNOF=*IP*

Description
This is the command that turns IP tracing off

Example

```
TCP SNOF=IP
```

Turns IP tracing off and closes the trace file if no other traces are active.

SNOF=TCPTB – Turn Off Transport Bridge Traces

Function

Turns off the transport bridge message tracing started by the SNAP=TCPB command. If there are no other active TCP/IP traces, it also closes an active trace file.

Format

```
TCP SNOF=TCPTB
```

Parameters

SNOF=TCPTB

Description
Turns transport bridge tracing off

Example

```
TCP SNOF=TCPTB
```

Turns transport bridge tracing off and closes the trace file if no other traces are active.

SNOF=TCPTS – Turn Off Transport Service Traces

Function

Turns off the transport service message tracing started by the SNAP=TCPTS command. If no other TCP traces are active, it closes the trace file, if one is open.

Format

```
TCP SNOF=TCPTS
```

Parameters

SNOF=TCPTS

Description
Turns transport service tracing off.

Example

```
TCP SNOF=TCPTS
```

Turns transport service tracing off and closes the trace file if no other traces are active.

Appendix A
Configuration Examples

This appendix provides examples of TCP-IP Stack configurations. It consists of two subsections. The first subsection provides TCP-IP Stack configurations in dynamic network services (DNS) networks. The second provides TCP-IP Stack configurations in termination systems/transport network (TS/TN) networks.

A.1. Telcon DNS Configurations

The following examples use DNS facilities, which provide connectionless network services for DCA Telcon networks.

DNS connectionless network layer services eliminate the need to define logical routes between hardware devices in a DCA network. With DNS, you provide a name and a DNS network address for each node in your network. DNS uses network addresses to accomplish node-to-node routing throughout the DCA network. You do not need to define sessions or termination systems with DCPTS, DCATS, or SESSN configuration statements.

DNS replaces TS/TN, in the host and DCPs. Your network, however, can include both DNS and earlier network layer protocols for this particular communications delivery.

A.1.1. Configuring the DCP as an IP Router Between a LLC LAN and the DDN

This example shows a DCP configured as an IP router between an LLC LAN and the DDN. An OS 1100 host running DDN 1100 is attached to the LAN through a Host LAN Controller (HLC). Figure A–1 depicts this configuration.

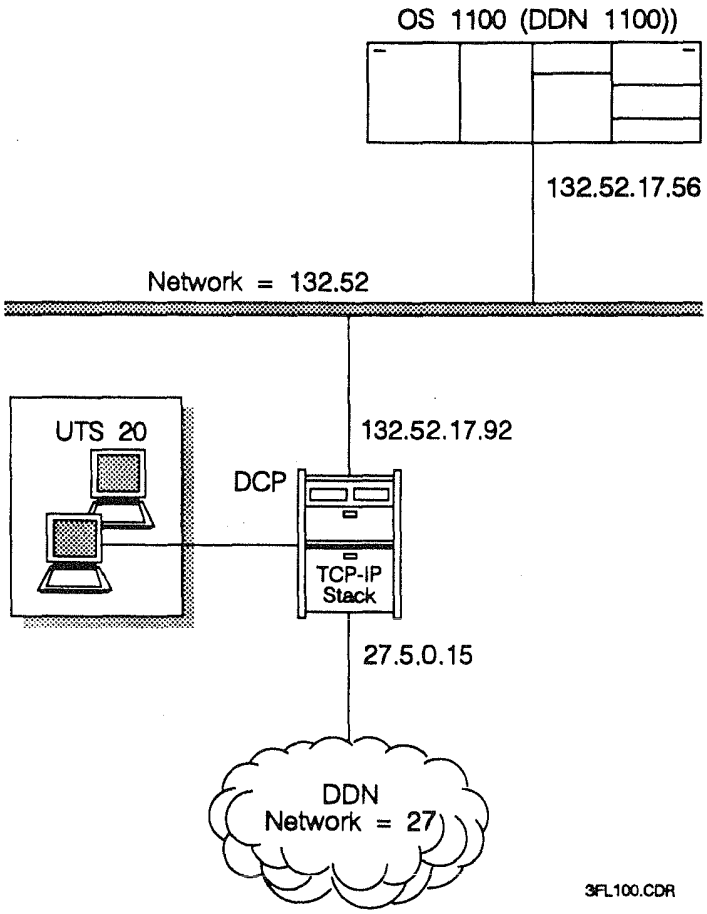


Figure A–1. DCP as an IP Router between an LLC LAN and the DDN

```
. CMS 1100 CONFIGURATION
. HLC ATTACHED HOST RUNNING DDN 1100 WITH CMS 1100 TCP/IP
.
NODE                    NODE-ADDRESS,10
.
TRACE                   ASG,*TRACE(+1).,F///1000
.
STATIC-BANKS            TCP/IP,DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
REM-HOSTS               FILE,CMS*TCPIP.RHOSTS,RFC952
.
```

```
PROCESS,DDP             TYPE,XCSU ;
                        ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                        STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT,5000 ;
                        INTERNET-ADR,INTH1

PROCESS,DCP             TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP

PROCESS,RSDCSU          TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP

INTERNET-ADR,INTH1      IP,IP1,132.52.17.56

IP,IP1                  LINK-LAYER,LAN1 ;
                        GATEWAY,132.52.17.92
                        INTERFACE-TYPE,IEEE802.3

LAN,LAN1                INPUT-NODE,LAN1I ;
                        OUTPUT-NODE,LAN1O ;
                        LOCAL-ADR,X08000BE00C0A ;
                        STATUS,UP

NEIGHBOR,PRC1           LAN,LAN1 ETHERNET-ADR,X08000B0C360B ;
                        LOAD,TEL*LOAD.PRC1
.


*****************************************************************
* TELCON CONFIGURATION
*****************************************************************
** OS 1100 HOST (ATTACHED TO LAN BY HLC) DNS ADDRESS DEFINITION:
*
HOST1           NETADR      NA=10
*****************************************************************
** DCP PROCESSOR DEFINITION:
*
DCP1            NETADR      NA=1
PRC1            PRCSR       NA=1
*****************************************************************
** DEMAND TERMINAL EXTERNAL END-USER (XEU) DEFINITION:
** (CONFIGURES DEMAND MODE TERMINAL ACCESS TO HOST)
*
DEMH1           XEU         DS=HOST1,DESTTSU='RSDCSU'
*****************************************************************
** XEU DEFINITION FOR DDP TERMINAL ACCESS:
** (CONFIGURES TERMINAL ACCESS TO DDN 1100 APPLICATIONS)
*
DDPH1           XEU         DS=HOST1,DESTTSU='DDP'
*****************************************************************
*  NETWORK CONSOLE DEFINITIONS
*
NMSC            XEU         NA=1,DESTSSU='NMSC'
OS              XEU         NA=1,DESTSSU='EUOS'
EUOS            EU          PRCSR=PRC1,TYPE=DCPOS,INFO='MONITOR'
*
*****************************************************************
*    ALLOWS NMS ACCESS THROUGH OS CONSOLE
*
CONSP1          TERM        PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                            DEST=NMSC,ALOC=YES,CEDS=YES,OPDS=YES,;
                            PPID=OFF
```

```
*
*****************************************************************
*    NMS LINE CLASS, LINE, AND TERMINAL DEFINITION
*
UN96SH          LCLASS      LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
*
LNMSP1          LINE        PRCSR=PRC1,CLASS=UN96SH,ADR=04
*
GNMSP1          GROUP       PRCSR=PRC1,LINE=LNMSP1
*
NMSCP1          TERM        GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                            DEST=NMSC,FMT=(24,80),AUTH=PRIV,;
                            ALOC=IMMED
*
*****************************************************************
*    UNISCOPE TERMINAL (TERMA1) DEFINITION FOR HOST ACCESS.
*     SECOND SCREEN OF NMS CONSOLE TERMINAL.  ($$SON TO TERMA1)
*
*
TERMA1          TERM        GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
*
*****************************************************************
** PSCS CONFIGURATION FOR DDN X.25 ATTACHMENT:
*
X2596           LCLASS      LPH=X25PKT,OPTIONS=(DIR,SYFD),SPEED=9600
X25DFDN1        X25DEF      NETWORK=DDNX25
PGPP1DN         PDNGRP      PRCSR=PRC1,X25DEF=X25DFDN1,VCGRP=(1,50),;
                            DTEADR='0000001215000'
LNP112          LINE        PDNGRP=PGPP1DN,CLASS=X2596,ADR=X'12'
*
*****************************************************************
** LAN PLATFORM CONFIGURATION FOR LLC LAN ATTACHMENT:
*
LANLC           LCLASS      LPH=ILML
LNP111          LINE        PRCSR=PRC1,CLASS=LANLC,ADR=X'11',;
                            STA=X'08000B0C360B'
*
*****************************************************************
** TCP-IP STACK CONFIGURATION:
*
** EU AND XEU DEFINITIONS FOR TCP-IP STACK:
** (CONFIGURES SERVER AND USER TELNET ON PRC1)
*
EUTCP           EU          PRCSR=PRC1,TYPE=TCPIP
USRTEL          XEU         DS=PRC1,DESTTSU='DTPX',DESTSSU='EUTCP'
*
** SUBNET AND IPADR DEFINITIONS FOR DDN X.25 AND LAN ATTACHMENTS:
*
SNDDNP1         SUBNET      PRCSR=PRC1,TYPE=DDN,IPNETID=27,;
                            PDNGRP=PGPP1DN,IPROUTER=YES
SNLLCP1         SUBNET      PRCSR=PRC1,TYPE=LANLLC,IPNETID=(132,52),;
                            LINE=LNP111,IPROUTER=YES
IPP1DDN         IPADR       PRCSR=PRC1,DCAEP=DCP1,;
                            IPADDR1=(27.5.0.15.LOCAL)
IPP1LAN         IPADR       PRCSR=PRC1,DCAEP=DCP1,;
                            IPADDR1=(132.52.17.92.LOCAL)
*
*****************************************************************
*END OF EXAMPLE
*****************************************************************
```

A.1.2. Configuring a DCP IP Router Between an LLC LAN and a Channel-Attached Host

This example shows a DCP configured as an IP router between an LLC LAN attachment and an OS 1100 host. The host, running DDN 1100, is attached to the DCP through a host channel configured as a TCP/IP subnetwork. This allows TCP-IP Stack on the DCP to perform IP routing between the host and the LAN. Using CMS 1100 TCP/IP communications, the OS 1100 host is equivalent to any other TCP/IP host. Figure A–2 depicts this configuration.

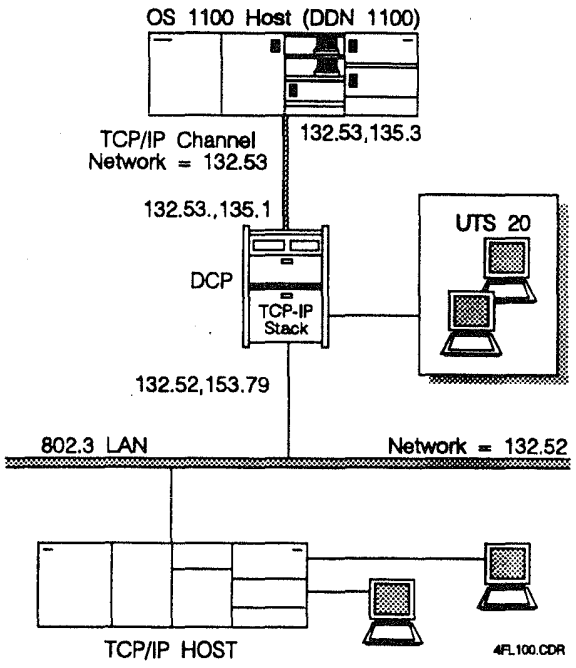


Figure A–2. DCP IP Router Between an LLC LAN and a Channel-Attached Host

```
*CMS 1100 CONFIGURATION
. CHANNEL ATTACHED HOST RUNNING DDN 1100 WITH CMS 1100 TCP/IP
.
NODE                    NODE-ADDRESS,10
.
TRACE                   ASG,*TRACE(1).,F///100
.
STATIC-BANKS            TCP/IP,DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS.
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
REM-HOSTS               FILE,CMS*TCPIP.RHOSTS,RFC952
.
```

```
PROCESS,DDP             TYPE,XCSU ;
                        ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                        STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT,5000 ;
                        INTERNET-ADR,INTH1

PROCESS,DCP             TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP

PROCESS,RSDCSU          TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP

INTERNET-ADR,INTH1      IP,IP1,132.53.135.3

IP,IP1                  LINK-LAYER,FEPP1 ;
                        INTERFACE-TYPE,DCP-CHANNEL ;
                        GATEWAY,132.53.135.1

FEP,FEPP1               PATHS,PATH1

PATH,PATH1              OWNER,FEPP1 ;
                        TYPE,BLOCK ;
                        INPUT-NODE,INODE8 ;
                        OUTPUT-NODE,ONODE8

RSI                     TIME-OUTS,YES ;
                        BLOCKING-FACTOR,25,4

*****************************************************************
* TELCON CONFIGURATION
*****************************************************************
** OS 1100 HOST DNS NETWORK ADDRESS DEFINITION:
*
HOST1           NETADR      NA=10
*****************************************************************
** DCP PROCESSOR DEFINITION:
*
DCP1            NETADR      NA=1
PRC1            PRCSR       NA=1
*****************************************************************
** EXTERNAL END-USER (XEU) DEFINITION:
** (CONFIGURES DEMAND MODE TERMINAL ACCESS TO HOST)
*
DEMH1           XEU         DS=HOST1,DESTTSU='RSDCSU'
*****************************************************************
*    HOST CHANNEL DEFINITION -- TRUNK AND CHANNEL
*
TKCH1P1         TRUNK       PRCSR=PRC1,LSUTYPE=DNS
*
CHANH1          CHANNEL     TRUNK=TKCH1P1,PPID=01D
*
*****************************************************************
*  NETWORK CONSOLE DEFINITIONS
*
NMSC            XEU         NA=1,DESTTSU='DTPX',DESTSSU='EUNMS'
EUNMS           EU          PRCSR=PRC1,TYPE=NMS
OS              XEU         NA=1,DESTSSU='EUOS'
EUOS            EU          PRCSR=PRC1,TYPE=DCPOS,INFO='MONITOR'
*
```

```
*****************************************************************
*    ALLOWS NMS ACCESS THROUGH OS CONSOLE
*
CONSP1          TERM            PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                                DEST=NMSC,ALOC=YES,CEDS=YES,OPDS=YES,;
                                PPID=OFF
*****************************************************************
*    NMS LINE CLASS, LINE, AND TERMINAL DEFINITION
*
UN96SH          LCLASS          LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
*
LNMSP1          LINE            PRCSR=PRC1,CLASS=UN96SH,ADR=04
*
GNMSP1          GROUP           PRCSR=PRC1,LINE=LNMSP1
*
NMSCP1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC,FMT=(24,80),AUTH=PRIV,;
                                ALOC=IMMED
*****************************************************************
*    UNISCOPE TERMINAL (TERMA1) DEFINITION FOR HOST ACCESS.
*     SECOND SCREEN OF NMS CONSOLE TERMINAL.  ($$SON TO TERMA1)
*
*
TERMA1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
*****************************************************************
** LAN PLATFORM CONFIGURATION FOR LLC LAN ATTACHMENT:
*
LANLC           LCLASS          LPH=ILML
LNP111          LINE            PRCSR=PRC1,CLASS=LANLC,ADR=X'11',;
                                STA=X'08000B0C360B'
*****************************************************************
** TCP-IP STACK CONFIGURATION:
*
** EU AND XEU DEFINITIONS FOR TCP-IP STACK:
** (CONFIGURES SERVER AND USER TELNET ON PRC1)
*
EUTCP           EU              PRCSR=PRC1,TYPE=TCPIP
USRTEL          XEU             DS=PRC1,DESTTSU='DTPX',DESTSSU='EUTCP'
*
** SUBNET AND IPADR DEFINITIONS FOR HOST CHANNEL AND LAN ATTACHMENTS:
*
SNCHAP1         SUBNET          PRCSR=PRC1,TYPE=IPCHAN,IPNETID=(132,53),;
                                CHANNEL=CHANH1,IPROUTER=YES
SNLLCP1         SUBNET          PRCSR=PRC1,TYPE=LANLLC,IPNETID=(132,52),;
                                LINE=LNP111,IPROUTER=YES
IPP1H1          IPADR           PRCSR=PRC1,DCAEP=DCP1,;
                                IPADDR1=(132,53,135,1,LOCAL)
IPP1LAN         IPADR           PRCSR=PRC1,DCAEP=DCP1,;
                                IPADDR1=(132,52,135,79,LOCAL)
*****************************************************************
*END OF EXAMPLE
*****************************************************************
```

A.1.3. Configuring a DCP Bridge Node Between a MAC LAN and a Telcon DNS Network

This example shows a DCP functioning as a bridge between a MAC LAN and a Telcon DNS network. The OS 1100 hosts in this network are configured to use the DDN 1100 applications for communication with hosts on the LAN. Figure A-3 depicts this configuration.

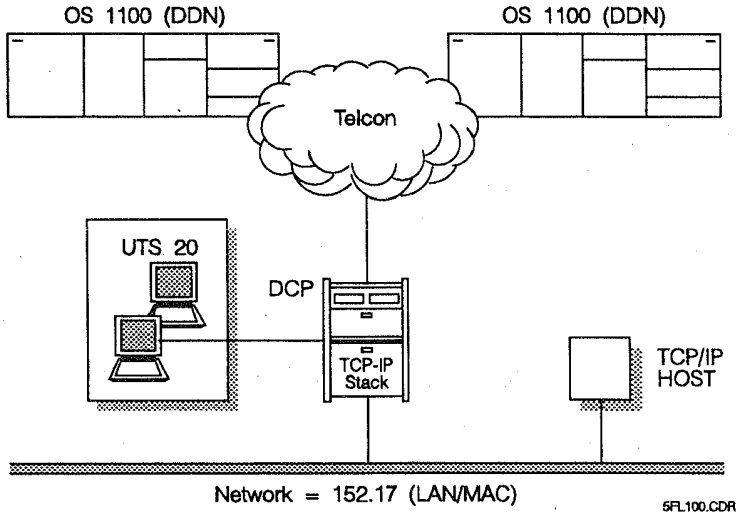


Figure A–3. DCP as a Bridge Between a LAN and a DNS Network

```
. CMS 1100 CONFIGURATION (FIRST HOST)
. DNS CHANNEL ATTACHED HOST (HOST1) RUNNING DDN 1100
.
NODE                    NODE-ADDRESS,10
.
TRACE                   ASG,*TRACE(+1).,F///100
.
STATIC-BANKS            DCP,DNS,DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
PROCESS,DDP             TYPE,XCSU ;
                        ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                        STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT,5000
.
PROCESS,DCP             TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP
.
PROCESS,RSDCSU          TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP
.
```

```
.TRANSPORT,DDNG          TRANSPORT-USER,DDNG ;
                         NODE-ADDRESS,1

.FEP,FEPP1               PATHS,PATH1

.PATH,PATH1              OWNER,FEPP1 ;
                         TYPE,BLOCK ;
                         INPUT-NODE,INODE8 ;
                         OUTPUT-NODE,ONODE8

.RSI                     TIME-OUTS,YES ;
                         BLOCKING-FACTOR,25,4

. CMS 1100 CONFIGURATION (SECOND HOST)
. DNS CHANNEL ATTACHED HOST (HOST2) RUNNING DDN 1100

.NODE                    NODE-ADDRESS,20

.TRACE                   ASG,*TRACE(+1).,F///100

.STATIC-BANKS            DCP,DNS,DEMAND

. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..

.APPLICATION,LOADEU      OWNER,HOSTMS

.PROCESS,DDP             TYPE,XCSU ;
                         ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                         STATE,UP ;
                         SOURCE,MASM A2-INFO,IP$NETD ;
                         INPUT-QUE-LIMIT,5000

.PROCESS,DCP             TYPE,SAI ;
                         SOURCE,MASM ;
                         ENTRY,NCHENT ;
                         STATE,UP

.PROCESS,RSDCSU          TYPE,CSU ;
                         SOURCE,PLUS ;
                         ENTRY,RSIDEMANDCSU ;
                         STATE,UP

.TRANSPORT,DDNG          TRANSPORT-USER,DDNG ;
                         NODE-ADDRESS,2

.FEP,FEPP1               PATHS,PATH1

.PATH,PATH1              OWNER,FEPP1 ;
                         TYPE,BLOCK ;
                         INPUT-NODE,INODE8 ;
                         OUTPUT-NODE,ONODE8

.RSI                     TIME-OUTS,YES ;
                         BLOCKING-FACTOR,25,4


************************************************************
**** TELCON CONFIGURATION
************************************************************<qa>
** DNS ADDRESS DEFINITION FOR REMOTE OS 1100 HOSTS:
*
HOST1          NETADR       NA=10
HOST2          NETADR       NA=20
*
** DCP PROCESSOR DEFINITION:
*
DCP1           NETADR       NA=1
PRC1           PRCSR        NA=1
```

```
*****************************************************************
** DEMAND TERMINAL EXTERNAL END-USER (XEU) DEFINITION:
** (CONFIGURES DEMAND MODE TERMINAL ACCESS TO HOST)
*
DEMH1           XEU             DS=HOST1,DESTTSU='RSDCSU'
DEMH2           XEU             DS=HOST2,DESTTSU='RSDCSU'
*****************************************************************
** XEU DEFINITION FOR DDP TERMINAL ACCESS:
** (CONFIGURES TERMINAL ACCESS TO DDN 1100 APPLICATIONS)
*
DDPH1           XEU             DS=HOST1,DESTTSU='DDP'
DDPH2           XEU             DS=HOST2,DESTTSU='DDP'
*****************************************************************
** XEU DEFINITION FOR DDP INTER-PROCESS CONTROL (IPC):
** (CONFIGURES HOST-TO-HOST ACCESS TO DDN 1100 APPLICATIONS)
*
DDNH1           XEU             DS=HOST1,SRCTSU=IPCTSU,DESTTSU='DDNG'
DDNH2           XEU             DS=HOST2,SRCTSU=IPCTSU,DESTTSU='DDNG'
*****************************************************************
*    HOST CHANNEL DEFINITION -- TRUNK AND CHANNEL TO HOST1 AND HOST2
*
TKCH1P1         TRUNK           PRCSR=PRC1,LSUTYPE=DNS
CHANH1          CHANNEL         TRUNK=TKCH1P1,PPID=0D
*
TKCH2P1         TRUNK           PRCSR=PRC1,LSUTYPE=DNS
CHANH2          CHANNEL         TRUNK=TKCH2P1,PPID=01D
*****************************************************************
*  NETWORK CONSOLE DEFINITIONS
*
NMSC1           XEU             NA=1,DESTTSU='DTPX',DESTSSU='EUNMS1'
EUNMS1          EU              PRCSR=PRC1,TYPE=NMS
OS1             XEU             NA=1,DESTSSU='EUOS1'
NMSC2           XEU             NA=1,DESTTSU='DTPX',DESTSSU='EUNMS2'
OS2             XEU             NA=1,DESTSSU='EUOS2'

*****************************************************************
*    ALLOWS NMS ACCESS THROUGH OS CONSOLE
*
CONSP1          TERM            PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                                DEST=NMSC1,ALOC=YES,CEDS=YES,OPDS=YES,;
                                PPID=OFF
*
*****************************************************************
*    NMS LINE CLASS, LINE, AND TERMINAL DEFINITION
*
UN96SH          LCLASS          LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
*
LNMSP1          LINE            PRCSR=PRC1,CLASS=UN96SH,ADR=04
*
GNMSP1          GROUP           PRCSR=PRC1,LINE=LNMSP1
*
NMSCP1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC1,FMT=(24,80),AUTH=PRIV,;
                                ALOC=IMMED
*
*****************************************************************
*    UNISCOPE TERMINAL (TERMA1) DEFINITION FOR HOST ACCESS.
*     SECOND SCREEN OF NMS CONSOLE TERMINAL.  ($$SON TO TERMA1)
*
*
TRM1P1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
*
*****************************************************************
** LAN PLATFORM CONFIGURATION FOR MAC LAN ATTACHMENT:
*
LANLC           LCLASS          LPH=ILML
LNP10B          LINE            PRCSR=PRC1,CLASS=LANLC,ADR=X'0B',;
                                STA=X'08000B0C360B'
*
```

```
*****************************************************************
** TCP-IP STACK CONFIGURATION :
*
** EU AND XEU DEFINITIONS FOR TCP-IP STACK:
** (CONFIGURES SERVER AND USER TELNET ON PRC1)
*
EUTCP         EU           PRCSR=PRC1,TYPE=TCPIP
USRTEL        XEU          DS=PRC1,DESTTSU='DTPX',DESTSSU='EUTCP'
*
** SUBNET AND IPADR DEFINITIONS FOR LAN ATTACHMENTS:
*
SNMACP1       SUBNET       PRCSR=PRC1,TYPE=LAN,;
                           IPNETID=(152,17),LINE=LNP10B
IPP1DNS1      IPADR        PRCSR=PRC1,DCAEP=DCP1,IPADDR1=(152,17,3,97,LOCAL)
IPP1DNS2      IPADR        PRCSR=PRC1,DCAEP=HOST1,IPADDR1=(152,17,3,98,LOCAL)
IPP1DNS3      IPADR        PRCSR=PRC1,DCAEP=HOST2,IPADDR1=(152,17,3,99,LOCAL)
*
*****************************************************************
*END OF EXAMPLE
*****************************************************************
```

A.1.4. Configuring a DCP Bridge Node Between a PDN and a Channel-Attached Host

This example configures a DCP to function as a bridge between a PDN and a channel-attached host. Figure A-4 depicts this configuration.

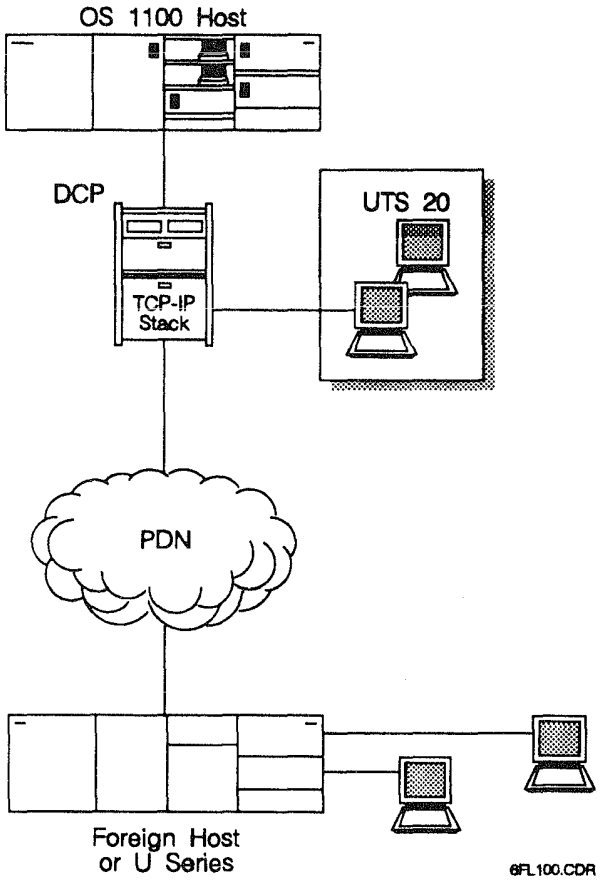


Figure A–4. DCP as a Bridge Node Between a PDN and Channel-Attached Host

```
. CMS 1100 CONFIGURATION
. DNS CHANNEL ATTACHED HOST RUNNING DDN 1100
NODE                    NODE-ADDRESS,10
.
TRACE                   ASG,*TRACE(+1).,F///100
.
STATIC-BANKS            DCP,DNS,DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
PROCESS,DDP             TYPE,XCSU ;
                        ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                        STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT,5000
```

```
PROCESS,DC              TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP
.
PROCESS,RSDCSU          TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP
.
TRANSPORT,DDNG          TRANSPORT-USER,DDNG ;
                        NODE-ADDRESS,1
.
FEP,FEPP1               PATHS,PATH1
.
PATH,PATH1              OWNER,FEPP1 ;
                        TYPE,BLOCK ;
                        INPUT-NODE,INODE8 ;
                        OUTPUT-NODE,ONODE8
.
RSI                     TIME-OUTS,YES ;
                        BLOCKING-FACTOR,25,4


************************************************************
**** TELCON CONFIGURATION
************************************************************
*   NETWORK ADDRESS (NA) DEFINITION
*
*   THE NETADR NA MUST MATCH THE CMS 1100 NODE NETWORK ADDRESS
*   PARAMETER.
*
HOST1         NETADR      NA=10
************************************************************************
*   DCP PROCESSOR DEFINITION AND NA ASSIGNMENT
*
PRC1          PRCSR       NA=1
*
*   HOST TRUNK AND CHANNEL DEFINITION
*
*   WITH DNS, THE CHANNEL TO A HOST IS CONFIGURED AS A LOGICAL TRUNK.
*   LSUTYPE MUST BE DNS.
*
*   EXTERNAL END USER (XEU) DEFINITION, DEMAND TO HOST
*
DEMH1         XEU         DS=HOST1,DESTTSU='RSDCSU'
*
DDNGN         XEU         DS=HOST1,SRCTSU=IPCTSU,DESTTSU='DDNG'
************************************************************
*    HOST CHANNEL DEFINITION -- TRUNK AND CHANNEL
*
TKCH1P1       TRUNK       PRCSR=PRC1,LSUTYPE=DNS
*
CHANH1        CHANNEL     TRUNK=TKCH1P1,PPID=01D
************************************************************
*  NETWORK CONSOLE DEFINITIONS
*
NMSC          XEU         NA=1,DESTTSU='DTPX',DESTSSU='EUNMS'
EUNMS         EU          PRCSR=PRC1,TYPE=NMS
OS            XEU         NA=1,DESTSSU='EUOS'
EUOS          EU          PRCSR=PRC1,TYPE=DCPOS,INFO='MONITOR'
```

```
*****************************************************************
*    ALLOWS NMS ACCESS THROUGH OS CONSOLE
*
CONSP1          TERM            PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                                DEST=NMSC,ALOC=YES,CEDS=YES,OPDS=YES,;
                                PPID=OFF
*****************************************************************
*    NMS LINE CLASS, LINE, AND TERMINAL DEFINITION
*
UN96SH          LCLASS          LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
*
LNMSP1          LINE            PRCSR=PRC1,CLASS=UN96SH,ADR=04
GNMSP1          GROUP           PRCSR=PRC1,LINE=LNMSP1

NMSCP1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC,FMT=(24,80),AUTH=PRIV,;
                                ALOC=IMMED
*****************************************************************
*    UNISCOPE TERMINAL (TERMA1) DEFINITION FOR HOST ACCESS.
*     SECOND SCREEN OF NMS CONSOLE TERMINAL.  ($$SON TO TERMA1)
*
TERMA1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
*
*****************************************************************
*   PSCS (X.25) DEFINITIONS
*
*   THIS PSCS CONFIGURATION SUPPORTS TCP-IP STACK CONNECTIONLESS
*   NETWORK SERVICE OVER THE TELENET PDN.  PSCS REQUIRES LCLASS,
*   X25DEF, PDNGRP AND LINE NDS'S.
*
*   X.25 LINE CLASS DEFINITION
*   (LPH TYPE MUST BE X25PKT)
*
X2596           LCLASS          LPH=X25PKT,SPEED=9600,OPTIONS=(DIR,SYFD)
*
*   TELENET X.25 DEFINITION
*
X25DFTEL        X25DEF          NETWORK=TELENET
*
*   TELENET PDNGRP DEFINITION
*   (DEFINES THE PDN ATTRIBUTES)
*
PGRPTEL         PDNGRP          PRCSR=PRC1,X25DEF=X25DFTEL,;
                                VCGRP=(20,30),DTEADR='31101765432'
*
*   TELENET LINE DEFINITION
*   (DEFINES THE DCP PORT CONNECTION)
*
LNP113      LINE                PDNGRP=PGRPTEL,CLASS=X2596,ADR=X'13'
*****************************************************************
*  TCP-IP DEFINITIONS
*
* TCP-IP EU AND XEU DEFINITION (CONFIGURES SERVER AND USER TELNET)
*
EUTCP           EU              PRCSR=PRC1,TYPE=TCPIP
USRTEL          XEU             DS=PRC1,DESTTSU='DTPX',DESTSSU='EUTCP'
```

```
*
*   TCP-IP SUBNET DEFINITION
*
SNIPPP1       SUBNET       PRCSR=PRC1,IPNETID=(31),TYPE=IPPDN,;
                           PDNGRP=PGRPTEL
*
*   TCP-IP STACK IPADR DEFINITIONS
*
*
*   IPADR DEFINITION FOR PRC1
*
IPP1TEL1      IPADR        PRCSR=PRC1,DCAEP=NMSC,;
                           IPADDR1=(31,4,0,1,LOCAL)
IPP1TEL2      IPADR        PRCSR=PRC1,DCAEP=DDNGN,;
                           IPADDR1=(31,4,0,2,LOCAL)
*
*   IPADR DEFINITION FOR THE FOREIGN HOST
*   (OBTAIN THE DTE ADDRESS FROM THE PDN ADMINISTRATOR)
*
IPP1DMH1      IPADR        PRCSR=PRC1,;
                           IPADDR1=(31,5,0,1),;
                           DTEADR='31101234567',NAME1='FHOST'
*****************************************************************
*END OF EXAMPLE
*****************************************************************
```

A.1.5. DCP-to-DCP Trunk Using DNS Over the DDN

This example configures a DCP-to-DCP trunk using PSCS, TCP-IP Stack, and DNS. It is illustrated in Figures A-5.

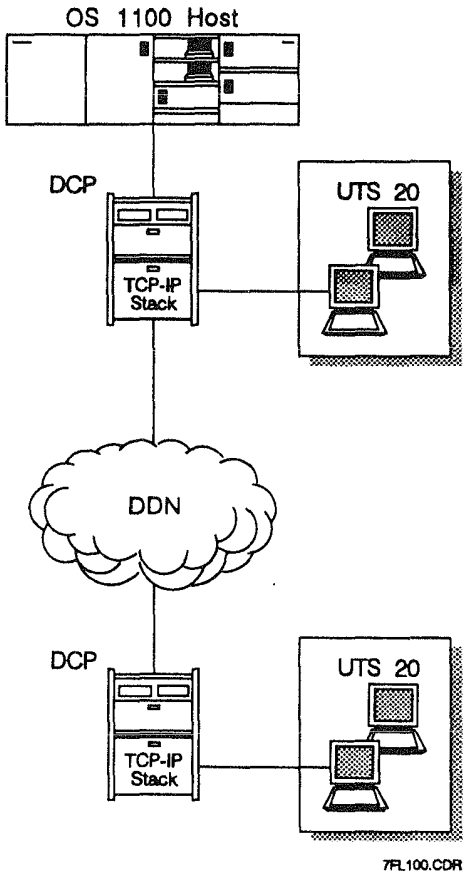


Figure A-5. DCP-to-DCP Trunk over the DDN

```
. CMS 1100 CONFIGURATION
. DNS CHANNEL ATTACHED HOST RUNNING DCA APPLICATIONS
.
NODE                    NODE-ADDRESS,10
.
TRACE                   ASG,*TRACE(+1).,F///100
.
STATIC-BANKS            DCP,DNS,DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
PROCESS,DDP             TYPE,XCSU ;
                        ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                        STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT,5000
```

```
PROCESS,DCP             TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP
.
PROCESS,RSDCSU          TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP
.
FEP,FEPP1               PATHS,PATH1
.
PATH,PATH1              OWNER,FEPP1 ;
                        TYPE,BLOCK ;
                        INPUT-NODE,INODE8 ;
                        OUTPUT-NODE,ONODE8
.
RSI                     TIME-OUTS,YES ;
                        BLOCKING-FACTOR,25,4

*****************************************************************
* TELCON CONFIGURATION
*****************************************************************
*   NETWORK ADDRESS FOR THE OS 1100 HOST
*
*   NETWORK ADDRESSES IN THE DNS ENVIRONMENT ARE ASSIGNED BY THE
*   TELCON/HOST NETWORK ADMINISTRATOR.
*
*   THE HOST NA MUST MATCH THE NETWORK ADDRESS PARAMETER IN THE
*   CMS 1100 NODE NDS.
*
HOST1           NETADR      NA=10
*****************************************************************
*   DCP PROCESSOR DEFINITIONS AND NA ASSIGNMENTS
*
PRC1            PRCSR       NA=1
*
PRC2            PRCSR       NA=2
*
*****************************************************************
*   EXTERNAL END USER (XEU) DEFINITION, DEMAND TO HOST
*
DEMH1     XEU               DS=HOST1,DESTTSU='RSDCSU'
*
*****************************************************************
*    HOST CHANNEL DEFINITION -- TRUNK AND CHANNEL
*
TKCH1P1         TRUNK       PRCSR=PRC1,LSUTYPE=DNS
*
CHANH1          CHANNEL     TRUNK=TKCH1P1,PPID=01D
*
*****************************************************************
*  NETWORK CONSOLE DEFINITIONS
*
NMSC1           XEU         NA=1,DESTTSU='DTPX',DESTSSU='EUNMS1'
EUNMS1          EU          PRCSR=PRC1,TYPE=NMS
OS1             XEU         NA=1,DESTSSU='EUOS1'
EUOS2           EU          PRCSR=PRC2,TYPE=DCPOS,INFO='MONITOR'
NMSC2           XEU         NA=1,DESTTSU='DTPX',DESTSSU='EUNMS2'
EUNMS2          EU          PRCSR=PRC2,TYPE=NMS
OS2             XEU         NA=1,DESTSSU='EUOS2'
*
*****************************************************************
*    ALLOWS NMS ACCESS THROUGH OS CONSOLE
*
CONSP1          TERM        PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                            DEST=NMSC1,ALOC=YES,CEDS=YES,OPDS=YES,;
                            PPID=OFF
*
```

```
CONSP2          TERM            PRCSR=PRC2,TYPE=OPERATOR,AUTH=PRIV,;
                                DEST=NMSC2,ALOC=YES,CEDS=YES,OPDS=YES,;
                                PPID=OFF
*************************************************************
*    NMS LINE CLASS, LINE, AND TERMINAL DEFINITION
*
UN96SH          LCLASS          LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
*
LNMSP1          LINE            PRCSR=PRC1,CLASS=UN96SH,ADR=04
*
GNMSP1          GROUP           PRCSR=PRC1,LINE=LNMSP1
*
NMSCP1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC1,FMT=(24,80),AUTH=PRIV,;
                                ALOC=IMMED
*
LNMSP2          LINE            PRCSR=PRC2,CLASS=UN96SH,ADR=04
*
GNMSP2          GROUP           PRCSR=PRC2,LINE=LNMSP2
*
NMSCP2          TERM            GROUP=GNMSP2,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC2,FMT=(24,80),AUTH=PRIV,;
                                ALOC=IMMED
*************************************************************
*    UNISCOPE TERMINAL (TERMA1) DEFINITION FOR HOST ACCESS.
*     SECOND SCREEN OF NMS CONSOLE TERMINAL.  ($$SON TO TERMA1)
*
TRM1P1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
TRM2P2          TERM            GROUP=GNMSP2,TYPE=UTS20,ADR=(X'21',X'64')
*************************************************************
*   TRUNK DEFINITION
*
*   DEFINING AN X.25 TRUNK REQUIRES TRUNK, LCLASS, PDNGRP, X25DEF,
*   AND LINE NDS'S.
*
TKDP1P2         TRUNK           PRCSR=(PRC1,PRC2),LSUTYPE=DNS
*
*   X.25 LINE CLASS DEFINITION
*   (LPH TYPE MUST BE X25PKT)
*
X2596           LCLASS          LPH=X25PKT,SPEED=9600,OPTIONS=(DIR,SYFD)
*
*   DDN X.25 DEFINITION
*
X25DFDDN        X25DEF          NETWORK=DDNX25
*
*   PDNGRP AND LINE DEFINITION FOR FEP (PRC1)
*
PGRPDDN1        PDNGRP          PRCSR=PRC1,X25DEF=X25DFDDN,;
                                VCGRP=(20,30,SVC)
*
LNP113          LINE            PDNGRP=PGRPDDN1,CLASS=X2596,ADR=X'13'
*   PDNGRP AND LINE DEFINITION FOR PRC2
*
PGRPDDN2        PDNGRP          PRCSR=PRC2,X25DEF=X25DFDDN,;
                                VCGRP=(20,30,SVC)
*
LNP222          LINE            PDNGRP=PGRPDDN2,CLASS=X2596,ADR=X'22'
*************************************************************
*   TCP-IP STACK DEFINITIONS
*
EUTCP1          EU              PRCSR=PRC1,TYPE=TCPIP
*
EUTCP2          EU              PRCSR=PRC2,TYPE=TCPIP
*
*   TCP-IP SUBNET DEFINITION
*
*   SUBNET FOR PRC1
*
```

```
SNDDNP1        SUBNET       PRCSR=PRC1,IPNETID=(31),TYPE=DDN,;
                            PDNGRP=PGRPDDN1
*
*   SUBNET FOR PRC2
*
SNDDNP2        SUBNET       PRCSR=PRC2,IPNETID=(31),TYPE=DDN,;
                            PDNGRP=PGRPDDN2
*   TCP-IP IPADR DEFINITIONS
*
*   IPADR DEFINITION FOR PRC1
*
IPP1TK12       IPADR        PRCSR=PRC1,DCAEP=TKDP1P2,;
                            IPADDR1=(31,4,0,1,LOCAL),;
                            IPADDR2=(31,7,0,2)
*
*   IPADR DEFINITION FOR PRC2
*
IPP2TK12       IPADR        PRCSR=PRC2,DCAEP=TKDP1P2,;
                            IPADDR1=(31,4,0,1),;
                            IPADDR2=(31,7,0,2,LOCAL)
*
*****************************************************************
*END OF EXAMPLE
*****************************************************************
```

A.2. Telcon TS/TN Configurations

The following examples use termination system/transport network (TS/TN) facilities, which provide connection-oriented network layer services for DCA Telcon networks.

A.2.1. Configuring a DCP Bridge Node Between the DDN and a Channel-Attached Host

This example configures a DCP as a bridge node between an OS 1100 host and a TCP/IP host across the DDN. TS/TN is used at the network layer over the host channel. DDN applications (FTP and SMTP) can be used with this configuration between the OS 1100 host running DDN 1100 and the TCP/IP host.

Server TELNET allows terminals on the TCP/IP host to access the OS 1100 host. User TELNET allows terminals on the DCP to access TCP/IP hosts. This configuration is illustrated in Figure A-6.

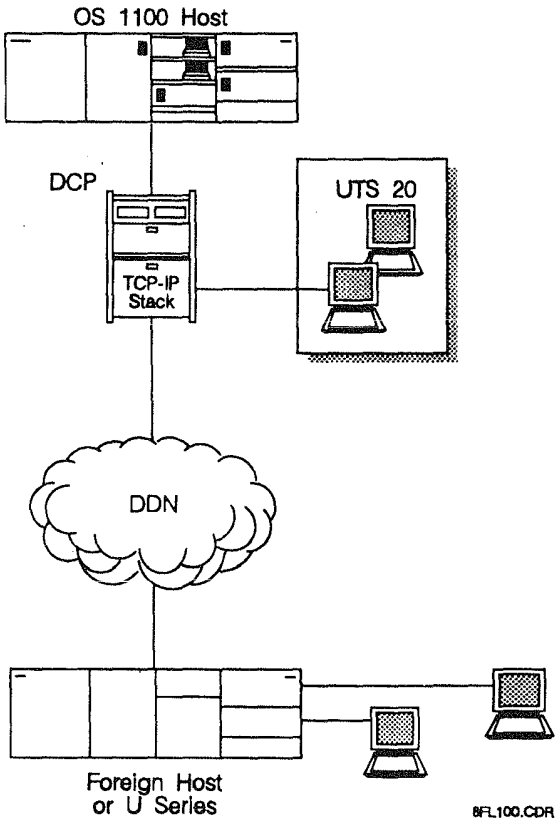


Figure A-6. DCP as a Bridge Between the DDN and a Channel-Attached Host

```
. CMS 1100 CONFIGURATION
. TS/TN CHANNEL ATTACHED HOST RUNNING DDN 1100
.
ADMIN                   SECURITY,PASSWORD,PSWD ;
                        KEYIN-NAME,CMS
.
TRACE                   ASG,*TRACE(+1).,F///100
.
STATIC-BANKS            DCP,TS/TN,DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
PROCESS,DDP             TYPE,XCSU ;
                        ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                        STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT,5000
.
PROCESS,DCP             TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP
.
PROCESS,RSDCSU          TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP
.
TRANSPORT,DDNG          TRANSPORT-USER,DDNG ;
                        NETWORK,NET10
.
NETWORK,NET10           SAI-PROCESS,FEPP1 ;
                        NETWORK-NUMBER,10
.
FEP,FEPP1               PATHS,PATH1
.
PATH,PATH1              OWNER,FEPP1 ;
                        TYPE,BLOCK ;
                        INPUT-NODE,INODE8 ;
                        OUTPUT-NODE,ONODE8
.
RSI                     TIME-OUTS,YES ;
                        BLOCKING-FACTOR,25,4


*****************************************************************
**** TELCON CONFIGURATION
*****************************************************************
*    DCP TO U SERIES OR FOREIGN HOST USING TCP-IP AND
*    SERVER TELNET OVER A DDN
*****************************************************************
*   DCP PROCESSOR DEFINITION
*
PRC1          PRCSR
*****************************************************************
*   TERMINATION SYSTEM DEFINITIONS
*
HOST1         DCATS       PRCSR=PRC1                    * OS 1100 HOST
*
DCPTS1        DCPTS       PRCSR=PRC1                    * DCP
*
*****************************************************************
*   SESSION DEFINITION
*
SSP1P1        SESSN       TS1=(DCPTS1,10,PRC1),TS2=(DCPTS1,11,PRC1)
*
SSP1H1        SESSN       TS1=(HOST1,10,PRC1),TS2=(DCPTS1,12,PRC1)
```

```
*****************************************************************
*   EXTERNAL END USER (XEU) DEFINITION, DEMAND
*
DEMH1          XEU          TS=HOST1,DESTTSU='RSDCSU'
*
*   EXTERNAL END USER (XEU) DEFINITION, DDN 1100 (TCP/IP) APPLICATIONS
*
DDNH1          XEU          TS=HOST1,SRCTSU=IPCTSU,DESTTSU='DDNG'
*****************************************************************
*    HOST CHANNEL DEFINITION
*
CHANH1         CHANNEL      TS=HOST1,PPID=01D
*****************************************************************
*  NETWORK CONSOLE DEFINITIONS
*
NMSC           XEU          TS=DCPTS1,DESTTSU='DTPX',DESTSSU='EUNMS'
EUNMS          EU           PRCSR=PRC1,TYPE=NMS
OS             XEU          TS=DCPTS1,DESTSSU='EUOS'
EUOS           EU           PRCSR=PRC1,TYPE=DCPOS,INFO='MONITOR'
*****************************************************************
*    ALLOWS NMS ACCESS THROUGH OS CONSOLE
*
CONSP1         TERM         PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                            DEST=NMSC,ALOC=YES,CEDS=YES,OPDS=YES,;
                            PPID=0FF
*
*****************************************************************
*    NMS LINE CLASS, LINE, AND TERMINAL DEFINITION
*
UN96SH         LCLASS       LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
LNMSP1         LINE         PRCSR=PRC1,CLASS=UN96SH,ADR=04
*
GNMSP1         GROUP        PRCSR=PRC1,LINE=LNMSP1
*
NMSCP1         TERM         GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                            DEST=NMSC,FMT=(24,80),AUTH=PRIV,;
                            ALOC=IMMED
*****************************************************************
*    UNISCOPE TERMINAL (TERMA1) DEFINITION FOR HOST ACCESS.
*     SECOND SCREEN OF NMS CONSOLE TERMINAL.  ($$SON TO TERMA1)
*
TERMA1         TERM         GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
*****************************************************************
*   PSCS (X.25) DEFINITIONS
*
*
*   X.25 LINE CLASS DEFINITION
*
X2596          LCLASS       LPH=X25PKT,SPEED=9600,OPTIONS=(DIR,SYFD)
*
*   DDN X.25 DEFINITION
*
X25DFDDN       X25DEF       NETWORK=DDNX25
*
*   X.25 PDNGRP DEFINITION
*   (DEFINES THE DDN ATTRIBUTES)
*
PGRPDDN        PDNGRP       PRCSR=PRC1,X25DEF=X25DFDDN,;
                            VCGRP=(20,30)
*
*   X.25 LINE DEFINITION
*   (DEFINES THE DCP PORT CONNECTION)
*
LNP113         LINE         PDNGRP=PGRPDDN,CLASS=X2596,ADR=X'13'
*****************************************************************
```

```
*   TCP-IP STACK DEFINITIONS
*
*
EUTCP          EU            PRCSR=PRC1,TYPE=TCPIP
USRTEL         XEU           TS=DCPTS1,DESTTSU='DTPX',DESTSSU='EUTCP'
*
*   TCP-IP SUBNET DEFINITION
*
*
SNDDNP1        SUBNET        PRCSR=PRC1,IPNETID=(31),TYPE=DDN,;
                             PDNGRP=PGRPDDN
*
*   TCP-IP IPADR DEFINITION
*
IPP1H1         IPADR         PRCSR=PRC1,DCAEP=HOST1,;
                             IPADDR1=(31,4,0,1,LOCAL)
*****************************************************************
*END OF EXAMPLE
*****************************************************************
```

A.2.2. Configuring a DCP Bridge Between the DDN and a TS/TN Network

This example configures a DCA data link between a DCP in a TS/TN network and a U Series system across the DDN, enabling the U Series system to operate as part of the DCA network. Terminals on both DCPs can access distributed data processing applications on both the OS 1100 host and the U Series system. Terminals on the U Series system can access applications on the OS 1100. Figure A–7 depicts this configuration.

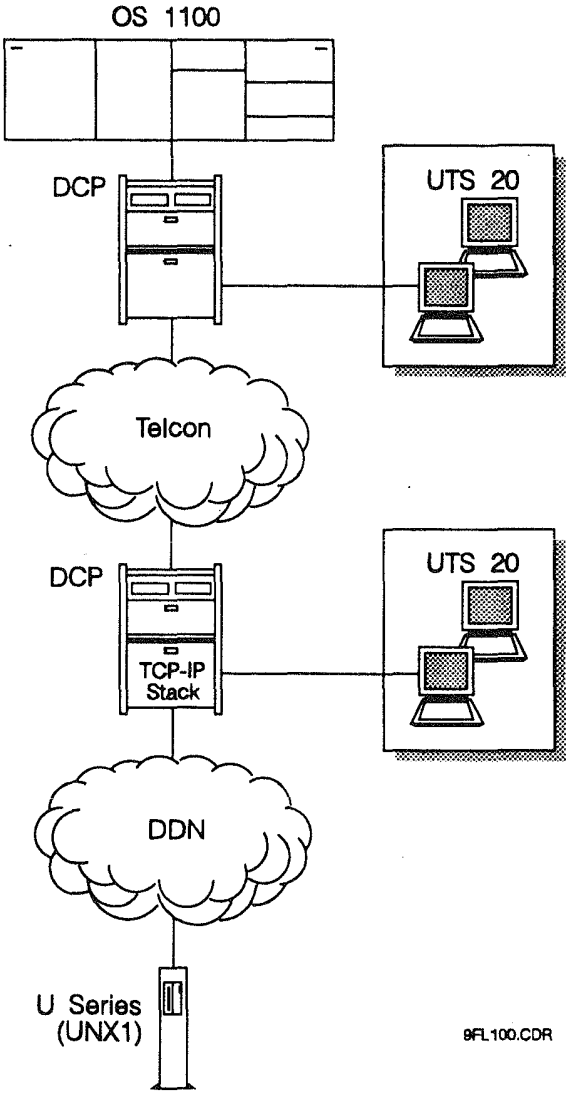


Figure A–7. DCP Bridge Between the DDN and a TS/TN Network

```
. CMS 1100 CONFIGURATION
. TS/TN CHANNEL ATTACHED HOST RUNNING DDP-PPC APPLICATIONS
.
ADMIN                   SECURITY,PASSWORD,PSWD ;
                        KEYIN-NAME,CMS
.
TRACE                   ASG,*TRACE(+1).,F///100
.
STATIC-BANKS            DCP,TS/TN,DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
PROCESS,DDP             TYPE,XCSU ;
                        ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                        STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT,5000
.
PROCESS,DCP             TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP
.
PROCESS,RSDCSU          TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP
.
NETWORK,NET10           SAI-PROCESS,FEPP1 ;
                        NETWORK-NUMBER,10
NETWORK,NET11           SAI-PROCESS,FEPP1
                        NETWORK-NUMBER,11
NETWORK,NET12           SAI-PROCESS,FEPP1
                        NETWORK-NUMBER,12
.
FEP,FEPP1               PATHS,PATH1
.
PATH,PATH1              OWNER,FEPP1 ;
                        TYPE,BLOCK ;
                        INPUT-NODE,INODE8 ;
                        OUTPUT-NODE,ONODE8
.
RSI                     TIME-OUTS,YES ;
                        BLOCKING-FACTOR,25,4
```

```
*********************************************************************
**** TELCON CONFIGURATION
*********************************************************************
** DCP PROCESSOR DEFINITION
*
PRC1          PRCSR
PRC2          PRCSR
*
** REMOTE DCA TERMINATION SYSTEM (DCATS )DEFINITIONS:
*
HOST1         DCATS        PRCSR=PRC1
UNIX1         DCATS        PRCSR=PRC2
*********************************************************************
*    HOST CHANNEL DEFINITION
*
CHANH1        CHANNEL      TS=HOST1,PPID=01D
*********************************************************************
```

```
*****************************************************************
** DCP TERMINATION SYSTEM (DCPTS) DEFINITION
*
DCPTS1          DCPTS           PRCSR=PRC1
DCPTS2          DCPTS           PRCSR=PRC2
*****************************************************************
** TERMINAL-TO-HOST AND HOST-TO-HOST SESSION DEFINITION:
*
SSP2H1          SESSN           TS1=(HOST1,12,PRC1),TS2=(DCPTS2,14,PRC2)
SSP1H1          SESSN           TS1=(HOST1,10,PRC1),TS2=(DCPTS1,10,PRC1)
SSP2H2          SESSN           TS1=(UNIX1,10,PRC2),TS2=(DCPTS2,10,PRC2)
SSH1U1          SESSN           TS1=(HOST1,11,PRC1),TS2=(UNIX1,11,PRC2)
*****************************************************************
*    EXTERNAL END-USER (XEU) DEFINITION,DEMAND TO HOST
*
DEMH1           XEU             TS=HOST1,DESTTSU='RSDCSU'
*****************************************************************
*      NETWORK CONSOLE DEFINITIONS
*          (TELCON NMS CONSOLES)
*
*     INTERNAL NMS SESSION REQUIRED FOR EACH DCP
*
SSP1P1          SESSN           TS1=(DCPTS1,11,PRC1),TS2=(DCPTS1,12,PRC1)
SSP2P2          SESSN           TS1=(DCPTS2,11,PRC2),TS2=(DCPTS2,12,PRC2)
SSP1P2          SESSN           TS1=(DCPTS1,13,PRC1),TS2=(DCPTS2,13,PRC2)
*****************************************************************
*     NMS XEU AND EU DEFINITIONS FOR EACH DCP
*
NMSC1           XEU             TS=DCPTS1,DESTTSU='DTPX',DESTSSU='EUNMS1'
EUNMS1          EU              PRCSR=PRC1,TYPE=NMS
NMSC2           XEU             TS=DCPTS2,DESTTSU='DTPX',DESTSSU='EUNMS2'
EUNMS2          EU              PRCSR=PRC2,TYPE=NMS
*****************************************************************
*     DCPOS VIRTUAL WORKSTATION DEFINITIONS FOR EACH DCP
*
OS1             XEU             TS=DCPTS1,DESTSSU='EUOS1'
OS2             XEU             TS=DCPTS1,DESTSSU='EUOS2'
*
EUOS1           EU              PRCSR=PRC1,TYPE=DCPOS,INFO='MONITOR'
EUOS2           EU              PRCSR=PRC2,TYPE=DCPOS,INFO='MONITOR'
*
CONSP1          TERM            PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                                DEST=NMSC1,ALOC=YES,CEDS=YES,OPDS=YES,;
                                PPID=OFF
CONSP2          TERM            PRCSR=PRC2,TYPE=OPERATOR,AUTH=PRIV,;
                                DEST=NMSC2,ALOC=YES,CEDS=YES,OPDS=YES,;
                                PPID=OFF
*****************************************************************
*     NMS CONSOLE LINE AND TERMINAL DEFINITION
*
UN96SH          LCLASS          LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
*
*     NMS CONSOLE FOR FEP (PRC1)
*
LNMSP1          LINE            PRCSR=PRC1,CLASS=UN96SH,ADR=04
GNMSP1          GROUP           PRCSR=PRC1,LINE=LNMSP1
NMSCP1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC1,AUTH=PRIV,ALOC=IMMED
*
*     NMS CONSOLE FOR PRC2
*
LNMSP2          LINE            PRCSR=PRC2,CLASS=UN96SH,ADR=01
GNMSP2          GROUP           PRCSR=PRC2,LINE=LNMSP2
NMSCP2          TERM            GROUP=GNMSP2,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC2,AUTH=PRIV,ALOC=IMMED
*
```

```
*
*     UNISCOPE TERMINAL DEFINITION FOR HOST ACCESS
*        (SECOND SCREEN OF NMS CONSOLE TERMINAL)
*        ($$SON TO TRMP1/P2,$$OPEN TO DEMH1)
*
TRM1P1        TERM         GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
*
TRM1P2        TERM         GROUP=GNMSP2,TYPE=UTS20,ADR=(X'21',X'64')
*****************************************************************
UDLC192       LCLASS       LPH=UDLCL,SPEED=19200,OPTIONS=(DIR,SYFD)
TKUP1P2       TRUNK        PRCSR=(PRC1,PRC2)
LNP110        LINE         TRUNK=TKUP1P2,CLASS=UDLC192,ADR=010
LNP210        LINE         TRUNK=TKUP1P2,CLASS=UDLC192,ADR=011
STUP1P2       STATION      LINE=LNP110,RSHLE=TKUP1P2,LSA=1,RSA=2
STUP2P1       STATION      LINE=LNP210,RSHLE=TKUP1P2,LSA=2,RSA=1
*****************************************************************
** PSCS CONFIGURATION FOR DDN X.25 ATTACHMENT
*
X25DFDN       X25DEF       NETWORK=DDNX25
PGPP2DN       PDNGRP       PRCSR=PRC2,X25DEF=X25DFDN,VCGRP=(1,50),;
                           DTEADR='000000154200'
X2596         LCLASS       LPH=X25PKT,SPEED=9600,OPTIONS=(DIR,SYFD)
LNP212        LINE         PDNGRP=PGPP2DN,CLASS=X2596,ADR=X'12'
*****************************************************************
** TCP-IP STACK CONFIGURATION
*
EUTCP         EU           PRCSR=PRC2,TYPE=TCPIP
SNDDNP2       SUBNET       PRCSR=PRC2,TYPE=DDN,IPNETID=26,;
                           PDNGRP=PGPP2DN
IPP2U1        IPADR        PRCSR=PRC2,DCAEP=UNIX1,;
                           IPADDR1=(26,42,0,15,LOCAL),;
                           IPADDR2=(26,13,0,5)
*
**IPADDR1=LOCAL IP ADDRESS ASSIGNED TO HOST1 (ACROSS THE TELCON NETWORK)
**IPADDR2=REMOTE IP ADDRESS OF UNIX1 (ACROSS THE DDN NETWORK)
*****************************************************************
*END OF EXAMPLE
*****************************************************************
```

A.2.3. Configuring a DCP to Link DCA Termination Systems Across the DDN

This example configures a DCA data link from a DCP to a U Series system across the DDN. The U Series system requires the appropriate Information Services (IS) product to support DCA communications. This allows the U Series system to operate as part of a DCA distributed data processing network. The U Series system is configured as a remote DCA termination system (DCATS). The network configuration is illustrated in Figure A-8.

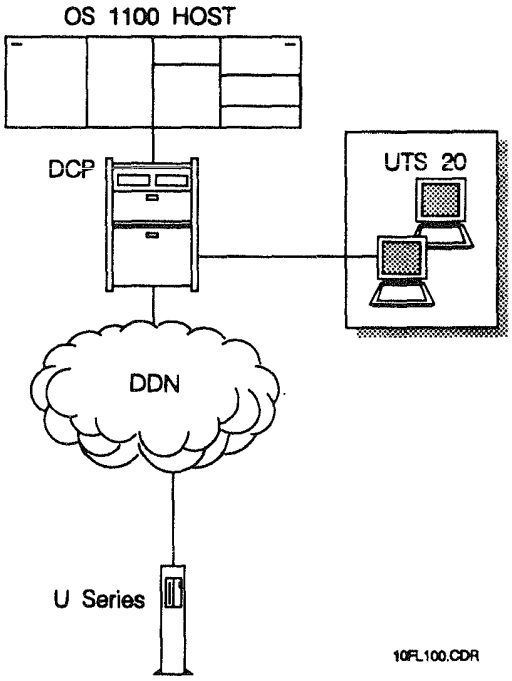


Figure A–8. DCP-to-U Series Using PSCS, TCP-IP Stack, and TS/TN

```
. CMS 1100 CONFIGURATION
. TS/TN CHANNEL ATTACHED HOST RUNNING DDP-PPC APPLICATIONS
.
ADMIN                   SECURITY,PASSWORD,PSWD ;
                        KEYIN-NAME,CMS
.
TRACE                   ASG,*TRACE(+1).,F///100
.
STATIC-BANKS            DCP,TS/TN,DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
PROCESS,DDP             TYPE,XCSU ;
                        ENTRY,IPFCSU,IP$CSU,SYS$LIB$*DDP-PPC-1.CBEP$$DDP ;
                        STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT,5000
```

```
.
PROCESS,DCP             TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP
.
PROCESS,RSDCSU          TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP
.
NETWORK,NET10           SAI-PROCESS,FEPP1 ;
                        NETWORK-NUMBER,10
NETWORK,NET11           SAI-PROCESS,FEPP1
                        NETWORK-NUMBER,11
.
FEP,FEPP1               PATHS,PATH1
.
PATH,PATH1              OWNER,FEPP1 ;
                        TYPE,BLOCK ;
                        INPUT-NODE,INODE8 ;
                        OUTPUT-NODE,ONODE8
.
RSI                     TIME-OUTS,YES ;
                        BLOCKING-FACTOR,25,4
*****************************************************************
**** TELCON CONFIGURATION
*   DCP PROCESSOR DEFINITION
*
PRC1          PRCSR
*
*   TERMINATION SYSTEM DEFINITIONS
*
HOST1         DCATS       PRCSR=PRC1                    * OS 1100 HOST
UNIX1         DCATS       PRCSR=PRC1                    * U SERIES
DCPTS1        DCPTS       PRCSR=PRC1                    * DCP
*****************************************************************
******
*   SESSIONS
*   NETWORK NMS SESSION
*
SSP1P1        SESSN       TS1=(DCPTS1,10,PRC1),TS2=(DCPTS1,11,PRC1)
*
*   DCP TO OS1100, DEMAND
*
SSH1P1        SESSN       TS1=(HOST1,10,PRC1),TS2=(DCPTS1,12,PRC1)
*
*   U SERIES TO 1100 HOST
*
*   THE LOGICAL CHANNEL NUMBER ASSOCIATED WITH THE U SERIES TERMINATION
*   SYSTEM (IN THIS CASE, 10) MUST MATCH THE LOGICAL CHANNEL NUMBER IN
*    THE U SERIES CONFIGURATION.
*
SSH1U1      SESSN           TS1=(HOST1,11,PRC1),TS2=(UNIX1,10,PRC1)
*****************************************************************
*   EXTERNAL END USER (XEU) DEFINITION, DEMAND
*
DEMH1         XEU         TS=HOST1,DESTTSU='RSDCSU'
*****************************************************************
*     HOST CHANNEL DEFINITION
*
CHANH1        CHANNEL     TS=HOST1,PPID=01D
*****************************************************************
*     NETWORK CONSOLE DEFINITIONS
*
NMSC          XEU         TS=DCPTS1,DESTTSU='DTPX',DESTSSU='EUNMS'
EUNMS         EU          PRCSR=PRC1,TYPE=NMS
OS            XEU         TS=DCPTS1,DESTSSU='EUOS'
EUOS          EU          PRCSR=PRC1,TYPE=DCPOS,INFO='MONITOR'
```

```
*****************************************************************
*     ALLOWS NMS ACCESS THROUGH OS CONSOLE
*
CONSP1          TERM            PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                                DEST=NMSC,ALOC=YES,CEDS=YES,OPDS=YES,;
                                PPID=OFF
*****************************************************************
*    NMS LINE CLASS, LINE, AND TERMINAL DEFINITION
*
UN96SH          LCLASS          LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
*
LNMSP1          LINE            PRCSR=PRC1,CLASS=UN96SH,ADR=04
*
GNMSP1          GROUP           PRCSR=PRC1,LINE=LNMSP1
*
NMSCP1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC,FMT=(24,80),AUTH=PRIV,;
                                ALOC=IMMED
*****************************************************************
*    UNISCOPE TERMINAL (TERMA1) DEFINITION FOR HOST ACCESS.
*     SECOND SCREEN OF NMS CONSOLE TERMINAL.  ($$SON TO TERMA1)
*
*
TERMA1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
*****************************************************************
*   X.25 DEFINITION FOR A DDN CONNECTION
*
*
X2596           LCLASS          LPH=X25PKT,SPEED=9600,OPTIONS=(DIR,SYFD)
*
*   DDN X.25 DEFINITION·
*
X25DFDDN        X25DEF          NETWORK=DDNX25,N1=256,L2OPT=16
*
*   PDNGRP AND LINE DEFINITION
*
PGRPDDN         PDNGRP          PRCSR=PRC1,X25DEF=X25DFDDN,;
                                VCGRP=(20,30)
*
LNP113          LINE            PDNGRP=PGRPDDN,CLASS=X2596,ADR=X'13'
*****************************************************************
```

```
******
*   TCP-IP STACK DEFINITIONS
*
*
EUTCP          EU            PRCSR=PRC1,TYPE=TCPIP
*
*   TCP-IP SUBNET DEFINITION
*
*
SNDDNP1        SUBNET        PRCSR=PRC1,IPNETID=(53),TYPE=DDN,;
                             LINE=LNP113
*
*   TCP-IP IPADR DEFINITIONS
*
*
IPP1U1         IPADR         PRCSR=PRC1,DCAEP=UNIX1,;
                             IPADDR1=(53,4,0,1,LOCAL),;
                             IPADDR2=(53,5,0,1)
*********************************************************************
*END OF EXAMPLE
*********************************************************************
```

A.3. Configuring the DCP as an IP Router Between an 802.3 LAN and an FDDI LAN

This example shows a DCP configured as an IP router between an 802.3 LAN, an 802.5 LAN, and an FDDI LAN. The DCP is connected to the 802.3 LAN using an 802.3 ILM-40; it is connected to the 802.5 LAN using an 802.5 ILM-40; and it is connected to the FDDI LAN using an FDDI ILM-40. An OS1100 host running DDN 1100 is attached to the 802.3 LAN through a Host LAN Controller (HLC). Figure A–9 depicts this configuration.

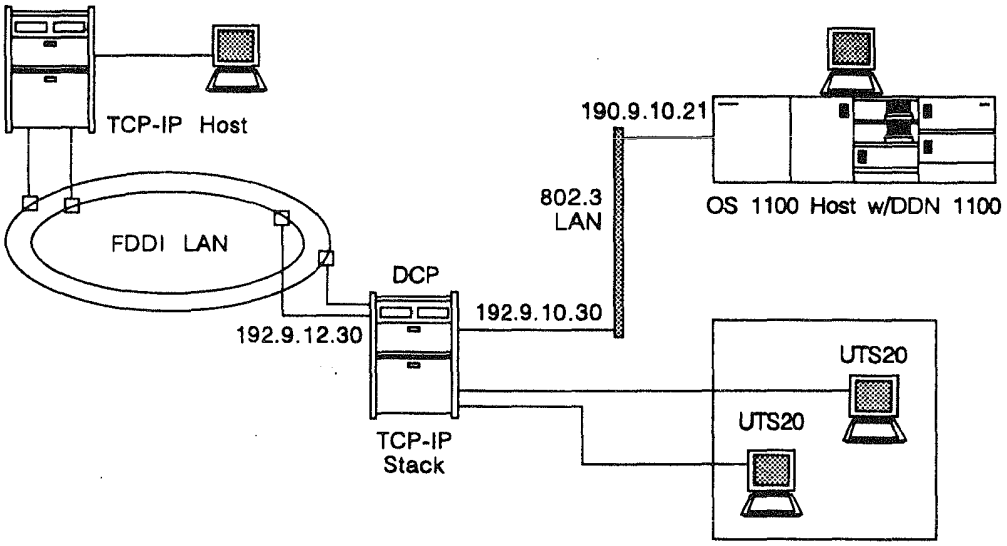


Figure A–9. DCP as an IP Router Between an 802.3 LAN, and an FDDI LAN

```
. CMS 1100 CONFIGURATION
. HLC ATTACHED HOST RUNNING DDN 1100 WITH CMS 1100 TCP/IP
.
NODE                    NODE-ADDRESS,10
.
TRACE                   ASG,*TRACK(+1).,F///1000
.
STATIC-BANKS            TCP/IP, DEMAND
.
. DEMAND AND CSACSU APPLICATIONS DO NOT REQUIRE APPLICATION STATEMENTS..
.
APPLICATION,LOADEU      OWNER,HOSTMS
.
REM-HOSTS               FILE,CMS*TCPIP.RHOSTS,RFC952
.
```

```
PROCESS,DPP
                        TYPE,XCSU  ;
                        ENTRY,IPFCSU,IP$CSU,IP$CSU,SYS$LIB$*DDP-PC-
                        1.CBEP$$DDP ;STATE,UP ;
                        SOURCE,MASM A2-INFO,IP$NETD ;
                        INPUT-QUE-LIMIT, 5000 ;
                        INTERNET-ADR,INTH1

PROCESS,DCP             TYPE,SAI ;
                        SOURCE,MASM ;
                        ENTRY,NCHENT ;
                        STATE,UP

PROCESSS,RSDCSU         TYPE,CSU ;
                        SOURCE,PLUS ;
                        ENTRY,RSIDEMANDCSU ;
                        STATE,UP

INTERNET-ADR,INTH1      IP,IP1,192.9.10.21
                        IP,IP1LINK-LAYER,LAN1 ;
                        GATEWAY,192.9.10.30 ;
                        INTERFACE-TYPE,IEEE802.3

LAN,LAN1                INPUT-NODE,LAN1I ;
                        OUTPUT-NODE,LAN10 ;
                        LOCAL-ADR,X08000BE00C0A ;
                        STATUS, UP

NEIGHBOR,PRC1           LAN,LAN1 EHTERNET-ADR,X08000B0C360B ;
                        LOAD,TEL*LOAD,PRC1

*****************************************************************
* TELCON CONFIGURATION
*****************************************************************
** OS 1100 HOST (ATTACHED TO LAN BY HLC) DNS ADDRESS DEFINITION:
*
HOST1           NETADR          NA=10
*****************************************************************
** DCP PROCESSOR DEFINITION:
*
DCP1            NETADR          NA=1
PRC1            PRCSR           NA=1
*****************************************************************
** DEMAND TERMINAL EXTERNAL END-USER (XEU) DEFINITION:
** (CONFIGURES DEMAND MODE TERMINAL ACCESS TO HOST)
*
DEMH1           XEU             DS=HOST1,DESTTSU-'RSDCSU'
*****************************************************************
** XEU DEFINITION FOR DDP TERMINAL ACCESS:
** (CONFIGURES TERMINAL ACCESS TO DDN 1100 APPLICATION)
*
DDPH1           XEU             DS-HOST1,DESTTSU='DDP'
*****************************************************************
* NETWORK CONSOLE DEFINTIONS
*
NMSC            XEU             NA-1K,
                                DESTTSU='DTPX',DESTSSU='EUNMS'
EUNMS           EU              PRCSR=PRC1,TYPE=NMS
OS              XEU             NA=1,DESTSSU='EUOS'
EUOS            EU              PRCSR=PRC1,TYPE=DCPOS,INFO='MONITOR'
*
*****************************************************************
*  ALLOWS NMS ACCESS THROUGH OS CONSOLE
*
CONSP1          TERM            PRCSR=PRC1,TYPE=OPERATOR,AUTH=PRIV,;
                                DEST=NMSC,ALOC=YES,CEDS=YES,OPDS=YES,;
                                PPID=OFF
*
```

```
*****************************************************************
*  NMS LINE CLASS, LINE, AND TERMINAL DEFINITION
*
UN96SH          LCLASS          LPH=U100L,SPEED=9600,OPTIONS=(DIR,SYHD)
*
LNMSP1          LINE            PRCSR=PRC1,CLASS=UN96SH,ADR=04
*
GNMSP1          GROUP           PRCSR=PRC1,LINE=LNMSP1
*
NMSPC1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'63'),;
                                DEST=NMSC,FMT(24,80),AUTH=PRIV,;
                                ALOC=IMMED
*
*****************************************************************
*    UNISCOPE TERMINAL (TERMA1) DEFINITION FOR HOST ACCESS
*      SECOND SCREEN OF NMS CONSOLE TERMINAL ($SON TO TERMA1)
*
*
TERMA1          TERM            GROUP=GNMSP1,TYPE=UTS20,ADR=(X'21',X'64')
*
*****************************************************************
** IEEE 802.3 ILM-40 CONFIGURATION
*
LANETHR         LCLASS          LPH=(ILML,'ENET$')
LNP11B          LINE            PRCSR=PRC1,CLASS=LANETHR,ADR=X'2B',;
                                STA=X'08000B0C362B'
*
*****************************************************************
** FDDI ILM-40 CONFIGURATION
*
LANFDDI         LCLASS          LPH=(ILML,'FDDI$')
LNP13B          LINE            PRCSR=PRC1,CLASS=LANFDDI,ADR=X'3B',;
                                STA=X'08000B0C363B'
*
*****************************************************************
** TCP-IP STACK CONFIGURATION:
*
** EU AND XEU DEFINITIONS FOR TCP-IP STACK:
** (CONFIGURES SERVER AND USER TELNET ON PRC1)
*
EUTCP           EU              PRCSR=PRC1,TYPE=TCPIP
USRTEL          XEU             DS=PRC1,DESTTSU='DTPX',DESTSSU='EUTCP'
*
** SUBNET AND IPADR DEFINITIONS FOR THE LAN ATTACHEMENTS:
*
SNENETP1        SUBNET          PRCSR=PRC1,TYPE=LAN,IPNETID=(192,9,10),;
                                LINE=LNP11B,IPROUTER=YES
SNFDDIP1        SUBNET          PRCSR=PRC1,TYPE=IPFDDI,IPNETID=(192,9,12),;
                                LINE=LNP13B,IPROUTER=YES
IPP1ENET        IPADR           PRCSR=PRC1,DCAEP=DCP1,;
                                IPADDR1=(192,9,10,30,LOCAL)
IPP1FDDI        IPADR           PRCSR=PRC1,DCAEP=DCP1,;
                                IPADDR1=(192,9,12,30,LOCAL)
*
*****************************************************************
* END OF EXAMPLE
*****************************************************************
```

Appendix B
TCP-IP Stack Configuration Concepts

This section provides a general description of the environment in which the TCP-IP Stack program product functions. It introduces you to the concepts involved in TCP/IP technology, and provides a brief history of the development of these protocols.

If you are responsible for configuring and installing TCP-IP Stack, but you are unfamiliar with TCP/IP technology, the Defense Data Network (DDN) and other environments in which the program product functions, this section can help you understand TCP/IP.

B.1. TCP/IP Development

TCP/IP is a communications architecture comprised of a set of layered communications protocols originally defined by the United States Department of Defense (DoD) Advanced Research Projects Agency (DARPA). The name TCP/IP represents two protocols that provide the primary communications services on which the architecture is based: transmission control protocol (TCP) and internet protocol (IP).

TCP/IP technology is based on communications research conducted on the Defense Advanced Research Projects Agency Network (ARPANET), which was initiated as a DoD experimental packet-switched network. DARPA research conducted in the 1970s led to a set of military standards for data communications that define the basic TCP/IP protocol suite. ARPANET began as a relatively simple packet-switched network that interconnected major computer science and defense-related research institutions in the United States. As ARPANET users adopted TCP/IP technology, the network gradually grew to include independent local area and wide area networks, allowing systems to communicate across network boundaries.

ARPANET is now part of a large system of interconnected networks, commonly called the Internet, which ties together the U.S. military production network (MILNET), major universities and research institutions, and a number of government agency and research laboratory networks. The Internet has grown to support communications among thousands of independent private and public data networks comprising tens of thousands of computer systems and terminals. This demonstrates the strength of TCP/IP technology, which became a mandated suite of protocols in 1983 for all interconnected networks and systems using ARPANET communications facilities.

Because TCP/IP is a well defined, flexible set of public-domain protocols, many vendors provide common TCP/IP protocol implementations for their equipment. TCP/IP communications technology has become popular in the private sector as a method of interconnecting systems from different vendors on local area networks (LANs) and wide area networks (WANs). Although it is not defined by international standards, TCP/IP has become a de facto standard for open systems communications in recent years.

B.1.1. The DoD Communications Model

The ARPANET network structure is based on a communications model developed by the DoD to define the basic requirements for data communications. The DoD model consists of three agents: processes, hosts, and networks. These components can be defined as follows:

- Processes

Processes are fundamental operations that ultimately communicate. End user applications are defined as processes in this model. A file transfer operation is one example.

- Hosts

Hosts are computer systems that execute processes. Hosts typically support multiple processes and allow them to operate concurrently.

- Networks

A network is a system of interconnected computers. Host computers are connected by networks. Data is exchanged between hosts through networks.

Figure B–1 illustrates the DoD conceptual model.

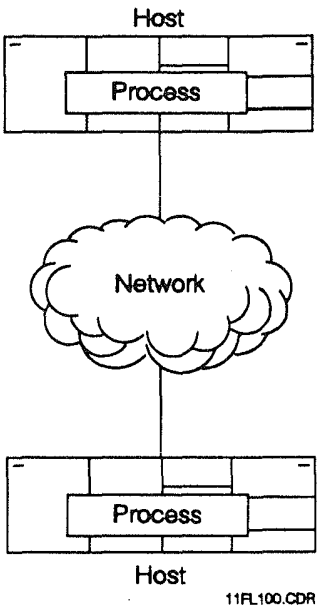


Figure B–1. DoD Communications Model

B.1.2. What is a Protocol?

A protocol is simply an agreement on how to communicate. Protocols define guidelines for interactions to help communicating parties understand one another.

In a layered communications architecture, a protocol is a set of rules by which data is exchanged between equivalent (peer) processing entities within the same layer. Each layer of the communications architecture can contain various processing entities that perform specific telecommunications functions. Protocols in a layered architecture support peer-to-peer data exchange between entities operating on different processing systems.

Protocols accomplish their tasks through two definitive methods:

- They establish two-way communications (called handshaking).
- They manage established communications paths by performing error handling and controlling session activity (traffic management).

B.1.3. What is an Internetwork?

LANs and WANs evolved out of a need for computer users to access resources beyond those available in a single computer system. Computer system users often need to exchange information and access computing resources over long distances, sometimes beyond the geographical boundaries of their own network.

The increasing need to share computing resources and data has demanded the merging of independent physical networks into sets of cooperative networks, called internetworks. The limitations of proprietary networks have caused users to seek connections to internetworks, and have imposed upon those networks standards for interoperability and data transmission reliability.

Interconnection of multiple networks, by means of a common set of communications protocols, enables any two stations on constituent networks to communicate. An internetwork (hereafter referred to as an internet), makes it possible for systems composed of otherwise incompatible hardware components to communicate despite their diverse technologies.

B.1.4. What is the Defense Data Network (DDN)?

The Defense Data Network (DDN) is a common name for the communications network and the supporting technology that interconnects all United States DoD data processing systems. The term DDN loosely refers to the DARPA Internet and TCP/IP technology, due to the defense-related development history of the Internet, and the fact that it links military and defense-related research facilities worldwide. DDN most accurately refers to MILNET and associated portions of the Internet used for communications among military installations.

MILNET was formed in 1984 when DARPA divided ARPANET into two separate packet-switched backbone networks using the same network technology. ARPANET remains primarily research-oriented, while MILNET was created to provide a reliable production network for military use. ARPANET and MILNET are connected at several points to allow communications among systems on both networks.

The Internet backbone service (both ARPANET and MILNET) is operated and maintained by the Defense Communications Agency (DCA) of the DoD. ARPANET and MILNET consist of numerous packet switched nodes (PSNs) at various geographic locations, interconnected by leased telecommunications lines, radio links, and satellite links. PSNs (originally called interface message processors, or IMPs) provide a standardized interface through which individual computer systems or networks can attach to the packet-switched backbone network. PSNs originally used a proprietary interface protocol, but DARPA eventually adopted a more universal interface based on CCITT X.25 protocols.

B.1.5. Why Implement TCP/IP?

TCP/IP protocols enable computer systems to communicate across interconnected subnetworks and transmit data across mixed communications media having different performance characteristics. TCP/IP supports interconnection of WANs and LANs, and provides data routing across dissimilar subnetwork types.

Because it uses standardized communications services and addressing schemes, TCP/IP has been used in numerous network environments that require communications among systems supplied by multiple vendors. The TCP/IP protocol structure makes differences in system hardware and network transmission media transparent to the user. It allows systems and networks to be interconnected to create an open systems communications environment. Ideally, users on any standard TCP/IP network can communicate with users on any other TCP/IP network.

B.1.6. TCP/IP Communications Architecture

To reduce the complexity of communications protocols, most networks use an architecture that divides network functions into logical layers. Each logical layer is limited to performing a specific type of communications service. Each layer provides specific services to the adjacent upper layer, and uses services provided by the adjacent lower layer.

When a system transmits data, processing entities at each layer encode the data and add information about the processing they perform before passing the data to the next layer. The layer-specific processing information is interpreted only by equivalent (peer) entities on the receiving system. Logically, entities at each layer communicate only with peer entities in the same architectural layer on other systems. Peer processing entities use specific protocols to exchange data. This concept is called peer-to-peer communications, and is fundamental to open systems communications architectures.

TCP/IP protocol structure is based on a layered architectural model that evolved from ARPANET experimentation. The main differences between the ARPANET TCP/IP model and the Open Systems Interconnection (OSI) model are in the location of reliability control (error detection/recovery) within the protocol hierarchy, and the application system structure defined by each model. The primary goal of the TCP/IP architecture is to interconnect multiple independent networks and support host-to-host communications across the resulting wide area network. The TCP and IP protocols together provide a common data transport service for all hosts in the network. Figure B–2 shows how data is exchanged between peer entities as it crosses a typical network between two hosts.

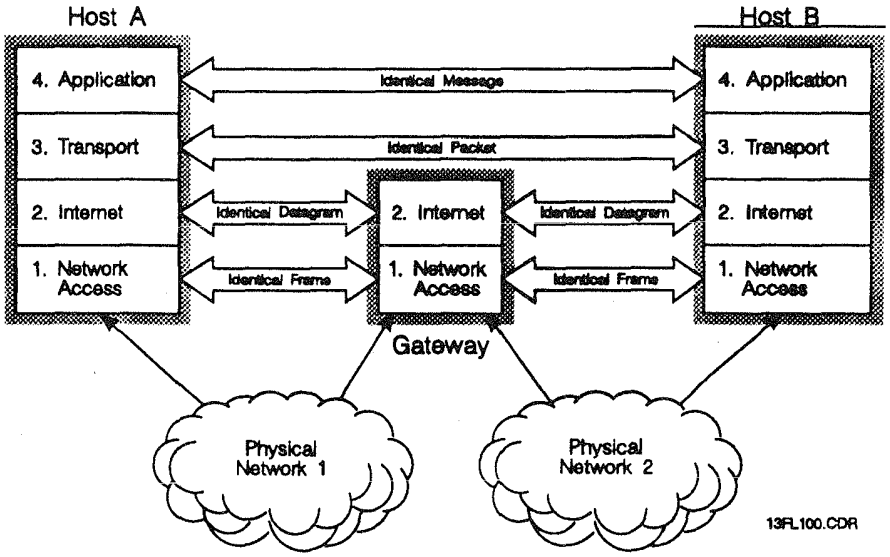


Figure B–2. Peer-to-Peer Communications in a TCP/IP Network

In the TCP/IP architecture, the process/application layer encompasses all application system services performed on hosts. It is not subdivided into specific functional layers as in the OSI model, and it does not include a defined set of communications services common to all application processes. Application services used in TCP/IP network environments are defined by functional and implementation standards that apply only to each specific service. For example, some application services adopted by the DoD are defined by MIL-STD specifications to ensure interoperability within the Internet.

The DCA has formally adopted some lower layer protocols that are common to both OSI and TCP/IP networks for use in the Internet. The Internet is still considered an experimental network, and it is expected to gradually adopt more internationally standardized protocols as they evolve. Consequently, current protocol implementation in TCP/IP networks conforms more closely to the OSI model than to the original ARPANET model. However, the original four-layer ARPANET model is still a valid conceptual description of TCP/IP architecture.

B.1.7. Internet Protocol Development

Protocol development for the Internet environment is an ongoing process. Because the Internet user community is so diverse, many services and protocols have been developed independently of the DoD to meet particular processing and communications needs. Any Internet user may develop a proposal for new services, protocols, or changes to existing services or protocols. Such a proposal is presented to the Internet user community as a Request for Comment (RFC), which is a formal document soliciting commentary and debate from the Internet community.

Software offered to the Internet community for general use is considered to be in the public domain. Unless formally adopted as a required Internet protocol, the implementation of any particular service is site-dependent. A list of required protocols for the Internet environment is issued periodically as an RFC.

A group of Internet users, the Internet Engineering Task Force (IETF), acts independently to examine available services, protocols, and implementation methods. The IETF meets periodically to make short-term implementation recommendations to the Internet community in support of system interoperability within Internet. The IETF is a task force of the Internet Activities Board (IAB), an independent group of researchers that monitors and directs much of the research, development, and problem-solving activities within the Internet community. The IAB exchanges information with DARPA and other organizations that support communications research in the Internet.

B.2. TCP/IP Functional Overview

The protocols comprising TCP/IP can be organized into four logical layers, as shown in Figure B–2. These layers rely upon a fifth layer that represents various physical subnetwork types that support TCP/IP communications. The following subsections describe the communications protocols in each layer of the TCP/IP architecture.

Table B–1. TCP/IP Logical Layers and Standard Protocols

Layer	Protocol
Process/application layer	File transfer protocol (FTP) Simple mail transfer protocol (SMTP) Network virtual terminal protocol (TELNET)
Transport (host-to-host) layer	Transmission control protocol (TCP) User datagram protocol (UDP)
Internet layer	Internet protocol (IP) Internet control message protocol (ICMP) Address resolution protocol (ARP)
Network access layer	CCITT X.25 (for packet-switched PDNs) FIPS 100 (DDN X.25, for Internet access) 802.2/802.3 (for LANs)

B.2.1. Process/Application Layer

The process/application layer contains end user application processes that operate on host systems. This layer interacts with the protocols at the transport layer to send or receive data. Additionally, this layer determines the form its data takes - a sequence of messages, or a stream of bytes. The process/application layer contains protocols for resource sharing (computer-to-computer) and remote access (terminal-to-computer).

B.2.1.1. Application Services Available through the Internet

All hosts in the Internet are required to implement a certain minimum set of services (protocols) to support interoperability. These services include:

- Electronic mail (simple mail transfer protocol, SMTP)
- File transfer (file transfer protocol, FTP)
- Remote login
- TELNET Network Virtual Terminal (NVT) protocol)

Electronic Mail

This application enables users to electronically create, send, and receive correspondence in the form of short text files. Functions are invoked interactively from the user's terminal. The main protocol used in this application is the simple mail transfer protocol (SMTP).

File Transfer

This application enables users to send or receive large, more complex data files. The main protocol used in this application is the file transfer protocol (FTP).

There are three basic purposes for file transfer:

- Store a file for subsequent retrieval
- Print a file (most often, on the local printer)
- Run a file as a program, or process it as data

Remote Login

This application enables users of a given system to connect (login) to remote systems and establish interactive sessions.

The main protocol used in this application is the network virtual terminal (NVT) protocol known as TELNET. The TELNET protocol specifies how terminals communicate with applications running on host computers, and how two terminals communicate with one another. TELNET was developed in the early 1970s, when most terminals were unintelligent devices. It was originally designed for scroll mode, rather than form mode terminals.

TELNET

The main function of TELNET is to establish and manage two simplex data streams, one in each direction. Because the protocol does not require identical data structures at each end of the network, the terminal process converts the stream of 8-bit bytes transmitted over the lines (as keys are hit) into a format that can be recognized by the network. TELNET then adapts the transmitted character code to

the network standard. TELNET thus provides a fairly general bidirectional, 8-bit byte communications facility.

TELNET enables terminals to access applications running on hosts connected to the DDN and other networks that use TCP/IP protocols. It supports a standard method of interfacing terminal devices with terminal-oriented processes.

As a service recipient, TELNET accommodates connection establishment and negotiates service options. It changes defaults and sets parameters on the virtual terminal with a handful of commands that may be initiated at any time by either side of the network. As a service provider, TELNET optimizes the function of the host interface by negotiating the most effective operating characteristics.

TELNET comprises several commands that are entered from the terminal. Through these commands, TELNET can initiate a TELNET connection, negotiate an option, send a specific command to the remote TELNET, or terminate the TELNET connection.

B.2.2. Transport (Host-to-Host) Layer

The transport layer is often called the host-to-host layer. It controls the quality of data transmission, ensuring that a reliable path exists at all times between end nodes. It is the highest layer responsible for the data integrity of end-to-end logical connections.

The transport layer ensures that an optimal quality of service is provided from the start of a session to its completion. It controls a session by passing along information about the quality of the communications. It uses various mechanisms, unrelated to the supported application processes, to ensure that data is exchanged reliably.

The most significant function of this layer is to provide successful end-to-end communications, or communications from one application system to another.

This layer organizes streams of data into segments, and passes the data and a destination address to the next layer for transmission. This layer also provides an interface between the upper layer protocols (ULPs) that provide application functions and the lower layer protocols that provide communications-oriented functions.

When multiple applications from one system are accessing the network, the transport layer provides a code that identifies the application program that sent the information, as well as a code that identifies the recipient application.

B.2.2.1. Transmission Control Protocol (TCP)

The transmission control protocol element provides reliable communication between pairs of processes (applications) on logically distinct hosts, across networks and sets of interconnected networks. TCP serves as the basis for the DoD concept of interprocess communications among systems.

TCP appears in the DoD protocol hierarchy at the transport layer. It provides reliable connection-oriented, ordered, two-way simultaneous, full-duplex data transfer, with capabilities for flow control. TCP, therefore, compensates for environments in which loss, damage, duplicated or out-of-sequence data, and network congestion might otherwise occur. Consequently, it is well-suited to support military, governmental, and commercial applications.

TCP provides its services on top of the internet and network access layers, which might be prone to data errors. TCP requires addressing and control information to be initialized and maintained during data transfer. It uses the following mechanisms to render data exchange more reliable:

- Error detection
- Sequencing
- Positive acknowledgment with retransmission (PAR)

TCP uses the PAR mechanism to recover from the loss of a segment of data by lower layers. In operation, the sending TCP resends a segment at timed intervals until a positive acknowledgment is returned.

- Flow control

The TCP flow control mechanism enables a receiving TCP to govern the amount of data dispatched by a sending TCP. The mechanism is based on a window that defines a contiguous interval of acceptable sequence-numbered data. As data is accepted, TCP moves the window upward in the sequence. This window is carried in every segment, enabling peer TCPs to maintain current window information.

- Multiplexing

Multiplexing allows multiple upper layer protocols (ULPs) within a single host, and multiple processes in a ULP, to use TCP simultaneously. This mechanism associates identifiers, called ports, with ULP processes accessing TCP services. Each ULP connection is uniquely identified with a socket. A socket is the concatenation of a port identifier and an internet address. Each connection is uniquely named with a socket pair. This naming scheme enables a single ULP to support connections to multiple remote ULPs. ULPs that provide popular resources are assigned permanent ports, which are referred to as well-known ports.

A message is always sent over a connection from one socket to another socket. TCP uses the pair of sockets that form a connection to differentiate between multiple users.

B.2.2.2. User Datagram Protocol (UDP)

UDP provides a connectionless datagram transport service to upper layer protocols in the process/application layer. It is less commonly used than TCP because it requires the upper layer protocols to provide any necessary error recovery and end-to-end reliability control functions. In general, this requires a user-written

application, because reliability control is not built into most process/application layer protocols common to the Internet environment. UDP has the following characteristics:

- Provides a data checksum service for data integrity error detection
- Follows the same port addressing and multiplexing rules as TCP
- Does not guarantee delivery of datagrams
- Requires application ULPs to provide end-to-end reliability control

B.2.3. Internet Layer

The internet layer provides a routing function across multiple networks. It relays data from one network to another, from the source host to the destination host.

B.2.3.1. Internet Protocol (IP)

IP is responsible for routing data from source to destination across the network. It accepts data packets (segments) from the transport layer and passes the data to the network access layer in the form of IP datagrams. Each IP datagram can be routed across the network according to the IP addressing information it contains. An IP (internet) address is a unique identifier that indicates the destination device to which the datagram should be sent.

The internet protocol is implemented within gateways as well as hosts. Gateways are processors that connect two subnetworks. IP can determine if a datagram can be delivered to its destination directly (on the same subnetwork), or if it should travel through a gateway (to a different subnetwork). It uses a routing algorithm to make this determination. IP can also check the validity of incoming datagrams, and determine if the incoming datagram should be processed locally or transported to another location. IP determines the appropriate route to the destination and passes the datagram to the corresponding subnetwork interface for transmission.

Another major function of the internet layer is datagram fragmentation and reassembly. Datagrams are broken into smaller pieces before they are transmitted across the network. IP software at the receiving end of the network then reassembles the fragments into full datagrams. Fragmentation is often necessary when a network cannot process the datagrams at their initial size. IP provides its services through four means of control: type of service, time to live, header checksum, and options, as described in the following paragraphs. These services are configured through parameters associated with the configuration statements described in Section 3.

Type of Service

Type of service controls the quality of service provided by the networks in the internet. It is a set of parameters used to select network transmission characteristics.

Time to Live

Time to live controls the lifetime of a datagram in the internet. It is set by the sender of a datagram and is the maximum amount of time the datagram is allowed to exist in the network.

Header Checksum

Header checksum ensures the reliability of data in the IP header.

Options

Options control functions not commonly used in most communications activities. There are six options:

Security	This option enables hosts to send security, compartmentation, handling restrictions, and transmission control code parameters.
Loose source, record route (LSRR)	This option enables the source of an internet datagram to supply routing information used by gateways to forward the datagram to its destination, and to record the route taken. This option is termed loose source route because the gateway or host IP can use any route of any number of other intermediate gateways to reach the next address in the source route.
Strict source, record route (SSRR)	This option enables the source of an internet datagram to supply routing information used by gateways to forward the datagram to its destination, and to record the route taken. This option is termed strict source route because the gateway or host IP must send the datagram directly to the next address in the source route through the next directly connected network. The datagram reaches the next gateway or host specified in the route.
Record route	This option enables the route of an internet datagram to be recorded.
Stream identifier	This option enables a 16-bit stream identifier to be carried through networks that do not support the stream format.
Internet timestamp	This option enables the timestamp alone, the timestamp and the internet address together, or the timestamp for a specific internet address to be entered in the IP header.

IP Output and Input Requests

When the IP receives an output message from an upper layer protocol (ULP), it builds an IP header from the information provided at the time the request was made. It then determines the local network address for the message.

If a datagram is too large for the network to process, the message is fragmented to satisfy the network interface. Finally, the datagram is passed to the network interface module.

When the IP receives a fragmented datagram from the network, the fragments are collected until a complete datagram is received. When a complete datagram is received, the IP header is verified and the datagram (with any additional information required by the ULP) is passed to the ULP.

B.2.3.2. Internet Control Message Protocol (ICMP)

ICMP is a sublayer protocol within the internet layer that enables error messages to be transported to upper layer protocols on behalf of lower layer protocols. Because internet layer and network access layer protocols have no reliability control functions, they must notify upper layer protocols of any transmission or data integrity errors.

ICMP provides a set of standard messages in a common format that can be interpreted by upper layer protocols. There are two types of ICMP messages: error messages and query messages. ICMP is a required protocol in the Internet environment, and it is actually integrated as part of IP. ICMP messages are transmitted as IP datagrams, using IP as the routing protocol. IP does not interpret ICMP messages.

B.2.3.3. Address Resolution Protocol (ARP)

In local area networks, a sublayer protocol called address resolution protocol (ARP) works in combination with IP to route data to LAN-connected hosts. ARP was developed specifically to map internet addresses used by IP to physical addresses used by the LAN subnetwork protocol. It is commonly implemented for LANs that are connected to the Internet through a gateway, and for stand-alone LANs that use TCP/IP protocols.

For example, an application on a LAN-connected host relies on IP to route data to another host connected to the same LAN. All hosts on the LAN have both an internet address and a physical (station) address. IP on the sending host can identify the destination host only by the destination internet address that the application provides. The LAN subnetwork protocol (MAC on an LAN) cannot recognize internet addresses. ARP can equate the internet address with the LAN station address of each host on that LAN to provide IP routing between those hosts.

IP on the sending host sends an ARP request over the LAN. ARP requests are broadcast at the physical layer to all hosts on the LAN. The ARP request header contains the station address and internet address of the sending host, and the internet address of the destination host. All hosts receive the ARP request, but only the host corresponding to the destination internet address will respond to the sender. ARP on the destination host returns the address mapping information, and both the sending and receiving hosts retain each other's addressing information for future use.

B.2.4. Network Access Layer

The services of the network access layer provide appropriate interfaces to the underlying physical transmission media that make up the network. This layer manages the exchange of data between a host computer and the network to which it is attached. It accepts internet protocol datagrams and transmits them over the physical subnetwork in the appropriate format. Network access services allow internet protocol to route data across multiple interconnected subnetworks, regardless of the subnetwork type.

At this layer, the Internet relies on national and international standards such as CCITT X.25 for WANs, and 802.3 (ISO 8802/3) for LANs.

B.2.4.1. WANs: CCITT Recommendation X.25

CCITT Recommendation X.25 was chosen as the standard set of network access protocols for use within the DDN. For DDN implementations, X.25 protocols must conform to the requirements defined in FIPS 100. X.25 protocols were originally developed by CCITT for packet-switched public data networks, and address the requirements for DTE-to-DCE interface protocols in that type of network. X.25 protocols are directly applicable to the Internet physical network structure, providing a standardized interface to the PSNs that make up the network backbone.

X.25 comprises three protocol layers. For DDN implementations, the X.25 packet level protocol (PLP) operates as a lower internet layer subprotocol with IP operating above it as the routing protocol. The X.25 lower layer protocols (X.25 LAPB and X.21 bis) together fulfill the role of TCP/IP network access layer services.

B.2.4.2. LANs: Logical Link Control (LLC) Protocol

LLC is an upper data link layer subprotocol developed by IEEE for use with various local area network types. LLC is described by the 802.2 standard, and was adopted by ISO as ISO 8802/2. For TCP/IP communications, LLC provides a standardized link layer interface to different types of LAN subnetworks described by associated IEEE standards (such as 802.3).

B.2.4.3. LANs: Media Access Control (MAC) Protocol

MAC is a lower data link sublayer used in conjunction with LLC. The 802.3 standard describes the MAC subprotocol for carrier sense multiple access with collision detection (CSMA/CD) LANs. CSMA/CD LANs are high-speed multipoint baseband networks. The associated subnetwork type is often called an 802.3 LAN.

ISO describes this protocol in the ISO 8802/3 standard. Subprotocols for other LAN subnetwork types are described in the IEEE 802 and ISO 8802 series of standards.

The 802.3 LAN type is closely associated with TCP/IP protocols. 802.3 uses the same transmission medium as Ethernet LANs, and accommodates Ethernet version 2 frame structures. 802.3 or Ethernet LAN subnetworks are most commonly implemented to interconnect UNIX host systems using TCP/IP communications protocols.

B.3. Summary

Figure B–3 shows how TCP/IP protocols implemented by Unisys software fit into the logical layers of the ARPANET architectural model. TCP-IP Stack implements TCP, UDP, IP, ICMP, and TELNET, and provides access to host-based Internet application services.

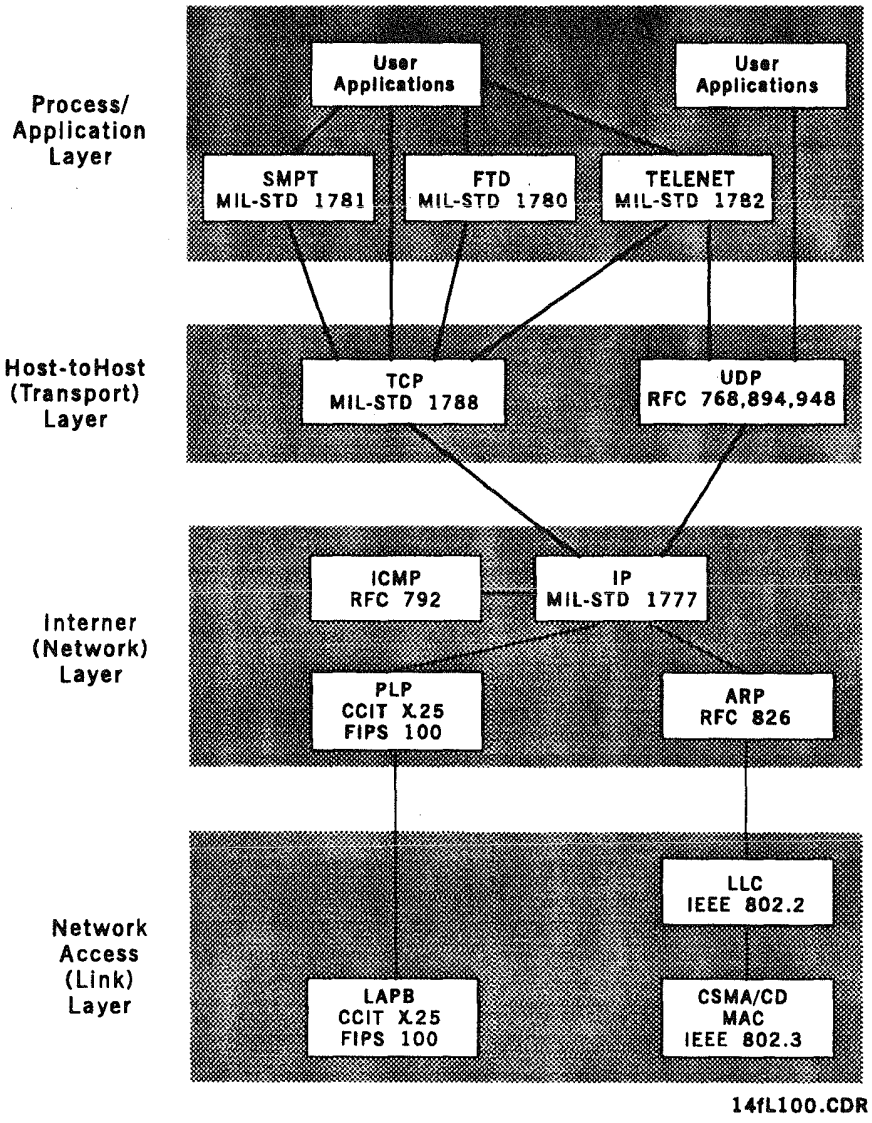


Figure B–3. Relationship of TCP/IP Protocols to Architectural Layers

B.4. TCP-IP Stack Addressing Concepts

TCP-IP Stack allows you to use standard addressing mechanisms that support data routing in TCP/IP networks.

B.4.1. IP Broadcast Address

IP includes a broadcasting mechanism by which a message can be transmitted to all destinations in a given network. The IP entity on all nodes can receive a datagram directed to a standard broadcast address.

The Internet standard IP broadcast address was originally specified to be all zeros. However, a broadcast address of all zeros caused unexpected problems with some early TCP/IP implementations on the Internet. The standard IP broadcast address was eventually changed to all ones: 255.255.255.255, also written as IP[-1,-1IP]. Not all implementors adapted to the new scheme, and to this day there are important implementations running on the Internet using a broadcast address of IP[0,0IP]. TCP-IP Stack recognizes both broadcast address types in incoming IP datagrams, and can transmit the standard (all ones) address.

Only the 802.2 and 802.3 LAN subnetwork types support broadcasting at this time. An IP broadcast message is transmitted to all stations on a LAN by means of the LAN broadcast address. A broadcast on the LAN always uses the same destination station address (X'FFFFFFFFFFFF'). Some LAN mechanisms (such as ARP) use broadcast addressing that involves only the link layer protocols and does not require IP broadcasting.

B.4.2. Subnet Address Masks

You can configure TCP-IP Stack to support subnet routing in local networks. Subnet routing allows an IP gateway to route datagrams to a local network (group of subnetwork attachments) which is identified to the rest of the TCP/IP network by a single network number. Because the range of unique network numbers is finite for each address class (A, B, or C), subnet routing was developed to increase the number of individual subnetworks that can be reached through IP routing.

A local network that uses subnet routing consists of two or more individual subnetworks attached to the gateway node. To other hosts in the greater TCP/IP network, each destination in the local network appears to have the same network number and a unique host number. However, the local subnetworks have locally-assigned network numbers that are known only to the gateway node. The host number portion of the IP address is used to define these locally-assigned network numbers. The local gateway node differentiates destinations in the local network by means of a subnet mask, which permits a nonstandard interpretation of part of the host number.

A subnet mask is a locally configured number, conforming to the IP address format, that TCP-IP Stack uses to interpret IP addresses in incoming datagrams. The software performs a logical AND on the subnet mask address and the incoming IP address to derive the locally-assigned network numbers of attached subnetworks. For TCP-IP Stack subnet routing configurations, you can use one of two methods to specify a subnet mask:

- TCP-IP Stack can automatically calculate an appropriate subnet mask for class A and B addresses.
- You can explicitly configure a subnet mask for class C addresses.

B.4.3. DDN Address Mapping Algorithm

The DDN supports a number of unique facilities and features that are not ordinarily available on generic X.25 networks. Among these is an algorithm that automatically maps IP addresses to DTE addresses. This makes it unnecessary to configure paired DTE addresses for every remote TCP-IP host one is trying to reach.

The algorithm is standardized and described in *Defense Data Network X.25 Host Interface Specifications*, published by the Defense Communications Agency.

The algorithm designates the four bytes of the IP address as `n.h.z.i` for a Class A network ($1 \leq n \leq 126$), and as n1.n2.h.i for a Class B network ($128 \leq n1 \leq 191$).

The terms n, n1 and n2 designate, in decimal notation, the parts of any valid Class A or B network number. The term h is also known as the host number or the trunk number, while the term i is the "IMP" or "PSN" number. DDN subscribers may be told that their link is number 16 (=h) on PSN (or node) 45 (=i).

The DTE addresses that are calculated from the IP address take the form

OOOO F ddddd OO SS

where F is a flag value that is zero (0) when the h component of the IP address is less than 64 ($h < 64$). Values of h greater than or equal to 64 ($h \geq 64$) are known as *logical* host numbers, and the flag digit F is set to one (1) for them. ddddd takes one of two formats: When h is less than 64 ($h < 64$), these five digits are designated iiihh, where iii is the i component of the IP address, in right-justified decimal notation, filled with zeros (0) to complete all three digits. hh is the component of the IP address, also in right-justified decimal notation, filled with zeros (0) to complete both digit positions.

When h is greater than or equal to 64 ($h \geq 64$), ddddd is the result of a calculation, expressed in right-justified decimal notation, and zero-filled to complete all five digits. The calculation follows the formula

$$r=(h*256)+i$$

where r is the result and h and i are the respective portions of the IP address.

B.5. TCP-IP Stack Nonstandard TCP Port Numbers

TCP port numbers are used to address certain applications, like Telnet. TCP-IP Stack supports a number of nonstandard port numbers. The following is an example of these numbers:

Port Number	Description
63	Reserved for BNA
97	DCA RTC trunks
98	DCA DNS trunks via IP (not DCP-DCP only)
264	DCA termination system
265	DCA DNS trunks via TCP (especially TS-1 and ES trunk)

B.6. Boundary Nodes between Subnetted Networks

Subnetted networks use part of the host address portion to divide a network into several smaller pieces (subnetworks). RIP will communicate subnetwork routing information among nodes connected to the subnetworks of the same network. But nodes on the boundary between two networks will not propagate subnetwwork routing information. Instead, they are known strictly as the router to the whole network, to nodes that are not paart of the subnetted network.

B.7. Host Name Tables

A host name table is a database containing IP addresses of specific destination hosts and corresponding names assigned to those hosts. Some hosts may have names, assigned by the network administration authority, that are unique throughout a given TCP/IP network.

TCP-IP Stack maintains a local host name table, allowing it to associate a name (such as HOST1) with a unique IP address. The host name is used when making TELNET connections, specifying the name on the CONNECT command instead of specifying the IP address.

B.8. TCP-IP Stack Routing Concepts

TCP-IP Stack allows you to configure a number of standard routing mechanisms used in TCP/IP networks.

B.8.1. IP Routing

The internet protocol (IP) provides a connectionless datagram routing service for TCP/IP networks. IP routing moves datagrams from a source system to a specific destination system across a network of interconnected subnetworks. IP routing nodes (often called IP gateways IP) provide logical connections between individual subnetworks, creating an overall path that datagrams can travel to reach a remote destination.

In TCP/IP networks, an IP router physically attaches to two subnetworks that are logically part of the same network, carrying data from one subnetwork to the next when appropriate. A router maintains IP addressing information about individual hosts and other routing nodes reachable through the subnetworks to which it is attached. Routers can exchange this information with other routers by means of various routing exchange protocols. To carry out these exchanges, TCP-IP Stack implements routing information protocol (RIP), which is commonly used on routers that interconnect 802.3 (Ethernet) LAN subnetworks.

In TCP/IP networks, a router can use only internet protocol to provide network layer routing services. A true IP router does not convert from one routing protocol to another as it routes datagrams between subnetworks. Both attached subnetworks must use IP as the routing protocol. With TCP-IP Stack, you can configure the following IP routing capabilities on a DCP:

- IP routing
- Subnet routing (subnetting)
- Configured IP routes
- Routing information protocol (RIP)
- Autonomous system numbers
- Host names

B.8.2. Default IP Gateways

It is rarely practical to configure all gateways TCP-IP Stack must know about. Occasionally, TCP-IP Stack may need to send datagrams to a destination network for which it has no routing information. You can configure another specific gateway in the network to which TCP-IP Stack can send datagrams that it otherwise cannot route. Such a 'default gateway' maintains routing information for a much larger portion of the network, enabling it to reroute datagrams by notifying the source node of an alternative route.

When it receives such a misdirected datagram, the default gateway generates a control message called a 'redirect', which is sent to the originator of the misdirected datagram (in this case, TCP-IP Stack). The redirect informs the originator of the gateway address to which it should direct datagrams to reach a particular network. You can configure a number of default gateways to which TCP-IP Stack directs datagrams when it knows of no route to the destination.

B.8.3. Routing Information Protocol (RIP)

TCP-IP Stack supports an internal gateway protocol called routing information protocol (RIP). This protocol was originally implemented in Berkeley UNIX systems, but has since become a widely-used de facto standard, described in RFC 1058.

B.8.3.1. Advantages of RIP

- Gateways need to maintain tables of routing information in order to move datagrams across multiple subnetworks to their destination. This routing information may be configured statically, or it may be acquired dynamically through RIP.
- Internal gateway protocols allow gateways internal to an autonomous system to share routing information. When the network topology changes, RIP automatically distributes the new routing information.

B.8.3.2. Disadvantages of RIP

- RIP was designed to operate specifically on LANs. It has few good safeguards against problems related to transmission delays, so it will not work well on slower networks, such as X.25 networks.
- RIP is also optimized to operate by broadcasting its routing information, so a different mechanism must be used to propagate routing information among gateways in nonbroadcasting networks. TCP-IP Stack implements a mechanism called dynamic neighbor discovery to support RIP operation over Telcon DNS networks.

- Finally, RIP is limited to networks having a cross section of 15 hops or less. A hop represents the movement of a datagram across a subnetwork to the next gateway in the route. Any network more than fifteen hops removed from a gateway is considered unreachable.

TCP-IP Stack lets you activate RIP on any directly connected network that uses IP or Telcon DNS routing protocols. If there are no other RIP gateways in a network, activating RIP introduces additional overhead without adding any benefits. Use RIP only in networks containing other RIP gateways. RIP and Subnetworks explain about boundary gateways.

B.8.4. Dynamic Neighbor Discovery

RIP was developed primarily for connectionless subnetworks such as LANs using IEEE or Ethernet link layer protocols. Connectionless LANs support broadcast addressing, which allows a message to be sent to all attached stations simultaneously. On connectionless LANs, mechanisms such as RIP and ARP can interview all connected hosts at once, eliciting a reply as to the whereabouts of a host owning a specific IP address.

Most networks based on connection-oriented link protocols do not support broadcasting, and therefore have no built-in support for routing exchange protocols. If you wish to include a network of this type in an internet, you generally cannot use RIP to propagate routing information throughout that network. TCP-IP Stack allows you to configure a Telcon DNS network as part of an internet, and make use of DNS routing capabilities to support TCP/IP communications. TCP-IP Stack implements a technique called dynamic neighbor discovery, which enables routing information protocol specifically for Telcon DNS networks.

Dynamic neighbor discovery maps IP addresses to DNS addresses for all TCP-IP Stack nodes in a DNS network. This method allows all TCP-IP Stack nodes in the DNS network to discover all other TCP-IP Stack nodes and exchange RIP routing information. You can identify such nodes as RIP neighbor gateways in your TCP-IP Stack configuration. You need to configure only a small subset of the TCP-IP Stack nodes as RIP neighbors.

B.8.5. Autonomous Systems

An autonomous system is a network that operates under a single administrative authority that has control over addressing and routing schemes used within the network. Gateways within an autonomous system can freely exchange addressing and routing information without affecting other interconnected networks. Within an internet, the network administration uniquely identifies each autonomous system by assigning a single autonomous system number to all of its constituent subnetworks.

Boundary nodes at the edges of the autonomous system may be attached to numerous other subnetworks belonging to other autonomous systems, each identified by a unique autonomous system number. The autonomous system number prevents internal gateway protocols from passing routing information from one autonomous system to another. Among other things, this helps to keep routing tables down to a manageable size.

You can assign autonomous system numbers in your TCP-IP Stack configuration to support operation of RIP over networks that use IP or Telcon DNS routing protocols. Do not use the autonomous system number to split a subnetworked network into multiple autonomous systems.

B.8.6. Network Bridging

A network bridge node connects two networks that use dissimilar communications protocols, allowing data to be passed from one network to a destination on the other network. TCP-IP Stack allows you to configure a DCP to function as a network bridge node between a TCP/IP network and a Telcon network.

A TCP-IP Stack bridge node lets you use DCA communications protocols across a TCP/IP network to interconnect parts of a Telcon network. It also lets you connect a Telcon network to a TCP/IP network, providing access to hosts running TCP/IP applications in the Telcon network. You can configure the following connections:

- DCP-to-DCP trunks across a TCP/IP network, supporting DCA communications
- DCP-to-DCATS links across a TCP/IP network, supporting DCA communications
- DCP attachment to a TCP/IP network, supporting hosts in a Telcon network

Bibliography

Douglas E. Comer and David L. Stevens, *Internetworking with TCP/IP*, Prentice Hall, New Jersey. 1991.

DCP Series SNA/net Configuration Guide (7831 5629). Unisys Corporation.

DCP Series TCP-IP Stack Programming Reference Manual (7831 5561). Unisys Corporation.

DCP Series TCP-IP Stack TELNET User Guide (7831 5553). Unisys Corporation.

DCP Series Local Area Network (LAN) Platform Configuration and Operations Guide (7831 5512). Unisys Corporation.

DCP Series Telcon Configuration Reference Manual (7831 5686). Unisys Corporation.

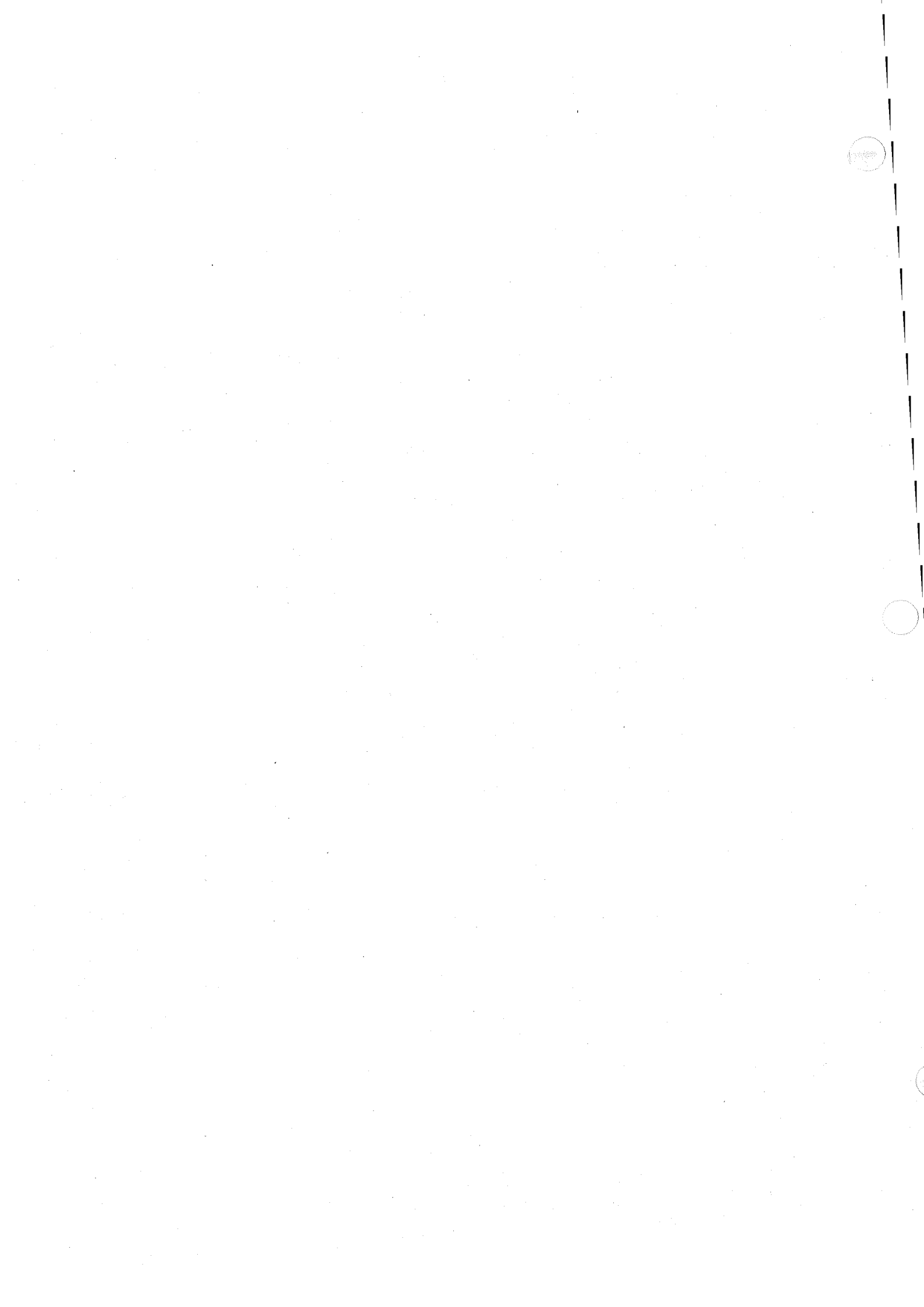
DCP Series Telcon Configuration Guide (7831 5678). Unisys Corporation.

DCP Series Telcon Installation Guide (7831 5645). Unisys Corporation.

DCP Series Telcon Message Manual (7436 0728). Unisys Corporation.

DCP Series Telcon Software Operations Guide (7831 5785). Unisys Corporation.

DCP Series X.25 Packet Switched Communications Software (PSCS) Configuration and Operations Guide (7831 5470). Unisys Corporation.



Glossary

A

absolute element
A OS 1100 element containing a complete program in a form suitable for execution by the Exec. Such elements normally occur as output from a collection of relocatable elements.

access rights
A feature of Telcon that enables you to limit terminal access to certain applications. For example, you can limit access to Network Management Services (NMS) to those terminals configured as NMS consoles.

ACK
See acknowledgment.

acknowledgment (ACK)
A system response, expressed as ACK, that indicates to the sender of a message that the message was received. Compare with negative acknowledgment.

address
An identifying number for a location in computer memory. Also, a unique number that identifies a particular network or communications software entity.

address resolution protocol (ARP)
A local area network (LAN) sublayer protocol that works in combination with the internet protocol (IP) in a TCP/IP environment to route data to LAN hosts. ARP maps internet addresses used by IP to physical addresses used by the LAN subnetwork protocol.

American National Standards Institute (ANSI)
The principal, centralized resource for information about voluntary standards developed in the United States.

ANSI
See American National Standards Institute.

application
A computer program that performs a task for the user, such as payroll processing or general ledger entries.

application environment
An environment that consists of one or more transport service users (TSUs) that interface with the termination system. TSUs provide the control and application addressing structure for one or more end users.

application interface bank (AIB)
An alternate file common bank (AFCB) that enables Hot-Standby software components to register with the redundant host, to send heartbeats to the monitor run, and to deregister with the resilient system. The AIB and the monitor run form the Automatic Recovery of Components (ARC).

application layer
See layer 7.

application management services (AMS)
The group of network control service functions associated with a particular applications environment. These functions are accessed, principally by CSUs and Network Management Services (NMS), to establish session paths between ports. Logically, each AMS function group is treated as a CSU with attached AMS end users that provide services to host applications for managing the activation, monitoring, control, and deactivation of the various application processes.

application mode
A keypad mode that assigns application control functions to numeric keypad keys.

architectural table
One of a group of internal tables used to implement DCA. area. A subdivision of a network that includes one or more regions or nodes. The regions within an area may be in different nodes. They may also overlap. The area subdivision is controlled by the area level of NMS authority.

ARP
See address resolution protocol.

ARPANET
Abbreviation for Advanced Research Projects Agency Network, part of the U.S. Defense Data Network. ARPANET requires the TCP/IP set of protocols. (This term is not spelled out in Unisys documentation.)

ASCII
Acronym for American Standard Code for Information Interchange (pronounced ASKEY), a 7-bit character code that defines 128 standard alphanumeric characters. ASCII is an industry standard that defines the codes for a character set to be used for information interchange between equipment of various manufacturers. It is the standard for digital communications over telephone lines. (This term is not spelled out in Unisys documentation.)

asynchronous transmission
A transmission in which the time interval between transmitted characters may be unequal. Transmission is controlled by sending start-stop bits.

attachment
An instance of an active association of a user program with the local area network (LAN) line module.

attribute
(1) A property or characteristic of entities or relationships with two parts, a type and a value. (2) A specific unit of information collected for an entity type (for example, a storage area attribute). (3) A security characteristic of a subject or object. When compared between subjects and objects, attributes determine which objects the subjects are authorized to access, and what type of access can be used. Security attributes include both mandatory and discretionary controls: project ID, account number, clearance level and clearance level range, security record owner, trusted privilege set, access list, and read-only or write-only.

autorecovery
A process initiated by the CMS 1100 contingency code that automatically reinitializes the CMS 1100 program following an abort. Autorecovery can be enabled or disabled by configuration.

B

backup host
A host that is not actively connected to a shared application group until a system failure occurs on the production host. When this happens, the Hot-Standby software installed on the backup host attaches the backup host to the application group and also recovers the failing host. The backup host takes over for the failed host as soon as the application group is successfully attached. The backup host can process non-shared batch and demand programs while the production host of a Hot-Standby system is running.

boot
To load the microcode and operating system into the random access memory of a computer.

bridge
The hardware or software used to connect two networks. See also transport bridge.

bus
(1) A single connective link between multiple devices. (2) A finite group of conductive paths that connect devices in parallel, so that all paths are shared by all devices. For example, plug-in cards in a PC connect to a bus.

C

call request
A control packet used to establish a network connection across a packet-switched PDN. The call request packet contains addressing information and optional special network facility requests such as requests for reverse charging, nondefault packet and window sizes, and closed user group.

carrier sense multiple access with collision detection (CSMA/CD)
A communications protocol defined by the IEEE 802.3 standard, which allows a station to sense whether or not a carrier is on a line (LAN cable) before transmitting data. A carrier is present only if another station is transmitting data. If a station senses no carrier, the station can transmit data. If a carrier is present, the station waits until it senses no carrier before it transmits data. If two stations transmit at the same time, a collision occurs. Each station waits for a calculated time period before attempting to retransmit. *Compare with contention.*

CCITT
See Consultative Committee for International Telegraph and Telephone.

CENLOG
See critical event notification and logging.

central processing unit (CPU)
A unit of a computer that includes the circuits that control the interpretation and execution of instructions. Sometimes called central processor.

Circuit-Switched Communications Software (CSCS)
A Unisys product that enables Telcon to communicate over an X.21 circuit-switched public data network (PDN).

circuit-switched network
A network that establishes a temporary physical circuit when it receives a connect request, and terminates the circuit when it receives a disconnect request.

CMS 1100
See Communications Management System

CMS 1100 network
One or more interconnected CMS 1100 nodes. Node interconnections may be direct or indirect. If indirect, it may be through an intermediate Telcon node or through a TCP/IP or OSI internetwork connection.

CMS 1100 node
A CMS 1100 program on an OS 1100 or Series 2200 host. Each host may contain more than one node; each node has a separate configuration file. .

command
An instruction or control signal that initiates a sequence of events.

Communications Management System (CMS 1100)
The software that manages all data communications into and out of OS 1100 host computers. CMS 1100 provides an interface between the OS 1100 and the Telcon/DCP.

communications session recovery
A Hot-Standby feature that automatically recovers terminal sessions after CMS 1100, Telcon, MCB, or the system fails.

communications system
The total environment in which Distributed Communications Architecture (DCA) controls logical structure as well as the interfaces and protocols. Logically, the communications system encompasses the transport network and all the connected termination systems, but not the attached TSUs and their end users.

communications system administrator (CSA)
A package of information-gathering facilities that enables the systems analyst or operator to acquire information about the operation of CMS 1100.

communications system user (CSU)
DCA Level I only. The applications-related control structure, external to DCA, that interfaces with the communications system through one or more ports. CSUs control one or more end users, directing data and commands to and from them. The standard Telcon CSU is called the Device Management Facility (DMF). CSUs in DCA Level I can be compared with TSUs in DCA Level II.

computer network
A set of one or more computing systems, communications facilities, and terminals interconnected to provide services to a set of users.

COMUS
Acronym for Computerized Onsite Maintenance of User System. COMUS is an OS 1100 processor that leads you interactively through the process of defining the configuration and parameters for the software products you want to install. For example, after you define a CMS 1100 configuration, COMUS calls the symbolic stream generator, which creates a runstream to generate a CMS 1100 configuration file. COMUS can also initiate generation of the Telcon software for DCPs in your network. (This term is not spelled out in Unisys documentation.) *See also* SOLAR.

COMUS Definition Language (CDL)
A language provided by COMUS that is used to change a default configuration.

configuration
The arrangement of a computer system or network, defined by the nature, number, and chief characteristics of its functional units.

configuration access packet (CAP)
A data structure used by the CFACCS (configuration access) portion of Telcon to obtain and communicate information about the Telcon configuration.

configuration commands
The instructions that dynamically modify the CMS 1100 configuration while CMS 1100 is executing. NEED MORE ADEQUATE DEFINITION.

configuration element
(1) In Telcon, the Telcon utility processor HCONFIG produces this element from the file containing Telcon network definition statements and uses it in Telcon system generation. This source element is converted into an omnibus element on the OS 1100 system and becomes a configuration file on the DCP using a portion of the download omnibus element. (2) In CMS 1100, the configuration element is sometimes called a configuration table or configuration table file. CMS 1100 uses it to manage data communications into and out of the OS 1100 host computer.

configuration file access routine (CFAR)
A set of routines that enables you to read the CMS 1100 configuration file.

configuration ID information table
A table that contains configuration names and ID numbers. The HCONFIG utility processor builds the table automatically. The table can be altered by online configuration.

configuration source file
A CMS 1100 or Telcon file that contains configuration information. CMS 1100 or Telcon reads this file during initialization to establish the initial operating environment and accesses this file during operation to obtain additional information.

connection
A logical communications path between two stations or users.

connectionless protocol
A protocol that transfers data without a preestablished logical connection betweensender and receiver. For example, dynamic network services (DNS) is a connectionless network layer protocol, whereas TS/TN requires that you configure logical connections (using the Telcon SESSN statement, or the CMS 1100 NETWORK or PORT statements). The OSI internetwork protocol is a connectionless network layer protocol for OSI networks.

console messages
The messages generated in the Telcon environment by Network Management Services (NMS) to inform you of errors, events, and activities that occur. NEED GENERIC DEFINITION; TOO PRODUCT-SPECIFIC.

console mode
An environment in which the experienced user can control the system.

Consultative Committee for International Telegraph and Telephone (CCITT)
An international advisory committee that establishes worldwide communications recommendations (standards) for use by telecommunications authorities. Voting members are nations, which often designate their PTT administrations as representatives, and other government groups, such as the National Institute of Standards and Technology (NIST). Non-voting members are often standards organizations. European term: Comite Consultatif International de Telegraphique et Telephonique.

critical event notification and logging (CENLOG)
The facility that enables Telcon to log error and nonerror events, and to generate warning messages to network administrator consoles for critical events. Logs are in a standard format and can be written to a disk file cataloged for this purpose. *See also*, PDS1100, CMMS, and CAP.

CSA
See communications system administrator.

cycle
A complete sequence of operations, at the end of which the series can be repeated.

cyclic redundancy check
A method for checking errors that reduces error rates to a minimum. The method involves using a cyclic redundancy check character to match check fields on the sending and receiving ends of a transmission. Any errors detected in this check are corrected. See also longitudinal redundancy check.

D

data link
An assembly of two or more terminal installations, and an interconnecting line.

data link layer
Layer 2 of X.25, which implements the HDLC protocol. Layer 2 processes layer 1 (physical layer) data to provide error-free point- to-point data communication. Layer 2 processes data acknowledgment frames sent back by a receiver, can reject erroneous frames it receives, and can retransmit rejected frames.

datagram
A self-contained package of data in a network carrying enough information to be routed from source to destination without reliance on earlier exchanges between source or destination and the transporting network.

DCA
See Distributed Communications Architecture (DCA). *Also* abbreviation for Defense Communications Agency, a function of the Department of Defense.

DCA Transport Protocol (DTP)
The layer 4 protocol that defines the set of rules to create and maintain an end-to-end communications path between computing systems or transport service users. It is responsible for establishing and ending connections, flow control, error recovery, message segmentation, and recombination of messages.

DCA transport protocol extension (DTPX)
The session layer protocol for DCA level II; also, the Telcon name for the session entity module that runs the DTPX protocol. DTPX implements data assurance.

DDN
See Defense Data Network (DDN).

Defense Data Network (DDN)
An X.25 LAN-based connectionless packet-switching network for the U.S. Department of Defense (DoD).

DDP-PPC
Distributed Data Processing Program-to-Program Communications.

Distributed Communications Architecture (DCA)
The Unisys architecture that draws together all aspects of the communications products by defining a set of logical concepts, protocols, interfaces, and guidelines that are used to design hardware, software, and network products.

Distributed Communications Processor (DCP)
A Unisys front-end processors that provides communications facilities for Unisys host computers.

distributed system
A group of connected, cooperating computers, where each computer does a portion of the total processing required by an application.

DTP
See DCA transport protocol (DTP).

DTPX
See DCA transport protocol extension.

E

electronic mail
Name given to the electronic generation, transmission, and display of business correspondence and documents.

end user
An operator, terminal, or program that can generate and receive data transmitted over a DCA system.

F

frame
A block of data link layer information, which consists of a flag, an address field, a control field, an information field, a frame check sequence field, and a flag. Packets (groups of network layer data) are transmitted in the information fields of data link layer frames.

front-end processor
A communications computer associated with a host computer. It may perform line control, message handling, code conversion, error control, and application functions, such as control and operation of terminals. Its functions may include those of a communications processor (nodal).

G

gateway
A means to convert the message protocol of one proprietary network to the format used by the protocol of another proprietary network. A gateway can be implemented in hardware or software.

H

HDLC
See high level data link control (HDLC).

high level data link control (HDLC)
A data link layer protocol defined by the International Organization for Standardization (ISO) that controls the flow and transmission errors of data being transmitted across a physical link. CCITT modified HDLC to create the link access procedure, balanced (LAPB).

host computer
By common usage, the term implies a medium-to-large central processing system attached to a network. Architecturally, however, there is no distinction between a DCA host and a DCA terminal, since both contain a termination system, although of vastly different powers. A computing system that is attached to a data transmission facility and executes programs on behalf of its users to provide communication and other services.

I

IEEE 802.3
A frame-based (bit-oriented) communications protocol that includes most of the functions of the lower three layers of the Open Systems Interconnection (OSI) Reference Model.

internetwork
A set of heterogenous networks interconnected by TCP/IP gateways.

interconnection
The process of linking computers, which enables them to send and receive electronic signals.

Internet address
A four-octet (32-bit) source or destination address composed of a network field and a host field. The latter may be further divided to include a local subnetwork address.

internet control message protocol (ICMP)
A collection of messages exchanged by IP modules in both hosts and gateways; reports errors, problems, and operating information.

Internet datagram
The package exchanged between a pair of IP modules, made up of an Internet header and a data portion.

internet protocol (IP)
The DoD protocol used for sending the basic unit of data, an IP datagram, through an internet.

interoperate
The successful exchange of data by computers. Computers interpreting and acting upon received data.

L

LAN
See local area network (LAN).

LCLASS
A configuration statement that defines a class of communication line.

LINE
A configuration statement used to specify a physical communications line. A LINE statement has parameters that define a particular port processor on a DCP, thus specifying the connection point in a network for a physical line and a line class, which specifies the protocol to be used on the line.

line module
The hardware in a DCP that terminates a serial communications line, host channel connection, and peripheral connections.

link
The physical interconnection between two nodes in a network. A link can consist of a data communications circuit or a direct channel (cable) connection.

link layer
See data link layer.

local area network (LAN)
A user-owned data communications network with high-speed communications capabilities. There are technical limits on LAN size.

M

MASM
Unisys OS 1100 Meta-Assembler. MASM is a general purpose assembly language processor.

N

network
A system of connected computers that interoperate.

Network Management Services (NMS)
The group of control and reporting functions that support network administration by people or programs.

NMS
See Network Management Services.

node
A point in a network, either at the end of a communications line (end node), or where two lines meet (intermediate node).

P

packet
A layer 3 block of information. Data is transmitted from station to station in packets. Transmission from point to point, forming stages of the route from station to station is handled by layer 2 blocks, called frames. Packets are transmitted through a network in the information fields of frames. Different types of packets are concerned with call set-up, flow and error control, and transmission of user data. Packets can request, accept, and clear calls, and can transmit or accept data or request retransmission. The data field of a layer 3 packet is used to communicate higher layer data and control information.

packet-switched network
A network in which data is transmitted in units, called packets. The packets can be routed individually over the best available network connection and reassembled to form a complete message at the destination.

packet switching
In a packet-switched network information is transmitted between DTEs in variable length blocks of data, called packets. Software maintains routing and data flow information tables for all active circuits, but data transmission hardware is shared between circuits.

PCF
Permanent correction file. A permanent file of various symbolic elements that provides a means to create and update a variety of symbolic elements.

PDN
See public data network (PDN).

peer protocol
A protocol that governs communication between program entities that have the same function in the same layer in each of two different open systems networks.

proprietary network
A network that uses protocols developed by only one vendor.

protocol
A set of rules governing network functionality.

public data network (PDN)
A network established and operated by a network administration or PTT whose sole purpose is to provide data transmission services to the public.

R

reassembly
The process of piecing together datagram fragments to reproduce the original large datagram. Reassembly is guided by fragmentation data carried in the datagram's IP headers.

receive not ready (RNR)
A command or response frame used by DCE or DTE to indicate a busy condition. A RNR command may be used by a DCE or DTE to ask for the status of the DTE or DCE, respectively.

receive ready (RR)
A supervisory frame used by a DCE or DTE to indicate that it is ready to receive an information frame and to acknowledge previously received information frames. An RR frame may also be used to indicate the clearance of a busy condition that was reported by the earlier transmission of an RNR frame. A DTE may use an RR frame to ask for the status of the DCE.

reliable transfer service
A software program that transfers information from the sender to the destination with a guarantee that the information will arrive intact.

relocatable element
An element containing a program part in relocatable binary format, suitable for combination with other relocatable elements to produce an executable program (absolute element).

remote concentrator
A communications computer that provides multiplexed communications ability to many low speed, often asynchronous lines, and one or more high speed, usually synchronous, lines. The remote concentrator may be polled by a computer system and may in turn poll terminals.

restart request
An X.25 control packet used to initialize the packet level and also to recover from severe packet level error conditions. Any existing network connections are terminated when a restart occurs. Restart request packets contain a cause code and a diagnostic code, which gives information about the reason for the restart action.

RNR
See receive not ready (RNR).

RR
See receive ready.

S

segment
The unit of data exchanged by TCP modules. The term may also be used to describe the unit of exchange between any transport protocol modules. A TCP segment maps into one IP datagram.

SESSN
A configuration statement that defines permanent logical Telcon channels between Telcon network service users.

service
A computer routine or program that performs computer maintenance and operations and prepares and corrects programs. Services are general purpose programs, such as debugging routines, executive routines, and diagnostic routines, and general input and output routines.

STATION
A configuration statement that defines the data link layer for an X.25 line.

symbolic element
An element containing information generally in human-intelligible format (typically card images). The most common usage of symbolic elements is as source language to be input to a language processor.

synchronous transmission
A transmission mode used on serial mode data circuits. A continuous pulse stream called a clock is provided to synchronize the transmitted and received bit stream, saving the channel capacity which is used for synchronization (start and stop bits) on asynchronous circuits.

T

TCP/IP
See transmission control protocol/internet protocol (TCP/IP).

TCP/IP gateway
A device, or pair of devices, that interconnect two or more networks or subnetworks, enabling the passage of data from one (sub)network to another. In this architecture, a gateway contains an IP module, a routing protocol module, and (for each connected subnetwork) a subnetwork protocol (SNP) module. A gateway is often called an IP router.

TELNET
A TCP-IP Stack protocol that provides a standard interface between terminals and processes.

transmission control protocol/internet protocol (TCP/IP)
A set of protocols developed by the Department of Defense (DoD) Advanced Research Projects Agency (ARPA) during the early 1970s. Originally designed to connect different kinds of networks and computers.

trunk
A set of one or more physical connections between two communication processors.

transport services
Software that provides a system foundation for message routing, device (such as printers) connection to the network, and transmission error detection.

TS
Termination system. This is a means by which a CSU is able to interface with another application system and make use of the communication system.

TS/TN
A combination of the terms termination system (TS) and transport network (TN). TS/TN is a connection-oriented protocol that handles the TS/TN interface. *See also* connectionless data protocol.

U

UDLC
See universal data link control.

universal data link control (UDLC)
A bit-oriented communications protocol, defined by Unisys, and used on data links such as trunks.

upper layer protocol (ULP)
Any protocol above IP or TCP in the layered protocol hierarchy that uses IP or TCP. This term includes transport layer protocols, presentation layer protocols, session layer protocols, and application programs.

V

virtual circuit
A path through the network over which X.25 data packets and control packets are exchanged. A virtual circuit is identified by logical channel numbers at each end of the connection. Synonymous with network connection.

W

WAN
See wide area network (WAN).

wide area network (WAN)
A public or private computer network serving a wide geographical area.

X

XEU
An external end user in a Telcon configuration statement.

X.21
CCITT recommendation that defines the protocol for communication between user devices and a circuit-switched network.

X.21 bis
CCITT recommendation that allows existing data terminal to access a digital network over telephone lines.

X.25
CCITT recommendation that defines the protocol for communication between packet-switched public data networks and user devices in the packet-switched mode.

Bibliography

DCP Series TCP-IP Stack Programming Reference Manual (7831 5561). Unisys Corporation.

DCP Series TCP-IP Stack TELNET User Guide (7831 5553). Unisys Corporation.

DCP Series Local Area Network (LAN) Platform Configuration and Operations Guide (7831 5512). Unisys Corporation.

DCP Series Telcon Configuration Reference Manual (7831 5686). Unisys Corporation.

DCP Series Telcon Configuration Guide (7831 5678). Unisys Corporation.

DCP Series Telcon Installation Guide (7831 5645). Unisys Corporation.

DCP Series Telcon Software Operations Guide (7831 5785). Unisys Corporation

DCP Series X.25 Packet Switched Communications Software (PSCS) Configuration and Operations Guide (7831 5470). Unisys Corporation.

General Index

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Parameters Index

K

L

M

N

P

R

S

T

U

V

X