Networks • Communications

Local Area Transport (LAT) Architecture

Network Manager's Guide

digital

software

Local Area Transport (LAT) Architecture

Network Manager's Guide

Order No. AA-DJ18B-TK

July 1985

The *Local Area Transport (LAT) Architecture Network Manager's Guide* is intended for network managers and system managers. It contains information about the LAT architecture. This guide also includes information for configuring and managing LAT networks.

SUPERSESSION/UPDATE INFORMATION: This is a revised manual.

digital

AA-DJ18B-TK
First Printing, July 1985

This manual was produced by Networks and Communications Publications.

Contents

3 LAT Performance Issues

4 LAT Troubleshooting

Examples

Figures

Tables

Preface

The *Local Area Transport (LAT) Architecture Network Manager's Guide* is a reference manual describing the Local Area Transport (LAT) architecture and the configuration guidelines that apply to server products and service nodes supporting this architecture. Also described are the performance issues inherent in LAT server products, and a chapter is provided on troubleshooting LAT networks.

Intended Audience

The LAT server products offered by Digital assume that four different types of users will be involved with these products:

- Terminal user – a user of one of the terminals attached to a LAT server
- Server manager – the person responsible for the server(s)
- System manager – the person responsible for setting up a service node to which the LAT servers will connect
- Network manager – the person responsible for the overall functioning of the local area network (LAN)

This manual is intended for the network manager described above. The manual may also be useful to both the system manager(s) and server manager(s).

Before reading this manual the reader should be thoroughly familiar with the material covered in one of the server operation guides listed in the next section.

Structure of This Guide

The *Local Area Transport (LAT) Architecture Network Manager's Guide* has four chapters:

Chapter 1 introduces the LAT architecture. This chapter gives an overview of the LAT layer structure and the services inherent in the LAT architecture. The chapter provides information necessary to understand the configuration parameters discussed in Chapter 2.

Chapter 2 describes the configuration parameters in LAT server products and explains the method for determining the desired values for these parameters.

Chapter 3 explains the performance issues involved in LAT server products and gives performance information. This performance information is related to the configuration parameters discussed in Chapter 2.

Chapter 4 contains a troubleshooting guide for LAT networks. This chapter contains troubleshooting procedures for diagnosing LAT network-related problems; for individual product problems, see the individual product documentation sets.

A glossary is included in the back of the manual. Some of the terminology used with LAT networks differs from that used with DECnet networks.

Associated Documents

The document sets supplied with each LAT server product describe in detail the hardware and software installation procedures, management procedures, and user procedures for the individual LAT server products. The products and the associated manuals are listed below.

LAT-11

- *LAT-11 Software Installation Manual*

  This manual describes the procedure for installing the LAT-11 software on the server hardware and, optionally, on the server's DECnet load host.

- *LAT-11 Manager's Guide*

  This guide contains the management procedures for the LAT-11 server, including a description of the privileged command set.

- *LAT-11 User's Guide*

  This guide contains a complete description of the user commands.

DECserver 100

- *DECserver 100 Terminal Server Site Preparation/Hardware Installation Guide*

  This guide describes the procedures for installing the DECserver 100 hardware.

- *DECserver 100 Terminal Server Software Installation Guide (op-sys)*

  This guide describes the procedure for installing the DECserver 100 software on the DECnet load hosts. In the title, *op-sys* is the name of the load host operating system.

- *DECserver 100 Terminal Server Operations Guide*

  This guide contains the complete operation procedures for the DECserver 100 product.

- *DECserver 100 Terminal Server User's Pocket Guide*

  This guide contains the operations information necessary for the DECserver 100 terminal user.

- *DECserver 100 Terminal Server Identification Card*

  This card is used to record the server's Ethernet address, DECnet node name, and serial number.

Ethernet Terminal Server

- *Ethernet Communications Server*
  Site Preparation and Planning Guide

  This guide covers the issues involved in choosing a site for the Ethernet Communications Server hardware.

- *Ethernet Communications Server*
  Installation Guide

  This guide covers the procedures for setting up the Ethernet Communications Server hardware.

- *Ethernet Communications Server*
  Operations and Maintenance Guide

  This guide contains information about maintenance procedures for the Ethernet Communications Server.

- *Ethernet Communications Server*
  Terminal Server Operations Guide

  This guide contains information about the operation of the Ethernet Terminal Server software that is down-line loaded into the Ethernet Communications Server.

- *Ethernet Communications Server*
  Terminal Server Software Installation Guide (*op-sys*)

  This guide contains information about the installation of the Ethernet Terminal Server software on the DECnet load host. In the title, *op-sys* is the name of the load host operating system.

- *Ethernet Communications Server*
  Terminal Server User's Pocket Guide

  This guide contains a brief description of the user commands available at an Ethernet Terminal Server terminal and some of the general functions of the server.

- *Ethernet Communications Server*
  Terminal Server Identification Card

  This card is used to record the server's Ethernet address, DECnet node name, and serial number.

In addition to these product manuals, further documentation can be found in the service node operating system documentation set which describes the use and operation of the LAT service node software.

The Ethernet is described in the following manual:

The Ethernet; A Local Area Network; Data Link Layer and Physical Link Layer Specifications, Version 2.0

Conventions Used in This Guide

This guide uses terminology that is standard for all LAT products.

This terminology is defined in the glossary at the back of this guide and is slightly different from that used with DECnet products.

The following graphic conventions are used in this manual:

Convention	Meaning
`dot matrix`	Dot matrix indicates examples of system output or user input. System output is in black; user input is in red.
UPPERCASE	Uppercase in commands and examples indicates that you should enter the characters as shown (enter either uppercase or lowercase).
italics	Italics in commands and examples indicate that either the system supplies or you should supply a value.
[ ]	Square brackets indicate that the enclosed text is optional. If there is more than one option, you can choose one and only one of the options. Do not type the brackets when you enter the command.
{ }	Braces indicate that the enclosed text is required and you must choose one and only one of the options. Do not type the braces when you enter the command.
(Key)	Indicates that you should press the specified key. (CTRL) indicates that you should press the (CTRL) key at the same time as the *x* key, where *x* is a letter. Note that unless otherwise specified every command line is terminated by pressing the (RET) key.

All numbers are decimal unless otherwise noted. All Ethernet addresses are given in hexadecimal.

NOTE

Generally, you can abbreviate command keywords to the first three characters or the number of characters that make the keyword unique.

1 LAT Architectural Overview

1.1 General Architecture

Local area networks (LANs) allow computing resources to be physically distributed throughout a facility. This distribution can often lead to a more cost-effective approach to computing as well as an increased responsiveness to individual user needs.

A typical LAN can have 1000 or more attachments on a single coaxial cable over one mile long. A potential problem in such installations is the limited bandwidth of the LAN. For this reason, communication architectures operating in this shared environment should use the bandwidth efficiently. The Local Area Transport (LAT) architecture has been designed to satisfy this goal. This chapter contains a brief overview of the LAT architecture.

The LAT architecture is a communications architecture used by Digital in many of its Ethernet products. Products using protocols defined by the LAT architecture may also make use of protocols defined by the Digital Network Architecture (DNA). For instance, many of the servers use the DNA Maintenance Operation Protocol (MOP) to down-line load themselves during server initialization. The protocols defined by the two network architectures are separate and were developed to satisfy different communication needs. They do not conflict with one another and can coexist on the same Ethernet network.

The LAT architecture makes the following assumptions about the underlying communication medium and the nature of a communication session:

- Communication is local to one logical LAN.
- Communication, once initiated, is basically under the control of one of the two communicating partners.
- The bandwidth of the medium is much greater than the bandwidth of a single communication session.

The architecture is based on a requester/provider model. Because of the asymmetric nature of the LAT protocol, one of the communication partners will always be the initiator of the communication. This allows the use of a much simpler protocol than is necessary in the more general case of symmetric communication.

This guide gives an overview of how the LAT architecture is used to provide LAN terminal connections, with terminal servers and service nodes as the building blocks. Terminal servers are special nodes with the dedicated function of providing terminal users with the ability to access services. Service nodes may be standard Digital systems (such as a VAX running VMS or a PDP-11 running RSX-11M-PLUS), or they may be special nodes called servers. Servers provide the service node front end software and hardware for computers (either Digital or non-Digital) that lack the LAT software or Ethernet interface hardware. Some servers provide the service node function in addition to the terminal server function. These servers may be service nodes, terminal servers, or both, depending on what functions they are supporting at any given time.

The topology of a typical LAT network is shown in Figure 1–1.

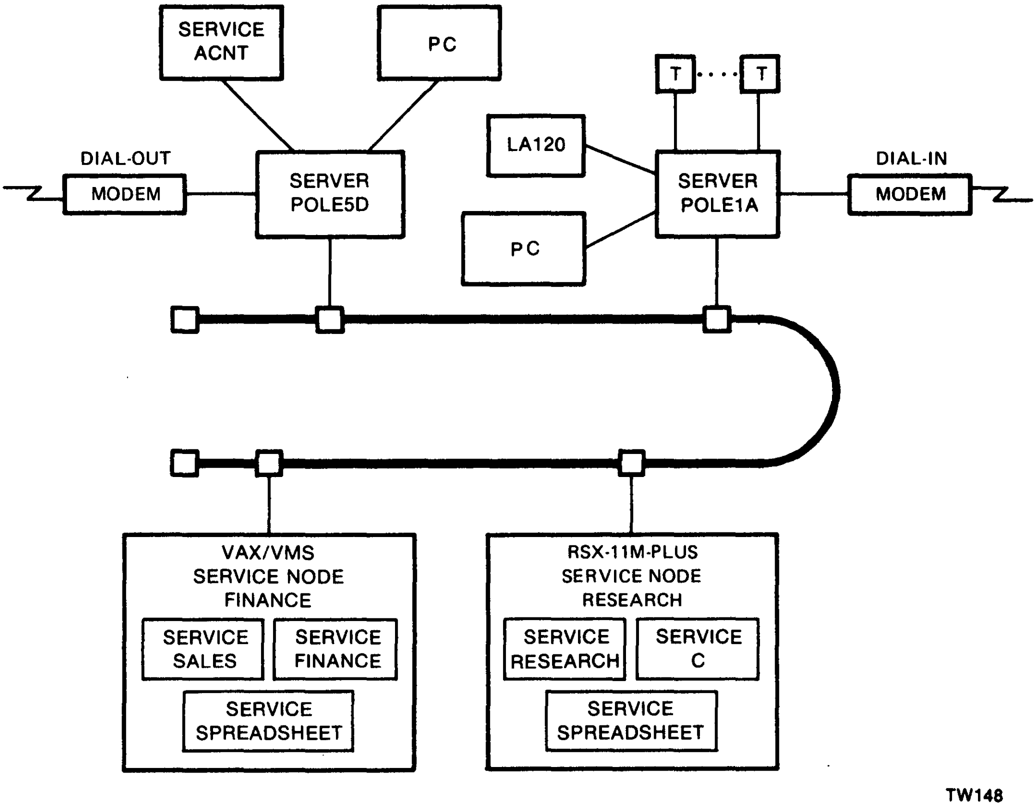


Figure 1–1: A Typical LAT Network

The terminal server is always the initiator of the communication session. The terminal server, on command from a terminal user, requests communication with one of several service nodes on the network. All of the terminal users on a terminal server can be connected to the same service node, or each user can be connected to different service nodes. Most terminal server implementations allow a single user to be connected to many service nodes at once.

The LAT architecture is a layered architecture. Like the Digital Network Architecture (DNA), the layers in an individual LAT node communicate with their counterparts in another node. This peer-to-peer communication is shown in Figure 1–2. The layers are shown in more detail in Figure 1–3. The Virtual Circuit layer and Slot layer are defined by the LAT architecture, whereas the Data Link layer and Physical Link layer are defined by the data link being used. All current LAT products use the Ethernet data link whose architecture is described in *The Ethernet; A Local Area Network*. The Network Management layer is usually not implemented in a dedicated node such as a terminal server, but is usually present in a nonserver service node where a sharing of the data link between multiple communication architectures is required.

The Virtual Circuit layer establishes and maintains shared virtual circuits between a terminal server and a service node. A virtual circuit is the logical data path between a terminal server and a service node. This data path includes the terminal server's network interface, the network hardware, and the service node's network interface. There is usually only one virtual circuit between any given terminal server and service node.

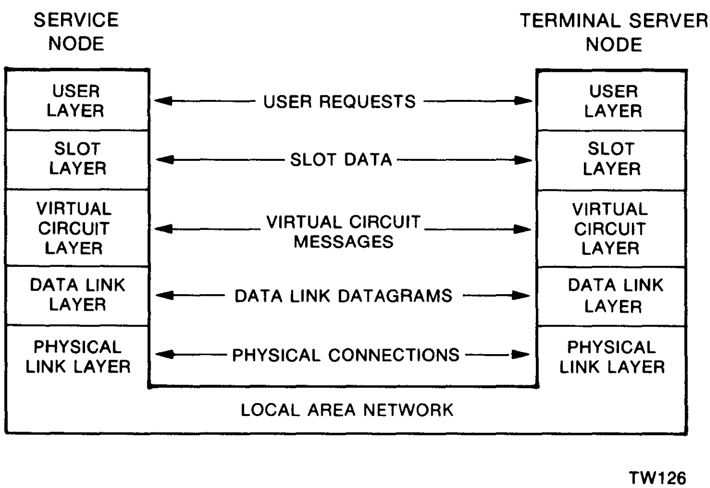


Figure 1–2: LAT Peer-to-Peer Communication

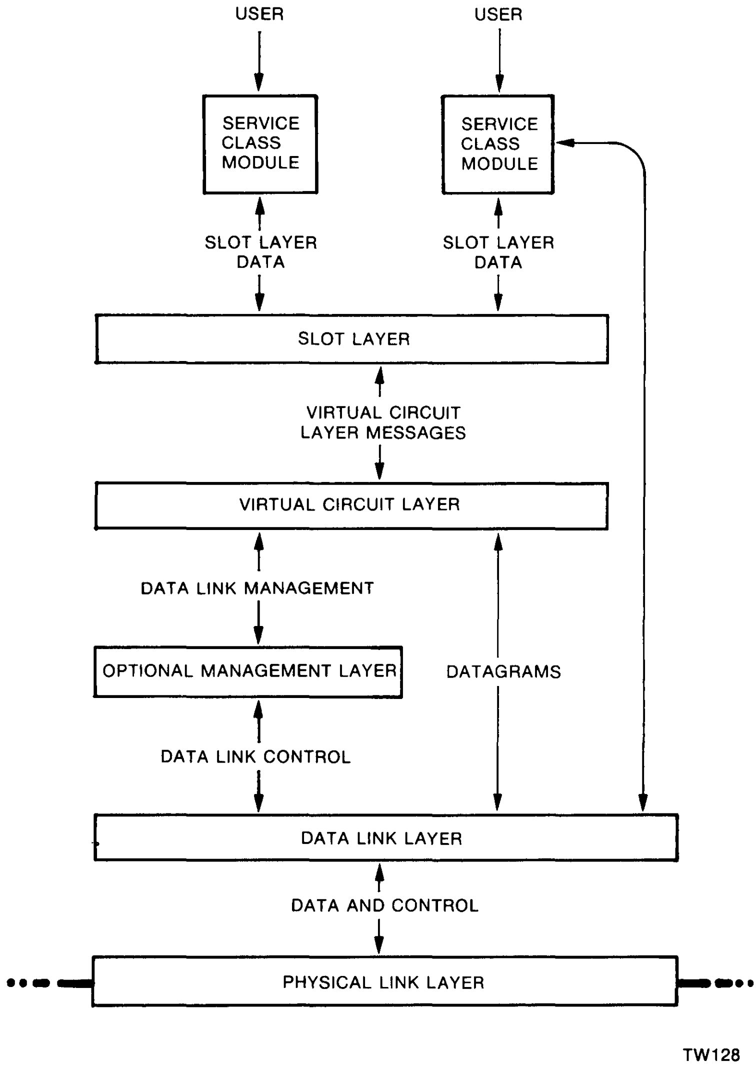


Figure 1–3: Layers in a LAT Network

The Slot layer establishes and maintains connections (also known as sessions) between terminal server users and the services offered by service nodes. A slot connection or session is a logical path between a user of the Slot layer and a service offered by a service node. The Slot layer multiplexes one or more users over the underlying virtual circuit set up by the Virtual Circuit layer. In this way, the overhead is minimized because only one circuit between the terminal server and a service node is needed.

The users of the Slot layer are Service Class modules. These modules add functionality to the Slot layer and frequently use other messages in addition to the standard Virtual Circuit layer messages. The service class for interactive terminals (Service Class 1) is used by all LAT terminal servers and LAT service nodes.

1.2 Virtual Circuit Layer

The Virtual Circuit layer resides between the Data Link layer and the Slot layer. It uses the interface provided by the Data Link layer to send and receive datagrams over the network, and provides an interface to the Slot layer for the transmission and reception of Virtual Circuit layer messages.

1.2.1 Virtual Circuit Layer Functional Description

The Virtual Circuit layer performs the following functions:

- Creates virtual circuits when the slot layer on a terminal server makes a request to a service node that does not currently have a virtual circuit associated with it. A virtual circuit is the logical path between a terminal server and a service node. The circuit provides a sound, reliable path of communication between the terminal server and the service node.
- Transmits Slot layer data using a timer for the transmission interval. This maximizes the amount of data sent per message. By sending only one message per interval, the utilization of the Ethernet and the load on the service node are kept more constant and predictable.
- Provides a "keepalive" function which maintains virtual circuits if no Slot layer data is exchanged for a period of time.
- Provides positive acknowledgment and message sequencing functions. This enhances the basic datagram functions provided by the Data Link layer.
- Provides automatic retransmission of unacknowledged messages.
- Stops virtual circuits automatically when all Slot layer usage is terminated.

1.2.2 Virtual Circuit Layer Messages

There are three types of Virtual Circuit layer messages:

START Message – used to create the virtual circuit.

Format:

DATA LINK HEADER	LAT HEADER	START DATA	DATA LINK CRC
14 OCTETS	8 OCTETS	VARIABLE	4 OCTETS

TW129

RUN Message – used to exchange Slot layer data on the virtual circuit.

Format:

DATA LINK HEADER	LAT HEADER	LAT SLOTS	DATA LINK CRC
14 OCTETS	8 OCTETS	UP TO 1492 OCTETS	4 OCTETS

TW130

STOP Message – used to terminate the virtual circuit.

Format:

DATA LINK HEADER	LAT HEADER	STOP DATA	DATA LINK CRC
14 OCTETS	8 OCTETS	VARIABLE	4 OCTETS

TW131

1.2.3 Virtual Circuit Layer Operation

The Virtual Circuit layer uses a synchronous, packet-oriented protocol consisting of three basic message types used in the manner described in the following paragraphs and summarized in Figure 1–4.

The terminal server initiates the virtual circuit by sending a START message to the service node in response to a request from the Slot layer. This message contains various terminal server parameters and information needed by the service node. If the service node approves of the virtual circuit connection request, the service node sends a START message back to the terminal server with its own parameters.

After establishing the virtual circuit connection, the virtual circuit goes into the "run" state, and from then on RUN messages are exchanged. At this point, the Slot layer is able to establish Slot layer connections and begin transmitting data, control information, and acknowledgments. These slot messages are packed into Virtual Circuit layer RUN messages by the Slot layer before it makes the transmit request to the Virtual Circuit layer. A single RUN message can contain slots for many different Slot layer connections, as long as all the connections are with same service node.

Virtual Circuit layer messages are exchanged on the basis of a circuit timer, not on the basis of when the message request is made. Only the terminal server maintains this timer. When this timer expires, the server transmits all messages which may be pending on every virtual circuit. With this approach, additional terminal server

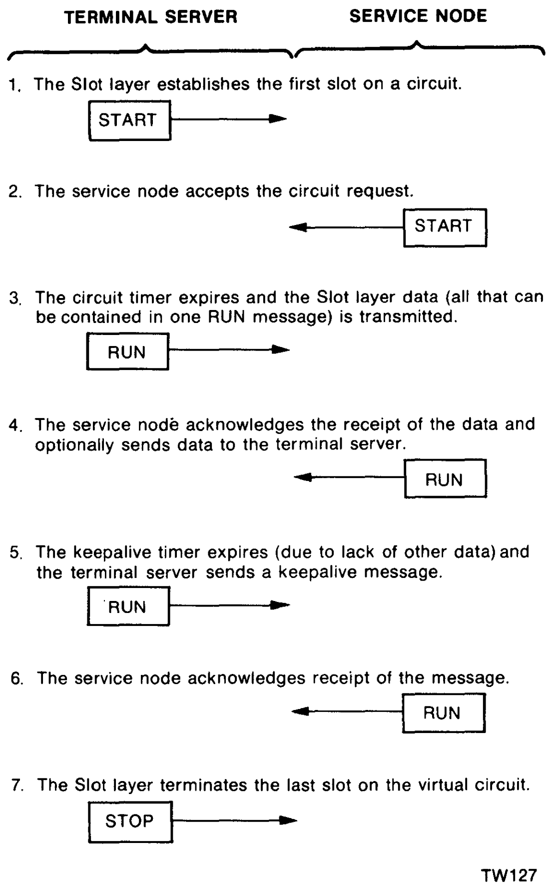


Figure 1–4: Virtual Circuit Layer Operation

users cause the message length to grow because of the presence of more slots in the RUN message. However, the maximum number of messages on the virtual circuit remains constant. If the number of slots is large enough, a second Virtual Circuit layer RUN message is required. This second RUN message is not transmitted immediately after the first; rather, it is held until the next circuit timer expiration. In this way, a more constant (that is, a predictably maximal) load is presented to the network.

Service nodes send messages when the service node has a message to send and has been requested to respond by the terminal server. The service node can also send one unsolicited message to the terminal server. This message is usually used in conjunction with the terminal server's keepalive timer (see Section 2.3.2) to provide a method of monitoring the status of the virtual circuit.

The service node and terminal server continue to exchange RUN messages until one or the other sends a STOP message. Either the service node or the terminal server can terminate the circuit by sending a STOP message. Usually the termination request comes from the terminal server and occurs when the last active slot on the virtual circuit is stopped. In this case, the terminal server sends a STOP message and the virtual circuit is disconnected.

1.3 Slot Layer

The Slot layer resides between the Virtual Circuit layer and the user or Service Class modules. It uses the interface provided by the Virtual Circuit layer to send and receive user-level messages over the virtual circuit.

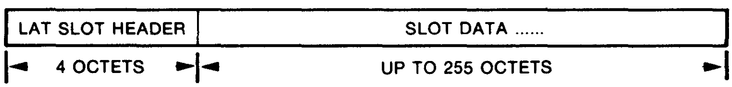
1.3.1 Slot Layer Functional Description

The Slot layer provides the following functions:

- Establishes Slot layer connections (called sessions) between terminal users and service nodes.
- Provides for multiplexing/demultiplexing of slot data over a single virtual circuit between a server and a service node.
- Provides two flow-controlled data channels and one out-of-band data channel for each session.
- Terminates sessions when requested by the Service Class.

1.3.2 Slot Layer Slot Formats

The Slot layer uses a protocol based on six types of slots. The general slot format is shown below:



TW132

START Slot – used by the terminal server to initiate a terminal session with a service node and by the service node to accept a session request.

REJECT Slot – used by the service node to reject a request for a session.

DATA-A Slot – used by both nodes to transmit flow-controlled data.

DATA-B Slot – used by both nodes to transmit flow-controlled data. This provides an alternate channel capability.

ATTENTION Slot – used by both nodes to transmit data that is not within the normal flow (that is, out-of-band data).

STOP Slot – used by either node to terminate the session.

1.3.3 Slot Layer Operation

The operation discussed in the following paragraphs is summarized in Figure 1–5.

After requesting the Virtual Circuit layer to establish a virtual circuit, the Slot layer requests a session with a service node using a START slot. The service node can accept the request with a START slot or reject the request with a REJECT slot. If the service node accepts the request, a session is established and terminal data is transmitted back and forth between the server and the service node using one or more of the three types of DATA slots.

The protocol provides for two flow-controlled data channels as well as one attention channel that is not flow-controlled. Data on the attention channel always has highest priority. Each of these data channels uses its own slot type. Usually one of the two flow-controlled data channels is used to carry normal user data and the other is used to carry special user data (such as parametric changes made by the user during a session). The two flow-controlled channels use a credit scheme to control data flow: each node can give credits to its session partner which allows the partner to transmit the specified number of additional messages.

Normal data flow continues until one of the session partners decides to terminate the session using a STOP slot. Normally this occurs when a user logs out of a service node or types a DISCONNECT command at the terminal. In response to this command, the Slot layer issues a STOP slot and the session is terminated. If the session is the last session on a virtual circuit, the virtual circuit is also terminated.

TERMINAL SERVER | **SERVICE NODE**

1 The terminal server user requests a service connection. This message includes initial flow control credit(s).

START ⟶

2 The service node accepts the connection request. The service node also includes initial flow control credit(s).

⟵ START

3 The terminal server user enters one or more characters which are stored in a data slot. If flow control credits are available, the slot is transmitted in a Virtual Circuit layer RUN message when the circuit timer expires. The terminal server decrements its flow control credit count for the session.

DATA ⟶

4 The service node processes the characters and possibly sends a response, usually with new flow control credit(s).
The service node decrements its flow control credit count for the session. The terminal server increments its flow control count for the session if the service node sent additional credits.

⟵ DATA

5 The service node sends some characters. This step may not be possible if the service node's flow control credit count for the session is zero.

⟵ DATA

6 The terminal server processes the characters and optionally sends response data from the user as well as possible flow control credit(s). The server decrements its flow control credit count for the session. The service node increments its flow control if the terminal server sent additional credits.

DATA ⟶

7 The service node requests disconnection due to the user logging out of the service.

⟵ STOP

TW133

Figure 1–5: Slot Layer Operation

1.4 Service Classes

The Service Class modules use the interface provided by the Slot layer to transmit and receive slots and provides an interface to the user to handle user-level requests. These modules are often not a physically separate layer, but rather are included in the Slot layer. Each service class can define additional fields in the slots and may define additional datagram messages which are transmitted by issuing a request directly to the Data Link layer. One service class may include the features of another service class.

1.4.1 Service Class 1 Functional Description

Service Class 1 is the service class used for interactive terminals. This service class also provides for a directory service. The directory service is a two-level name space which provides names for service nodes and names for services on these nodes. A service node is known by a single node name but can provide multiple services. In addition, a service node associates a rating value with each service name. For more information on rating values and how they are used, see Section 2.3.8.

Service Class 1 provides an additional feature called group codes. This feature allows the logical partitioning of the network into smaller networks. Each service node and terminal server can specify groups to which they wish to belong (some terminal servers only allow individual terminals this ability, some allow both server-wide and terminal-specific group specification). Only components in the same group are capable of establishing sessions. For more information on group codes, see Section 2.3.4.

1.4.2 Service Class 1 Messages

Service Class 1 defines additional fields in the Slot layer messages and implements a service announcement message that provides the directory service. The announcement message has fields for the node to specify its node name, the group codes currently enabled for the node, the service names and ratings currently available at the service node, and the service classes supported by the node. The format of the service announcement message is shown as follows:

NODE NAME	GROUP CODES	SERVICE NAMES AND RATINGS	SERVICE CLASSES

TW134

1.4.3 Service Class 1 Operation

A service node periodically sends out its service announcement message using a multicast address to which all terminal servers listen. This multicast message contains a node name, group codes, service names, and service ratings for the service node. This message is directly transmitted by the service class module and is not considered a Slot layer or Virtual Circuit layer message, because neither a virtual circuit nor a slot connection is needed for the message to be received by a terminal server. Using these multicast messages, the terminal servers build up a directory of node names and service names. The node names are used by the Virtual Circuit layer and the service names are used by the Slot layer. On some service nodes, if no service names are set up, the node name is the default service name.

When a terminal user requests a connection, the connection request specifies a service name. The terminal server's Slot layer uses the service name to look up the service node offering the requested service. If more than one node offers the same service, the terminal server decides which node to use based on a rating parameter. This rating value is set up on the service nodes by the service node managers or automatically calculated by the service node's LAT software (see Section 2.3.8). The directory operation is summarized in Figure 1–6.

1.4.4 Supporting Service Class 1 Services on a Server

Some of the servers offered by Digital provide the functions of a terminal server and a service node simultaneously. Using these servers, the server can support terminals and also offer services such as non-Digital computer systems or dial-out modems. This allows the server to act as a front end communications processor for computer systems that have either no LAT service node software or no Ethernet connection hardware available to them. Although the server can have both service node and terminal server connections, any individual connection is always established as one or the other.

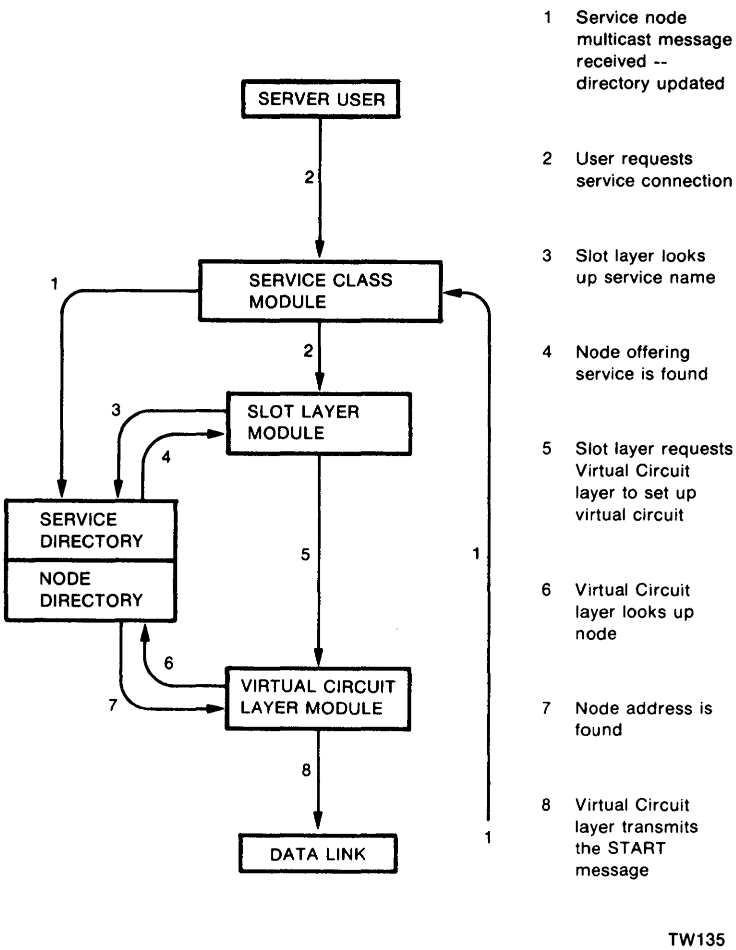


Figure 1–6: Service Class 1 Directory Operation

2
LAT Configuration Guidelines

2.1 Introduction to LAT Control

Control of LAT occurs at the service node and at the terminal server. At the service nodes that are running Digital-supplied operating systems and LAT software, LAT operation is controlled by a management utility program which interfaces with the operating system's LAT port driver and LAT database. An example of this interaction (taken from VAX/VMS) is shown in Figure 2–1. On service nodes that are servers, LAT service node operation is controlled by the server's command interface.

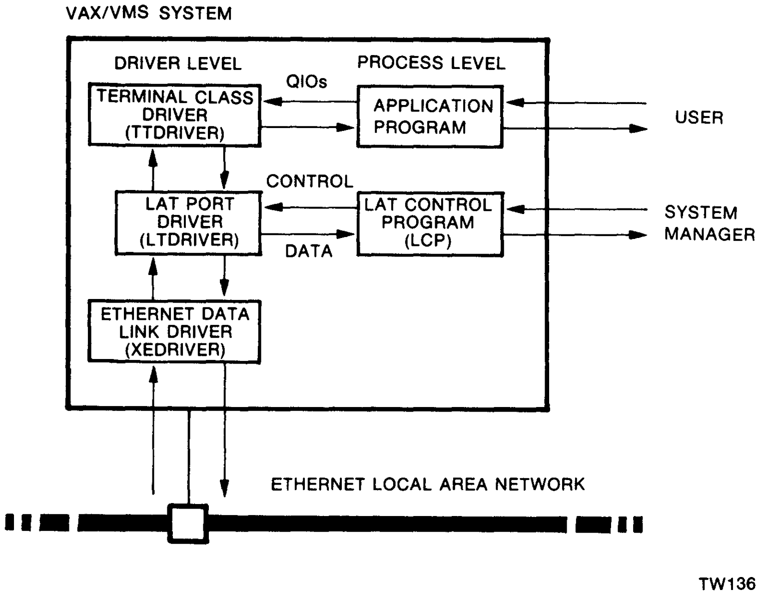


Figure 2–1: LAT Terminal Driver Control on a Service Node (VAX/VMS)

In the example shown in Figure 2–1, the management utility is the program LCP. The network manager uses LCP commands to control the operation of the LAT port driver (LTDRIVER). LTDRIVER provides an interface between the Ethernet data link driver (XEDRIVER) and the VAX/VMS terminal class driver (TTDRIVER). Application programs can make generic terminal QIO requests to the TTDRIVER without having to know whether the terminal is connected via an LAT terminal server. Other operating systems supporting LAT network terminals use a similar approach in their implementations. The name of the management utility and the syntax of the commands may vary from one service node to another; however, all service nodes have some form of control over most of the parameters mentioned in this chapter.

The terminal servers also have a control capability. The control functions in the terminal server are accessed by using a set of privileged commands on a privileged terminal on the terminal server. Any terminal on a terminal server can be used in this fashion if the user knows the password necessary to change a terminal to privileged status.

Control of LAT nodes falls into four categories:

- Starting and stopping LAT operation (see Section 2.2) – Service nodes allow managers to start and stop the LAT protocol. Terminal servers have a method for loading the control program and beginning network operation.
- Setting control characteristics (see Section 2.3) – Service nodes allow managers to set parameters at startup time and/or while running. Terminal servers have a privileged command capability that allows a terminal server manager to modify control characteristics.
- Showing operating characteristics (see Section 2.4) – Service nodes and terminal servers have one or more SHOW commands to allow the manager to verify control parameter settings.
- Accessing counter information (see Section 2.5) – Service nodes and terminal servers also have a SHOW COUNTERS command to allow managers to access the counters kept by the LAT software and the Ethernet data link. There is also a method for resetting these counters.

2.2 Starting and Stopping LAT Operation

LAT operation at the service node is started by invoking the control program and performing a startup sequence. On most service nodes this procedure consists of issuing a START command, possibly with optional parameters. After issuing this command the service node and any associated service names are made known to the network (via a multicast message; see Section 1.4.3) and are available to terminal server users. Service nodes that are servers usually start their service node operation whenever the first service name is defined and stop their service node operation whenever the last service name is deleted.

LAT operation at the terminal server is started by booting the terminal server. This boot procedure may require the network and a DECnet load host, or it may be done by loading from a local device. The procedure for booting specific LAT terminal server products is described in the individual product documentation sets.

2.3 Setting Control Parameters and Characteristics

The following sections discuss the control parameters of LAT products. Some of these parameters are set on service nodes, some on terminal servers, and some on both. The format of the discussions includes the usage of the parameter, the range of permissible values, the default value, and a general description of the parameter. Also indicated are any restrictions on the values the parameter can have.

2.3.1 Circuit Timer

Usage: Both service nodes and terminal servers

Range: 10–200 milliseconds

Default: 80 milliseconds

Description: The circuit timer defines the interval between Virtual Circuit layer messages sent from the terminal server to the service node.

The Virtual Circuit layer RUN messages sent from the terminal server are sent solely on the basis of the circuit timer and not because of characters typed at the terminals. A message is always held until the circuit timer expires, even if multiple messages are required to deliver all the individual terminal session (Slot layer) data. In cases where multiple messages are required, only one message per circuit timer is transmitted. In this way, a more constant load is presented to the network.

The selection of a circuit timer value represents a trade-off between terminal response time and service node performance. A short circuit timer value gives better terminal response time at the expense of increased service node loading. A long circuit timer value gives slower terminal response time but improves service node loading. A short circuit timer also causes more loading on the network because of the increase in message traffic. In general, a longer circuit timer should be used on a heavily loaded service node, whereas a shorter circuit timer should be used on a stand-alone or lightly used service node.

If the terminal server is used primarily for file transfer (from more than 8 personal computers, each using a line speed of 9600), then the circuit timer may have to be set to a shorter time interval. This may be necessary because the terminal server can transmit only one message per circuit timer interval. If the circuit timer is too long and the terminal server is handling many file transfers at the same time, a backlog of messages occurs.

The actual circuit timer value is set up at the terminal server. However, a service node manager may have the option of defining a range of circuit timer values that the service node will accept. If a terminal server makes a connect request with a circuit timer value outside this range, the service node may deny the request.

The default value of 80 milliseconds provides a good balance between terminal response time and service node loading. Performance data for various circuit timer values is given in Chapter 3.

Restrictions: A service node manager must be careful, when setting the circuit timer value, not to exclude terminal servers with higher or lower values unless this is explicitly intended. A terminal server manager should be sure when setting the value that all intended service nodes will accept the value given. With many implementations, the timer precision is such that minor changes to the value have no effect. Note that not all implementations allow the timer value to be set as low as 10.

2.3.2 Keepalive Timer

Usage: Terminal servers only

Range: 10–255 seconds

Default: 20 seconds

Description: The keepalive timer defines the interval between idle RUN messages sent out by the terminal server to service nodes on active circuits. These messages provide a method for the Virtual Circuit layer to continually monitor the status of all active circuits. If an idle message is not acknowledged, the Virtual Circuit layer notifies the Slot layer of a suspected "circuit down" event.

The value used for the keepalive timer is a trade-off between fast "circuit down" detection and unnecessary traffic flow on the network. The default value represents a good compromise value. Increase the value for heavily loaded networks.

Restrictions: None

2.3.3 Retransmit Limit

Usage: Both service nodes and terminal servers

Range: 4–255 retransmission attempts

Default: 10 for terminal servers, 60 for service nodes

Description: The retransmit limit defines the number of times a message is retransmitted before the virtual circuit is declared "down."

The value chosen depends on the traffic load of the network and the quality of the network. If traffic load is heavy, or if the network is experiencing "noise" problems, you should set the value higher than the defaults. For more rapid error detection, you may want to specify a lower value than the default.

Restrictions: Some implementations have this parameter permanently set.

2.3.4 Group Codes

Usage: Both service nodes and terminal servers

Range: 0–255

Default: Group code 0 enabled

Description: Group codes are used to partition effectively a single network into smaller networks. Using group codes, the terminal server user's view of the network can be restricted to a smaller group of service nodes than the entire network.

On the service node, setting a group code means that the node is to be included in the group of nodes sharing this common code. A service node can have many different group codes set and, therefore, be in many different groups at the same time. The terminal server software finds out what groups a service node is in by reading the multicast configuration message sent out periodically by the service node.

On the terminal server, setting a group code means that the group is available to the terminal server as a whole and/or to an individual terminal (depending on how the terminal server implements group codes). The terminal user sees only those service nodes that match the group codes set for the terminal the user is on. When a terminal server user requests a session, at least one common group code must be defined for both the terminal and the desired service node name; otherwise the request fails.

On both nodes, if no group codes are explicitly set up, the node is, by default, a member of the zero (0) group. In cases where all of the network nodes use this default group, the network is not partitioned and all service nodes are available to all terminal servers and terminal server users.

Restrictions: Some implementations restrict the number of group codes that can be set. Not all terminal servers allow both the setting of server-wide group codes and terminal-specific group codes.

2.3.5 Datagram and Slot Size

Usage: Both service nodes and terminal servers

Range: 576–1518 octets for datagram, 1–255 octets for slot

Default : 1518 for datagram, 127 for slot

Description: The datagram and slot size parameters determine the size of the individual slots and the size of the entire datagram. A slot is used for individual terminal data. These slots are then bundled together into a Virtual Circuit layer RUN message and sent to the common service node. The datagram size includes the Ethernet header (14 octets), the LAT Virtual Circuit layer message header (8 octets), plus the slot data in the message (up to 1492 octets, with each slot having a 4-byte header). The datagram is followed by an Ethernet trailing CRC sequence consisting of 4 octets.

On most nodes, these values can be left at the default setting. However, under conditions where memory is at a premium, you may wish to decrease these values to save on use of valuable memory (usually some sort of operating system "pool"). For information on memory usage in specific systems, see the manager's guide provided for the service node's control program. The actual slot and datagram size used may be subject to negotiation between the communication partners. For this reason, the parameters frequently represent maximum sizes and not absolute sizes.

Restrictions: Most implementations have these parameters permanently fixed in the software. Other implementations may allow you to change only the maximum or minimum for these sizes.

2.3.6 Node Names and IDs

Usage: Service nodes only

Range: ASCII string (1–16 characters) for node name

ASCII string (1–64 characters) for node identification

Default: None

Description: Node names allow a service node to be known by more than just a physical network address.

Node identification strings allow a short informational ASCII string to be associated with each node name. This message can be used as a short news string or provide additional node identification.

Restrictions: Care must be taken to coordinate the node names throughout the network to avoid duplicate node names. If DECnet is present on the service node, it is highly recommended that the DECnet node name be used as the LAT service node name.

2.3.7 Service Names and IDs

Usage: Service nodes only

Range: ASCII string (1-16 characters) for service name

ASCII string (1-64 characters) for service identification

Default: None

Description: Service names allow a service node to be known by more than just a node name. A service node can set up several service names and be known to the terminal servers by these service names. If multiple service nodes set up the same service name, then a sharing of this service name is done based on ratings (see Section 2.3.8).

This sharing technique is often used in VAXclusters to allow the cluster to be known by a cluster name as well as by its individual node names. Using this technique in conjunction with the DYNAMIC rating value (see Section 2.3.8), a cluster can be set up that evenly distributes terminal users over the entire cluster as users connect to the cluster service name. Sharing of a service name is not limited to VAXclusters; this technique can be used with any collection of nodes offering the same service.

Service identification strings allow a short informational ASCII string to be associated with each service name. This message can be used as a short news string or to provide additional service identification.

Restrictions: Care must be taken to coordinate the service names throughout the network to avoid duplicate service names where they are not intended.

2.3.8 Service Ratings

Usage: Service nodes only

Range: 0–255 and DYNAMIC

Default: DYNAMIC

Description: A rating value can be applied to every service name. This value is used by the terminal servers to choose between two or more nodes offering the same service name. The terminal server usually chooses the service with the **highest** rating when multiple service nodes offer the same service.

A rating of DYNAMIC causes the service node to supply a dynamic rating that depends on the current usage level of the service. This dynamic rating usually has a fixed range of values and is calculated using a service-node-specific algorithm. All current service node implementations use this form of the service rating.

Restrictions: The rating values do not represent an absolute ranking; the terminal server takes the rating values into consideration when establishing a session; however, it may also use other information, such as the fact that a virtual circuit may already be established to a lower rated service.

2.3.9 Maximum Circuits

Usage: Both service nodes and terminal servers

Range: 1–255

Default: Implementation dependent

Description: The maximum circuits value specifies the maximum number of virtual circuits that are allowed to exist at the same time. Control over this parameter is provided to allow the manager to set an upper limit on node resources needed for the LAT virtual circuits.

Restrictions: Many implementations have this parameter permanently set in the software. Some implementations allow the value to be changed only when the software is first installed on the system.

2.3.10 Maximum Connects/Sessions

Usage: Both service nodes and terminal servers

Range: Implementation dependent

Default: Implementation dependent

Description: The maximum connects/sessions value specifies the maximum number of terminal sessions that are allowed to exist at the same time. Control over this parameter is provided to allow the manager to set an upper limit on the resources needed for the LAT sessions.

Restrictions: Many implementations have this parameter permanently set in the software. Some terminal server implementations provide this feature on a per-terminal basis. Some implementations also provide this feature on a per-circuit basis.

2.4 Showing Operating Characteristics

All LAT terminal servers and LAT service nodes provide some form of a SHOW command to display the current parameter settings.

On the servers, the command is the SHOW SERVER command. A typical display produced by a SHOW SERVER command (for the DECserver 100 terminal server) is shown in Example 2–1. The actual display produced by a terminal server may vary from the display shown.

```
Address:   AA-00-03-49-F1-00      Uptime: 2 20:46:43
Name:      Marketing Pod          (Marketing Pod)
Location: BLDNG 4, SECT 2         (BLDNG 4, SECT 2)
Number:           175             ( 175)
Circuit Timer:     80             ( 80)
Keep_alive Timer:  20             ( 20)
Console Port:       1             (1)
Software ID: PS0801ENG            (PS0801ENG)
Login Limit:        2             (2)
Dump:              ENA            (ENA)
Heartbeat:         ENA            (ENA)

Load Host: AA-00-03-F3-C1-05      (TOOK)
```

Example 2–1: Typical SHOW SERVER Display

On service nodes, the command is usually the SHOW CHARACTERISTICS command. A typical display produced by a SHOW CHARACTERISTICS command is shown in Example 2–2. The actual display produced by a service node may vary from the display shown.

```
Node name = \DEVELP\                 Service name = \DEVELP\
Node Identification = \DEVELP -- Systems Software Development\
Service Identification = \DEVELP -- Systems Software Development\
Groups = (0)
Multicast timer = 20 seconds
LAT Version = 5.0                    LAT Protocol is active
```

Example 2–2: Typical SHOW CHARACTERISTICS Display

2.5 Accessing LAT Counters

All LAT products keep counter information. These counters provide information such as the number of bytes transmitted, the number of transmission errors, and so on. The counters are shown using the SHOW COUNTERS, SHOW NODE, and SHOW SERVER commands.

When using the counters for traffic analysis or error detection it is often desirable to be able to reset (or zero) the counters. The ZERO COUNTERS command is provided for this purpose. Many of the counters are actually kept by the data link driver or data link hardware. In some implementations the ZERO COUNTERS command may not zero all of these counters. For more information on your specific product, see the operation or management guide for your terminal server product or the manager's guide for your service node.

The format of the counter display may vary from implementation to implementation. A typical terminal server counter display (from the DECserver 100 Terminal Server) is shown in Example 2–3. A typical terminal server node display (from the DECserver 100 Terminal Server) is shown in Example 2–4.

```
                              * ETHERNET COUNTERS *

Seconds Since Zeroed:          1223     Excessive Collisions:           0
Bytes Received:             1728681     Carrier Check Failure:          0
Bytes Sent:                  789753     Frame Too Long:                 0
Frames Received:              12891     Heartbeat Absent:               0
Frames Sent:                  11627     Late Collision:                 0
Multicast Bytes Rcv'd:         1781     Data Underrun:                  0
Multicast Bytes Sent:           910     Block Check Error:              0
Multicast Frames Rcv'd:         178     Framing Error:                  0
Multicast Frames Sent:           10     Data Overrun:                   0
Frames Sent, Deferred:          112     System Buffer Unavailable:      0
Frames Sent, 1 Collision:        41     User Buffer Unavailable:        0
Frames Sent, 2+ Collisions:      10

                              * SERVER COUNTERS *

Messages Received:            11472     Duplicates Received:            4
Messages Transmitted:         10359     Illegal Messages Rcv'd:         0
Messages Retransmitted:          22     Illegal Slots Rcv'd             0
                                        Duplicate Node Count            1
```

Example 2–3: A Typical SHOW COUNTERS Display

```
LAB01                    32 Connected    LAB01 -- VAX 11/780 System

Physical Address: AA-00-01-2C-1E-01  Seconds Since Zeroed:        2359
Messages Received:             2872  Duplicates Received:           11
Messages Transmitted:          2827  Illegal Messages Rcv'd:         0
Messages Retransmitted:          12  Illegal Slots Rcv'd:            0
                                     Duplicate Node Count:           1
```

Example 2–4: A Typical SHOW NODE Display

An explanation of the counters is given in Table 2–1. The counter names used in Table 2–1 are the architectural names for the counters; these names may vary slightly from the names used in individual terminal server product displays. The use of error counters is covered more fully in Chapter 4.

Table 2–1: LAT Counter Descriptions

These counters are maintained by the LAT layers:

Counter Name	Counter Description
Seconds Since Last Zeroed	The number of seconds since the counters were last zeroed.
Messages Transmitted	The number of Virtual Circuit layer messages transmitted.
Messages Received	The number of Virtual Circuit layer messages received.
Messages Retransmitted	The number of Virtual Circuit layer messages that had to be retransmitted.
Duplicate Messages Received	The number of Virtual Circuit layer messages with the same sequence number that were received more than once.
Illegal Messages Received	The number of Virtual Circuit layer messages received which had an illegal format. These messages cause the virtual circuit to be terminated.
Illegal Slots Received	The number of slots received which had an illegal format. These slots cause the underlying virtual circuit to be terminated.
Duplicate Node Count	The number of times the terminal server received a directory multicast message that assigned a different physical node address to a node for which the terminal server already had a defined physical address.

(continued on next page)

Table 2–1: LAT Counter Descriptions (cont.)

These counters are maintained by the Data Link layer:

Counter Name	Counter Description
Total Bytes Received	The number of bytes received over the data link without any error.
Total Bytes Sent	The number of bytes sent over the data link without any error.
Total Frames Received	The number of frames received without error.
Total Frames Sent	The number of frames sent without error.
Total Multicast Bytes Received	The number of multicast bytes received.
Total Multicast Frames Received	The number of multicast frames received.
Total Frames Sent (Deferred)	The number of frames sent without error after being initially deferred.
Total Frames Sent (Single Collision)	The number of frames sent without error after a single collision and backoff.
Total Frames Sent (Multiple Collision)	The number of frames sent without error after more than one collision and backoff sequence.
Send Failures	The number of frames that were not transmitted because of a transmit error. Transmit errors include excessive collisions, short circuit, open circuit, frame too long, and carrier check failed. Collision detect errors (heartbeat failures) are usually kept in a separate counter.
Receive Failures	The number of frames that were lost because of a receive error. Receive errors include block check error, framing error, and frame too long. Frames received for disabled multicast addresses and/or protocol types are usually kept in a separate counter. In addition to the receive errors specified, a count is usually maintained of the number of times the hardware and software could not keep up with receive data.

3 LAT Performance Issues

3.1 Terminal Server Performance Issues

On the terminal server, performance is measured by the response time of the terminal server's terminals and the throughput of these terminals.

3.1.1 Response Time

The response time of the individual terminals on the terminal server is probably the single most important performance issue from the point of view of the terminal user. The response time is almost entirely dependent on the circuit timer value. As explained in Section 2.3.1, the circuit timer can be varied at the terminal server to adjust the terminal response time and the load on the service nodes.

For normal interactive functions the circuit timer should be set at the default value of 80 milliseconds. This value will give an average response time in the range of 45 to 125 milliseconds, depending on whether the service node's terminal driver or applications software does the echoing. Expect response times of 0.5 times the circuit timer plus propagation delay when the service node's terminal driver does the echoing. With the recommended circuit timer of 80 milliseconds, this response time will average 45 to 50 milliseconds.

For application programs with the read-with-no-echo function, the character echo response time will average at least 1.5 times the circuit timer, plus propagation delays, plus application program delays. With the recommended circuit timer of 80 milliseconds, this response time will average at least 120 milliseconds. For certain applications, such as EDT, which issue multiple write I/O functions for each character typed (includes screen position information), the response time may be longer.

3.1.2 Throughput

The throughput of a terminal on a terminal server is largely dependent on the terminal speed. Higher throughput values are attained with higher terminal speeds. Although the user of a LAT terminal should see roughly the same throughput as a user of a directly connected terminal, a heavily loaded Ethernet may decrease the observed throughput.

The observed throughput may also be decreased when the amount of terminal data the terminal server must handle in one circuit timer interval exceeds the capacity of one Virtual Circuit RUN message (1492 bytes minus Slot layer overhead). An example of this would be 9 terminals running at 9600 bits per second (100% line utilization) and a circuit timer of 80 milliseconds.

The aggregate input throughput is usually significantly less than that attainable for output throughput because the terminal server has to check the input stream for special characters, such as the switch characters and XON/XOFF flow control characters.

3.2 Ethernet Utilization Issues

Ethernet utilization is an important issue in determining the capacity of the network to handle multiple terminal servers and multiple service nodes. The nature of the LAT protocol is such that utilization of the network bandwidth is optimized when a high number of terminals are in use on each of the servers. If only one or two terminals are in use at each server, the LAT protocol accounts for a higher proportion of the total Ethernet usage. Network utilization is also affected by the circuit timer value: higher timer values result in lower utilization values; lower timer values increase utilization values.

3.3 Service Node Performance Issues

The service nodes that support LAT should experience a decrease in terminal driver load. This decrease is due to the amount of information conveyed per hardware interrupt. In the case of a DZ11, only one character can be conveyed per interrupt. In the case of LAT, many characters from many terminals may be conveyed with a single interrupt from the Ethernet interface. Preliminary estimates indicate that the service node performance for LAT terminals is comparable to or slightly better than that of terminals connected directly through a DMF32 device. LAT terminals provide service node load savings of up to one and one-half times the character interrupt load of DZ11 terminals.

To achieve these performance gains, the configuration parameters must be set up properly. The primary configuration parameter affecting service node performance is the circuit timer value. While the service node cannot set the circuit timer, it may be able to declare a range of circuit timer values that it will accept. The higher the circuit timer, the better the service node performance will be. Any gain achieved by setting the circuit timer higher must be weighed against the increase in response time that will be seen at the terminals. The slot size and datagram size also affect performance; smaller sizes lead to more protocol overhead and decreased overall performance.

3.4 Configuring the LAT Network for Maximum Performance

This section describes a set of configuration guidelines that help you to build LAT networks with maximum performance. All the guidelines presented below are optional; however, failure to follow these guidelines may result in unnecessary performance degradation.

- Minimize the number of terminal servers present on the network. Wherever possible, utilize one terminal server completely before attaching another one. This decreases the amount of Virtual Circuit layer message traffic by minimizing the number of different virtual circuits.
- Minimize the number of service nodes that are accessed from any one terminal server. Plan terminal server use so that in most cases the terminal server is not accessing a different service node for each user. All terminal server users connecting to different service nodes uses much more of the data link bandwidth than all terminal server users connecting to one service node.
- Use group codes to organize service nodes and terminal servers in such a way as to minimize the number of services seen by any one user. If all service nodes and all terminal servers of a large network are in the same group, then the terminal server's SHOW SERVICES command displays too many services to be useful to users.
- Set the circuit timer to the default value of 80 milliseconds. Gradually increase this value until terminal server users detect a noticeable delay. However, if the terminal server is primarily used with personal computers that are performing file transfer operations (more than 8 computers each operating at a line speed of 9600), the circuit timer should be DECREASED until the service nodes experience an unacceptable interrupt load.

4 LAT Troubleshooting

This chapter describes various troubleshooting techniques you can use to help you diagnose a LAT network that is experiencing problems. Before you use these procedures, follow the procedures for diagnosing your specific server product as described in your individual terminal server documentation. The procedures described in this chapter assume you have already fully tested the individual servers and have not found the source of the problem.

4.1 Using LAT Counter Information

Nodes implementing the LAT protocol keep counter information both for the LAT protocol in particular and the data link in general. LAT counters maintain counter information for LAT users only, while data link counters may include other users of the data link (such as DECnet).

The LAT counters include two that can be used for error detection: the Illegal Messages Received counter and the Illegal Slots Received counter. Both of these counters indicate that a LAT node on the network is sending out incorrectly formatted messages. This indicates that a faulty node is probably present on the network. These counters are usually kept on a per-node basis to aid you in finding the problem node. Software on this node is either faulty or has been corrupted and should be reloaded. It is also possible that the node's hardware is faulty. If the problem persists, contact your local Digital software specialist for information about services Digital offers to assist in diagnosing and resolving the problem.

The data link counters include several counters that can be used for error detection:

- **Send Failures/Send Failures Error Mask** – Send failures usually indicate a problem on the Ethernet cable. Either an open or short circuit was detected, or a problem with the Ethernet physical-level protocol was detected. The error mask gives you more information about the cause of the send failure (see your individual product documentation for more details).
- **Receive Failures/Receive Failures Error Mask** – Receive failures usually indicate a problem with the Ethernet interface or, possibly, a problem on the Ethernet cable. Either a block check error occurred or a framing error was detected. The error mask gives you more information about the type of receive failure (see your individual product documentation for more details).

For both send and receive failure counters, a low value does not indicate a bad condition. For instance, a few shorted cable indications are expected if a new tap is being inserted into the Ethernet cable. When these counter values become high (greater than 20 per hour), you should suspect problems with the Ethernet interface and/or the Ethernet cable.

The data link also provides several counters that give you an idea of the traffic load on the Ethernet:

- **Total Frames Sent – Initially Deferred** is the number of transmit packets that had to be deferred because the Ethernet was busy. During normal Ethernet loads this counter can be expected to be fairly high because of the nature of the Ethernet access algorithm.
- **Total Frames Sent – Single Collision** is the number of transmit packets that were initially deferred and then deferred a second time because of a collision on the Ethernet.
- **Total Frames Sent – Multiple Collisions** is the number of transmit packets that, before being successfully transmitted, were initially deferred and then deferred again two or more times because of additional collisions.

If either of the last two counters is very high, the Ethernet may be too busy. This affects the timing of the server's protocol and may cause active virtual circuits to go down. It may also lead to corrupted service announcement multicast messages, which may cause services to seem unavailable for no apparent reason. If you consistently observe high values for these counters, you may have to decrease your Ethernet traffic. This can be done by using fewer terminal servers, limiting the number of virtual circuits, increasing the circuit timer value, or limiting the usage of PC file transfers (see Chapter 3 for performance tuning information).

4.2 Network Testing Procedures

After verifying that there is an abnormal network condition, there are several testing procedures that you can follow to help you find the source of the problem. These procedures all make use of the loopback facility of the Ethernet. All Digital Ethernet node products have the ability to loop back data that is sent to them when it is sent to their own physical address. Loopback simply means that the receiving node will transmit back to the sender the exact data that it received. The sender can then compare the returned data to verify that it is identical to the data it sent out.

Some nodes have the additional capability to initiate the loopback test either through the Network Control Program (NCP) LOOP CIRCUIT command on DECnet nodes or through the LOOP CIRCUIT command on some terminal server nodes. Section 4.2.2 describes the use of these commands to test the network.

In this section the following terminology is used to describe the nodes involved in loopback testing:

- **Command node** – the node where the LOOP CIRCUIT command is actually typed at a terminal.
- **Executor node** – the node where the loopback data is initially transmitted from and finally received for data comparison. For DECnet nodes, the command node does not have to be the executor node (see a description of the NCP TELL command in the DECnet documentation). For terminal server nodes, the command node and the executor node are always the same node.
- **Target node** – the node where the loopback data is received and transmitted back to the executor node.
- **Assistant node** – the node that provides loopback assistance (see Section 4.2.1).

4.2.1 Loopback Assistants

Loopback assistant is the term used to describe the node that you can designate to help with the loopback test. This node acts as a relay point between your node and the target node. The assistant node can give full assistance, by both transmitting and receiving the test data, or partial assistance, by either only transmitting or only receiving the test data. For loopback assistant tests to be meaningful, you should use an assistant node that is physically located between the executor node and the target node. The three types of loopback assistance are shown in Figures 4–1 to 4–3.

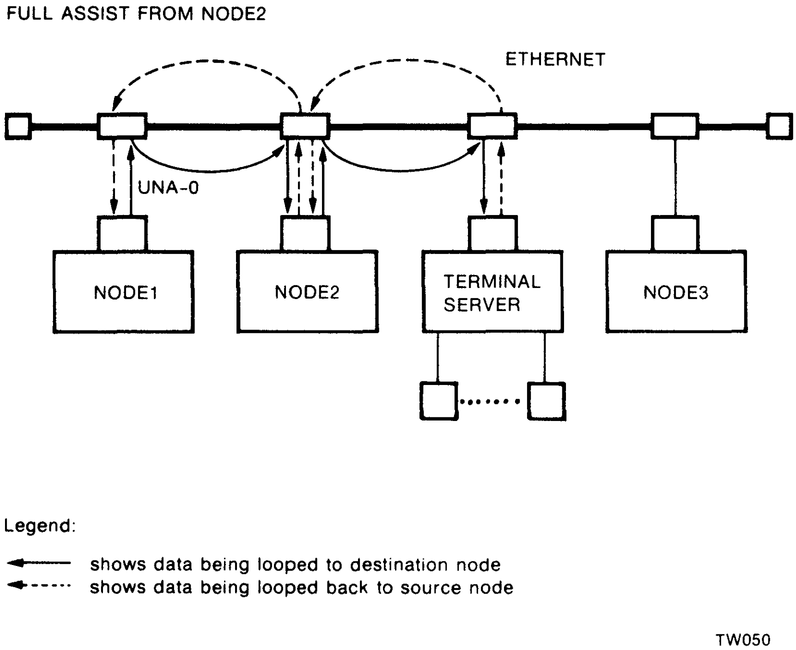


Figure 4–1: Loopback Test with an Assistant Node Giving Full Assistance

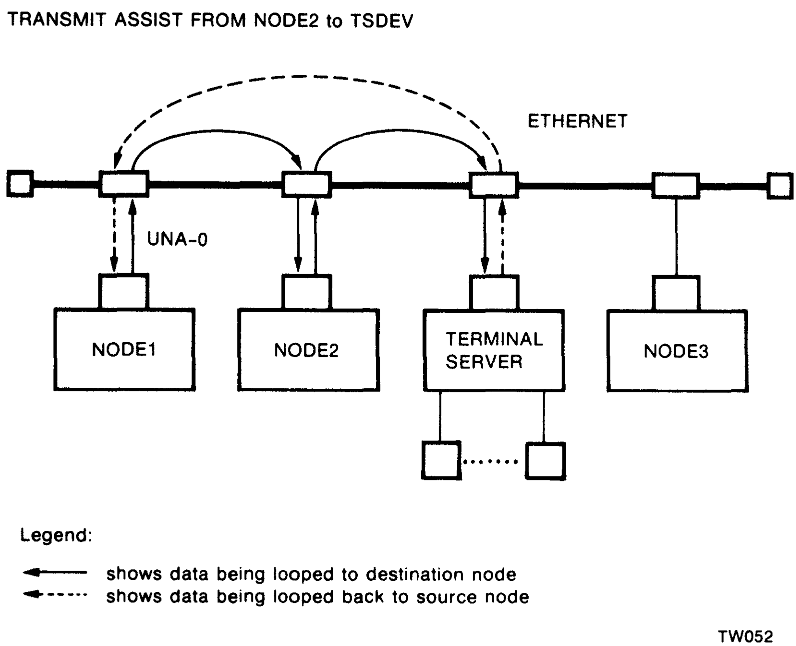


Figure 4–2: Loopback Test with an Assistant Node Giving Transmit Assistance

RECEIVE ASSIST FROM NODE2 to NODE1

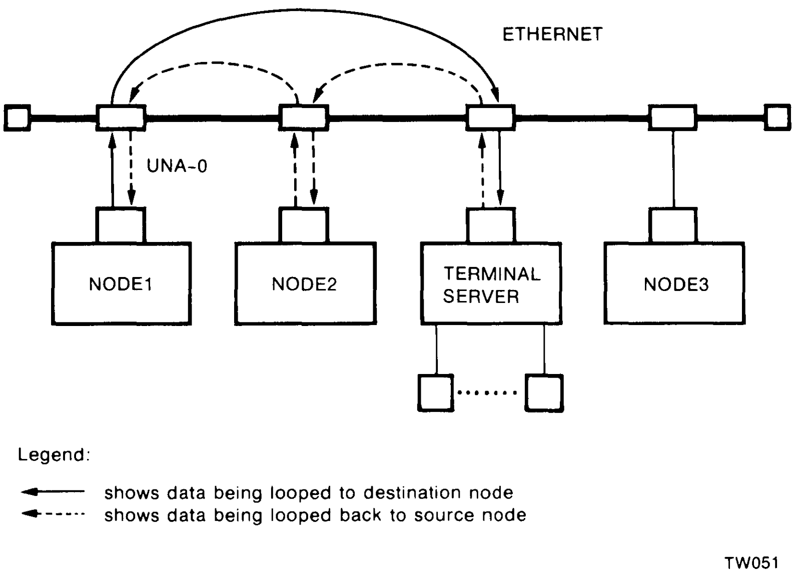


Figure 4–3: Loopback Test with an Assistant Node Giving Receive Assistance

4.2.2 The LOOP CIRCUIT Command

When you use the LOOP CIRCUIT command, you are looping data from the executor node to the target node and back to the executor node. The LOOP CIRCUIT command either completes with a failure message (see Sections 4.2.2.2 and 4.2.2.3) indicating that the test did not complete successfully, or reprompts for another command indicating that the loopback test completed without any failures. In the following syntax description, some of the options are designated "(NCP only)," which means that the syntax applies only to the DECnet NCP LOOP CIRCUIT command.

4.2.2.1 LOOP CIRCUIT Command Syntax – The LOOP CIRCUIT command has the following format:

LOOP CIRCUIT

Circuit identification (NCP only – mandatory):	*circuit-id*
Loopback node specification (mandatory):	{ PHYSICAL ADDRESS address NODE *node-name* }
Loopback data size/count specification (optional):	COUNT *count* LENGTH *length*
Loopback data type specification (NCP only – optional):	WITH { MIXED ONES ZEROS }
Loopback assistance specification (optional):	HELP { FULL RECEIVE TRANSMIT } { ASSISTANT PHYSICAL ADDRESS *address* ASSISTANT NODE *node-name* }

where

circuit-id — (NCP only) specifies the circuit to test. The circuit name is in the form *dev-ctl*, where *dev* is the 2- or 3-letter device name and *ctl* is the controller number. The server has only one Ethernet circuit; therefore, no circuit name is required.

PHYSICAL ADDRESS *address*	specifies the physical address of the target node you wish to test. The physical address of a terminal server can be found on the terminal server hardware (see the server's operation guide). The physical address of a service node can be found by using the DECnet NCP SHOW EXECUTOR CHARACTERISTICS command (see the DECnet NCP documentation). When you are using NCP to perform the test, you can also specify the target node by using the NODE parameter described below.
NODE *node-id*	(NCP only) specifies the DECnet node ID of the target node you wish to test. You can use this method of specifying the target node only if you have set up the node name in the node database. You can also specify the server by using the PHYSICAL ADDRESS parameter described above.
COUNT *count*	specifies the number of loopback frames that are to be sent. The count must be in the range of 1 to 65535. The default is to loop 1 frame.
LENGTH *length*	specifies the length of the loopback messages. The length must be a decimal integer value in the range of 32 to 1484. The default is 40 bytes.
WITH *data-type*	(NCP only) specifies the type of data to be used for the test. Data is all ONES, all ZEROS, or MIXED ones and zeros. The default is MIXED.
HELP *help-type*	specifies the type of help that is desired when performing a loopback test using an assistant node (see Section 4.2.1). The assistant node can give FULL assistance by both transmitting and receiving the test frame, RECEIVE assistance by only receiving the test frame, or TRANSMIT assistance by only transmitting the test frame. The default is to have the assistant node (if any) give FULL assistance.

ASSISTANT PHYSICAL ADDRESS *address*	specifies the Ethernet physical address of the node that is to assist in the loopback test. When you are using NCP to perform the test, the assistant node can also be specified using the ASSISTANT NODE parameter described below. If the HELP parameter is not specified and an assistant node is, then HELP FULL is assumed.
ASSISTANT NODE *node-id*	(NCP only) specifies the DECnet node ID of the node that is to assist in the loopback test. The assistant node can also be specified using the ASSISTANT PHYSICAL ADDRESS parameter described above. If the HELP parameter is not specified and an assistant node is, then HELP FULL is assumed.

For example (using VAX/VMS NCP):

```
NCP>LOOP CIRCUIT UNA-0 PHYSICAL ADDRESS AA-00-04-00-F9-04 - (RET)
_ASSISTANT PHYSICAL ADDRESS AA-00-04-00-82-04 (RET)
```

causes the loopback test to be performed with node AA-00-04-00-F9-04 using the assistant node AA-00-04-00-82-04. In this command, the HELP, COUNT, LENGTH, and WITH parameters have not been specified. This means that, by default, the assistant node will provide FULL assistance. The test consists of one block of 40 bytes of MIXED data. Note the use of the dash (-) at the end of the first line is the continuation character which causes NCP to interpret the two lines as a single command. Terminal servers do not provide a line continuation feature.

4.2.2.2 Using the NCP LOOP CIRCUIT Command – When a loopback test completes, NCP will either provide a failure indication or, if the test is successful, simply reprompt for the next command. The failure messages NCP gives for the LOOP CIRCUIT command are all of the same format:

```
NCP - Loop failed, cause
      Unlooped count = n
```

where *cause* is a message identifying the cause of the loopback test failure and *n* is the count of messages not looped.

4.2.2.3 Using the Terminal Server LOOP CIRCUIT Command – The terminal server LOOP command either completes with a failure message indicating that the test did not complete successfully, or completes with a message indicating that the loopback test completed without any failures.

NOTE

The terminal server LOOP CIRCUIT command is not supported by all terminal servers; see your individual product documentation to determine whether your server supports the LOOP CIRCUIT command. In addition, the syntax for the LOOP CIRCUIT command may be slightly different than the syntax documented (some keywords may be optional).

4.3 The DECnet Remote Console Facility

Network nodes that implement Phase IV DECnet have a remote console facility (RCF) that you can use to set up a logical connection between your terminal on a DECnet host node and the terminal server. This allows your terminal to connect directly to the terminal server. Using RCF, you can enter local mode and execute terminal server commands just as you would if your terminal were directly attached to the terminal server hardware. This feature provides diagnostic benefits without having to dedicate a terminal to this function. Another benefit of this facility is the ability to perform remote diagnostics from a single, centralized terminal.

On some servers, you can also use RCF to cause an up-line dump of the server image to occur. This allows you to force a crash dump of a server that is experiencing problems. RCF is also used by Digital software specialists for diagnostic purposes in special situations.

The use of the remote console facility is server-dependent and you are referred to the documentation for your particular server for further information.

4.4 Using DECnet Event Logging

Most of the LAT terminal servers use the DNA Maintenance Operations Protocol (MOP) to down-line load themselves during server initialization. In order for the down-line load request to be successful, there must be at least one running DECnet node on the Ethernet at the time of the request. Depending on how many alternate loading hosts are set up and running, the server can issue a load request to any number of loading hosts. This can lead to confusion about the status of the server's load request and identity of the actual load host.

In order to avoid confusion about the load status, you should use NCP to enable event logging at all DECnet loading host nodes. Event logging is a service provided by DECnet nodes that generates an event message whenever a network event occurs. With event logging enabled, you receive an event message whenever a load sequence completes. This message gives a positive indication that the server's software has been correctly loaded.

When event logging is set up on a DECnet node, the destination, or sink, of the event messages must be specified. It is suggested that you set up one DECnet sink node to receive all the logging events associated with down-line loading. In this way, all load request status information is available at one node.

For more information on event logging and the ability to set up sink nodes, see the loading host DECnet documentation. For more information about the loading sequence for a particular server, see the server's operation or management guide.

Glossary

circuit timer

The LAT protocol timer used to determine at what intervals a terminal server should transmit Virtual Circuit layer messages. Each time this timer expires, and data is available to send, the terminal server transmits one message to each service node that currently has a virtual circuit established.

down-line load

The process by which a DECnet load host node on the Ethernet transfers system software to a terminal server node and causes it to be executed.

Ethernet

A local area network that employs coaxial cable as a passive communications medium to interconnect different types of computers, server products, and office equipment at a local business site. No switching logic or central computer is needed.

flow control

The protocol mechanism used by the LAT protocol that ensures that a sending network node does not overrun a receiving node with more data than it can accept.

group code

A LAT protocol feature that allows network nodes to belong to collections of nodes called groups. Network nodes become a member of a group by enabling the group code for that group. Only network nodes that have the same group code enabled can communicate with one another.

host (host node)

A DECnet node that is used either to down-line load the system software into a terminal server or to up-line dump the memory image from a terminal server. This capability is necessary for terminal servers that do not have any form of local secondary storage (disk or tape units). The primary load host node is the node that actually loads the terminal server. Multiple backup host nodes can be specified that will be used to dump the server's memory in the event the primary host is unavailable.

local area network (LAN)

A privately owned data communications system that offers high-speed communications channel(s) optimized for connecting together information processing equipment.

message

The unit of information transmitted by the LAT Virtual Circuit layer. A message in the LAT protocol is always completely contained in one data link frame.

multicast

A feature of the Ethernet protocol that provides special addresses that nodes can transmit to or receive from that are not node specific. These multicast addresses allow a collection of nodes to be known by one physical address. The action of multicasting a message occurs when a message is sent to one of these multicast addresses.

network

A configuration of two or more computers linked to share information and resources. The computers may be either general purpose computers or dedicated computers such as LAT terminal servers.

node

A network management component consisting of a computer system and the associated network software.

octet

A group of 8 bits. In the Ethernet protocol, all fields are measured in octets.

protocol

A basic procedure or set of rules that governs and controls the flow of data between information processing equipment. Also, a set of conventions between communicating processes regarding the format and contents of messages to be exchanged. The LAT products use one protocol as the framework for communication. Most LAT servers also use the DNA Maintenance Operations Protocol (MOP) to down-line load and up-line dump their system software.

rating

A LAT protocol value given to service names to indicate the relative level of availability of the service on a specific service node. The servers use this rating to choose the preferred service node in cases where multiple service nodes offer the same service.

response time

The time between a user entering a character at a terminal keyboard and the time the first response to the character is seen by a user. Response time can be either echo response time, in which case the terminal driver is the process generating the response, or application response time, in which case an application program generates the response.

server (server node)

A network node usually containing specialized software that performs a dedicated function. LAT terminal servers provide the facilities of a network terminal switch. Servers can also provide the functions of a service node for computer systems that lack the necessary software or hardware.

service

A logical function offered by a service node supporting the LAT protocol. This service node can be a computer system with integrated LAT protocol software, or a server that provides the LAT protocol software and Ethernet interface for a computer system that lacks the neccesary software/hardware components. This function may be a specific hardware/operating system combination (such as system ABC running VAX/VMS), a generic hardware/software combination (such as RSX-11M-PLUS), or a software package providing a function (such as VMSMAIL).

service class

A feature of the LAT protocol which allows for extension of the basic protocol. Service classes are additional protocols that enhance the services of the basic LAT protocol. Interactive terminals use the Service Class 1 extension.

service name

A LAT protocol field in the service announcement multicast message used by service nodes to indicate the services that they provide. A service name can be used by more than one service node. In that case, a terminal server selects the service node with the highest rating for that service name. The service name is specified by terminal server users in the CONNECT command when establishing a service connection.

service node

A node, implementing the LAT protocol, that provides one or more services to the terminal servers in the LAT network. This node can be a server.

slot

A variable length field in the Virtual Circuit layer RUN message. The field is used to carry information about one connection (also called a session) between a terminal server user and a service node. This field contains header information (which includes the slot identification, the type of slot, and the slot length) and user data.

virtual circuit

A logical communication path which includes the server, the intervening local area network hardware, and the service node. In LAT networks, there is at most one virtual circuit between a terminal server and a service node. The Virtual Circuit layer establishes the virtual circuit when it receives a request from the Slot layer for a service node for which a virtual circuit is not currently established. The virtual circuit is terminated when no more Slot layer traffic is present.

Index

Local Area Transport (LAT) Architecture
Network Manager's Guide
AA-DJ18B-TK

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor				Excellent
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

____

General comments:

____

Suggestions for improvement:

____

Name ____________________ Date ____________
Title ____________________ Department ____________
Company ____________________ Street ____________
City ____________ State/Country ____________ Zip Code ____________

DO NOT CUT FOLD HERE AND TAPE

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY LABEL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

digital

SOFTWARE DOCUMENTATION
1925 ANDOVER STREET TWO/E07
TEWKSBURY, MASSACHUSETTS 01876

DO NOT CUT FOLD HERE

CUT ALONG DOTTED LINE

digital

AA-DJ18B-TK

Printed in U.S.A.