Local Area Transport (LAT) Network Concepts

Order No. AA–LD84A–TK

June 1988

This manual describes the Local Area Transport (LAT) architecture, the LAT protocol, and LAT network concepts. This manual is intended for the server manager, the system manager, and the network manager.

Supersession/Update Information:	Revised for new covers.
Software Version:	DECserver 500 V1.0

digital

AA-LD84A-TK
First Edition, June 1988

The postage-prepaid Reader's Comments form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

This manual was produced by Networks and Communications Publications.

Contents

Preface

1 Introduction to LAT Networks

1.1 Services 1-3

1.2 Sessions 1-4

1.3 LAT Software 1-7

1.4 LAT Nodes 1-8

1.4.1 Overview of Terminal Servers 1-10

1.4.1.1 Devices Supported by All Terminal Servers 1-13

1.4.2 Overview of Service Nodes 1-13

1.4.2.1 General-Purpose Service Nodes 1-14

1.4.2.2 Dual-Purpose Servers 1-16

1.5 Benefits of LAT Networks 1-19

1.6 LAT Network Configurations 1-21

1.6.1 Logical Partitioning of LAT Networks Using Groups 1-23

1.6.2 Access to Other Networks from a LAT Network 1-24

1.7 LAT Management Tasks 1-27

1.8 Summary 1-28

2 LAT Nodes

2.1 Terminal Servers . . . 2-1
2.1.1 Modifiable Characteristics . . . 2-2
2.1.1.1 Server Characteristics . . . 2-3
2.1.1.2 Port Characteristics . . . 2-3
2.1.1.3 Local-Service Characteristics . . . 2-4
2.1.1.4 Server Databases for Characteristics . . . 2-4
2.1.2 Software Port Configurations . . . 2-7
2.1.3 Security Features . . . 2-8
2.1.3.1 Passwords . . . 2-8
2.1.3.2 Three Levels of Security for Interactive Ports . . . 2-9
2.1.4 The Server Command Interface . . . 2-9
2.1.4.1 Privileged Commands . . . 2-10
2.1.4.2 User Commands . . . 2-10
2.1.4.3 On-Line Help . . . 2-11
2.1.5 Personal Computers on LAT Networks . . . 2-11
2.1.5.1 Personal Computers Attached to Terminal Servers . . . 2-12
2.1.5.2 Server Software on Personal Computers . . . 2-12
2.2 Service Nodes . . . 2-14
2.2.1 General-Purpose Service Nodes . . . 2-14
2.2.1.1 The LAT Control Program (LCP) . . . 2-15
2.2.2 Dual-Purpose Servers . . . 2-15
2.2.2.1 Printers as LAT Services . . . 2-16
2.2.2.2 Personal Computers as LAT Services . . . 2-16
2.2.2.3 Modems as LAT Services . . . 2-16
2.2.2.4 Non-LAT Hosts as LAT Services . . . 2-17

3 LAT Architecture

3.1 Fundamental Features of LAT Architecture 3-2
3.2 Overview of the Architectural Layers of LAT Networks 3-3
3.3 Service Class Layer 3-8
3.3.1 Service Class 1 Functions 3-9
3.3.2 Service Class 1 Messages 3-9
3.3.2.1 Service-Announcement Messages 3-10
3.3.2.2 Solicit-Information and Response-Information Messages 3-10
3.3.2.3 Command and Status Messages 3-11
3.3.3 Service-Class Flow Control 3-12
3.4 Slot Layer 3-13
3.4.1 Slot Layer Functions 3-13
3.4.2 Types of Message Slots 3-13
3.4.2.1 Start Slot 3-13
3.4.2.2 Reject Slot 3-13
3.4.2.3 Data Slots 3-14
3.4.2.4 Stop slot 3-14
3.4.3 Slot Flow Control 3-14
3.5 Virtual Circuit Layer 3-17
3.5.1 Virtual Circuit Layer Functions 3-17
3.5.2 Virtual Circuit Layer Messages 3-18
3.5.2.1 Start Messages 3-18
3.5.2.2 Run Messages 3-18
3.5.2.3 Stop Messages 3-19
3.5.3 Virtual-Circuit Flow Control 3-19
3.6 A VAX/VMS Implementation of LAT Layers 3-20

4 LAT Communications Processes

4.1 The Service-Announcement Process 4-2
4.1.1 Maintaining Service Ratings 4-2
4.1.2 Multicasting Service-Announcement Messages 4-3
4.1.3 Maintaining Service and Service-Node Directories 4-3
4.2 The Session-Establishment Process 4-4
4.2.1 Selecting a Suitable Service Node 4-4
4.2.2 Ensuring that a Virtual Circuit Exists 4-7
4.2.2.1 Exchanging Session Information over a Virtual Circuit . 4-7
4.2.3 Starting Sessions 4-9
4.2.3.1 Sending a Start Slot in a Run Message 4-9
4.2.3.2 Responding to Start Slots 4-10
4.3 Session Management 4-12
4.3.1 Passing User Data for Sessions 4-12
4.3.2 Performing Automatic Failover 4-13
4.3.3 Disconnecting Sessions 4-13
4.4 Virtual Circuit Maintenance 4-14
4.4.1 Retransmitting Messages 4-14
4.4.2 Managing Balanced Virtual Circuits 4-14
4.5 The Host-Initiated-Request Process 4-17
4.5.1 Translating a Server Name 4-17
4.5.2 Making Host-Initiated Requests 4-17
4.5.3 Processing Host-initiated Requests 4-18
4.6 Connection-Queue Maintenance 4-20

5 Managing a LAT Network

5.1 Planning the Physical Configuration of a LAT Network 5-2
5.2 Preparing for Server Hardware Installation 5-2
5.3 Installing Server Distribution Software 5-4
5.4 Customizing the Load Host's Node Database 5-5
5.5 Using the Remote Console Facility (RCF) 5-6
5.6 Customizing Serverwide Characteristics 5-8
5.7 Configuring Server Ports 5-8

5.7.1 Configuring Ports for Interactive Devices 5-9
5.7.1.1 Assigning Port Groups 5-10
5.7.1.2 Educating Users 5-10
5.7.2 Configuring Ports for Applications Devices 5-11
5.8 Setting Up Service Nodes for Offering Services 5-11
5.8.1 Setting Up Service-Node Characteristics 5-12
5.8.1.1 Service-Node Names 5-12
5.8.1.2 Node Identification String 5-12
5.8.1.3 Service-Node Groups 5-13
5.9 Creating Services 5-20
5.9.1 Assigning Service Names 5-20
5.9.2 Special Considerations for Dual-Purpose Servers 5-20
5.10 Managing Services 5-21
5.10.1 Managing Services on a General-Purpose Service Node . . 5-21
5.10.2 Managing Services on Dual-Purpose Servers 5-21
5.10.2.1 Using Service Passwords 5-22
5.11 Setting Up VMS Service Nodes for Host-Initiated Requests . 5-22
5.12 Setting Up Server Ports for Host-Initiated Requests 5-24
5.12.1 Deciding Whether To Assign a Service Name 5-24
5.13 Managing the Connection Queue 5-25
5.14 Managing Communications on a LAT Network 5-25
5.14.1 Selecting the Value of the Multicast Timer 5-26
5.14.2 Selecting the Value of the Circuit Timer 5-26
5.14.3 Selecting the Value of the Retransmit Limit 5-26
5.14.4 Selecting the Value of the Keepalive Timer 5-27
5.15 Using the Terminal Server Manager (TSM) Product 5-27
5.15.1 Management Directory 5-28
5.15.2 Database Manipulation 5-29
5.15.3 Server Selection 5-29
5.15.4 Initial Configuration 5-29
5.15.5 Wildcard Operations 5-29
5.15.6 Fault Management 5-29
5.15.7 Command File Support 5-30

Index

Figures

1-1 Arrangement of LAT Nodes on an Ethernet Cable 1-2
1-2 Session between Terminal Server and Service Node 1-5
1-3 Multiple Sessions 1-6
1-4 Sample Three-Node LAT Network 1-9
1-5 Applications Program Accessed as a Service 1-15
1-6 Devices Offered as Services by Dual-Purpose Servers 1-18
1-7 Sample LAT Network with Several Types of LAT Nodes . . . 1-22
1-8 Example of an Ethernet LAN with a DECnet/SNA Gateway . 1-25
1-9 LAT Server User Accessing an IBM file 1-26
3-1 LAT Layers and the Underlying DNA Layers 3-5
3-2 Comparison of DECnet and LAT Architectures 3-6
3-3 Interlayer Interfaces of LAT Architecture 3-7
3-4 LAT Protocol Implementations 3-8
3-5 Slot Flow Control on a LAT Network 3-15
3-6 Operation of Slot Flow Control 3-16
3-7 A VMS Implementation of LAT Network Layers 3-21
4-1 VAXcluster with a Common Service Name 4-8
4-2 Terminal Server Establishing a Session 4-10
4-3 Session-Establishment Functions of Servers 4-11
4-4 Operation of the Virtual Circuit Layer 4-16
4-5 The Path of a Host-initiated Request 4-18
4-6 Service-Node Functions in Host-Initiated-Request Process . . 4-21
4-7 Processing Queued Connection Requests for Same Port . . . 4-22
4-8 Processing the Queue After Port 8 Becomes Available 4-24
5-1 Relationship of Remote Console to a Terminal Server 5-7
5-2 Group Assignments on a Sample LAT Network 5-15
5-3 Logical Subnetwork Formed by Group 1 5-16
5-4 Logical Subnetwork Formed by Group 6 5-17
5-5 Logical Subnetwork Formed by Group 8 5-18
5-6 Logical Subnetwork Formed by Group 9 5-19
5-7 TSM Program and Terminal Servers on a LAT Network . . . 5-28

Tables

1-1 LAT Terminal Servers 1-10
1-2 Modem Control and Service Software on Terminal Servers . 1-12
1-3 Devices Supported on All Terminal Servers 1-13
1-4 General-Purpose Service Nodes 1-14
1-5 Dual-purpose Servers 1-17
1-6 Devices Supported by Dual-Purpose Servers 1-19
1-7 Job Titles and Job Responsibilities in LAT Documentation . 1-27
3-1 Components of LATplus/VMS 3-20
5-1 Operating Systems with Load-Host Capability 5-5
5-2 Groups on a Sample LAT Network 5-14

Preface

Local Area Transport (LAT) Network Concepts describes the implementation of Digital's Local Area Transport **(LAT)**) architecture. LAT architecture defines a model for communications on Ethernet local area networks. This guide discusses LAT communications functions, LAT nodes, setting up and managing LAT networks, and the LAT architecture.

Intended Audience

This guide is intended for anyone interested in gaining a general introduction to LAT networks, such as the server manager, LAT network manager, and system managers of LAT service nodes. It is particularly useful to network managers and planners that require an understanding of basic LAT components, features, and operations.

Structure of This Document

Local Area Transport (LAT) Network Concepts has five chapters:

Chapter 1	Introduces the LAT network components
Chapter 2	Provides general information about Digital's family of special-purpose computers known as terminal servers and Digital's multiuser computers that function as LAT service nodes
Chapter 3	Describes the LAT architecture
Chapter 4	Summarizes LAT communications processes
Chapter 5	Summarizes the managements tasks involved in setting up and managing a LAT network

Associated Documents

The following terminal server documents provide specific information about many LAT products. For information on how to order terminal server documentation, see the last few pages of this guide.

DECserver 100 Documents

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

  Describes pursuit requirements and procedures for installing the DECserver 100 hardware.

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

  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*

  Contains the complete operation procedures for the DECserver 100 product.

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

  Contains the operations information necessary for the DECserver 100 terminal user.

- *DECserver 100 Terminal Server Identification Card*

  Used to record the server's Ethernet address, DECnet node name, and serial number.

DECserver 200 Documents

- *DECserver 200 Problem Determination Guide*

  Describes DECserver 200 problem determination and troubleshooting. It is intended for the server manager.

- *Using DECserver 200 Documents*

  Directs installers, server managers, and server users to information contained in the DECserver 200 documentation set. Flowcharts for different audiences suggest logical reading sequences.

- *DECserver 200 Hardware Installation/Owner's Guide*

  Explains how to install the DECserver 200 hardware unit and how to test its operation. The guide also describes the DECserver 200 controls and indicators.

- *DECserver 200 Identification Card*

  Provides the space to record the serial number, Ethernet address, DECnet node address, and DECnet node name of the server. This document is intended for the network manager, the software installer, and the server manager.

- *DECserver 200 Software Installation Guide (op-sys)*

  Explains how to install the DECserver 200 distribution software, how to establish down-line load hosts, and how to verify the DECserver 200 system installation. In the title, *op-sys* is the name of the load host operating system.

- *DECserver 200 User's Guide*

  Describes the user command interface and the general functions of the server. This guide provides complete information for using all nonprivileged server commands.

- *Terminal Server User's Reference Card*

  Describes and gives examples of the most frequently used nonprivileged server commands for server users.

- *DECserver 200 Management Guide*

  Describes all the initial and day-to-day management tasks required of the DECserver 200 manager. The topics cover all the information needed to configure the ports and to customize the permanent and operational databases of the server.

- *Terminal Server Commands and Messages Reference*

  Describes the usage and syntax of all terminal server commands. This reference also lists and describes all status and error messages issued by the server. This reference is intended for the server manager but is useful for terminal users who want more detailed reference information.

- *DECserver 200 Commands Mini-Reference*

  Summarizes all privileged and nonprivileged server commands and characteristics in a mini-reference.

- *DECserver 200 Technical Manual*

  Describes hardware logic, diagnostic firmware and software, and general operating procedures. The level of technical information assumes previous training or experience with Ethernet networks and with Digital VAX-11 or PDP-11 architecture.

- *Terminal Server Glossary*

  Defines terms used in the server documentation sets. This document is intended as a reference for all users of server documentation.

DECserver 200 On-Line Documentation

- *DECserver 200 Release Notes*

  Describes any discrepancies between the actual product and the information in the documentation set. These notes are intended for the software installer and the DECserver 200 manager.

- On-Line Help

  Provides two forms of server help: tutorial help and command reference help. Tutorial help provides basic information about logging in and using the server. Command reference help provides detailed information about using the server commands available at each privilege level.

DECserver 500 Documents

- *Using DECserver 500 Documents*

  Directs you to information contained in the DECserver 500 documentation set. Flowcharts for different audiences suggest logical reading sequences.

- *Terminal Server Commands and Messages Reference*

  Describes the usage and syntax of all terminal server commands and Terminal Server Configurator (TSC) commands. This reference also lists and describes all status and error messages issued by the server and by the TSC.

- *DECserver 500 Software Installation Guide (op-sys)*

  Explains how to install the DECserver 500 distribution software, how to establish down-line load hosts, and how to verify the DECserver 500 system installation. In the title, *op-sys* is the name of the load host operating system.

- *DECserver 500 User' s Guide*

  Describes the user interface and the general functions of the server. This guide provides complete information for using all nonprivileged server commands.

- *Terminal Server User' s Reference Card*

  Describes and gives examples of the most frequently used nonprivileged server commands on a reference card.

- *DECserver 500 Identification Card*

  Provides the space to record the serial number, Ethernet address, DECnet node address, and DECnet node name of the server.

- *DECserver 500 Hardware Installation Guide*

  Explains how to install the DECserver 500 hardware unit and how to test its operation. The guide also describes the DECserver 500 controls and indicators.

- *DECserver 500 Management Guide*

  Describes all the initial and day-to-day management tasks required of the DECserver 500 manager. The topics cover all the information needed to configure the ports and to customize the permanent and operational databases of the server.

- *DECserver 500 Commands Mini-Reference*

  Summarizes all privileged and nonprivileged server and TSC commands and characteristics in a mini-reference.

- *DECserver 500 System Owner's Guide*

  Describes the hardware and software components of the DECserver 500 system. Procedures are also provided for powering up and shutting down the server, for solving simple server problems, and for expanding or reconfiguring the server hardware.

- *DECserver 500 Problem Determination and Service Guide*

  Describes the server's troubleshooting tools and procedures. This guide also explains how to isolate server faults and how to repair the server to the field replaceable unit level.

- *DECserver 500 Troubleshooting Quick Reference Card*

  Lists the server status and error codes, which are displayed in the LED display located on the handle of the CPU module. This card also lists the server boot modes, device and vector addresses for modules, and provides a configuration chart for recording the server hardware configuration and cable destinations.

- *DECserver 500 Technical Manual*

  Describes how the DECserver 500 system software and hardware components interact to perform server functions. This manual also describes the hardware specifications, the controls and indicators, and the diagnostics self-test program. Detailed descriptions of the hardware components are not included but are found in the hardware manual for the specific component.

- *Terminal Server Glossary*

  Describes terms used in the server documentation sets.

DECserver 500 On-Line Documentation

- DECserver 500 Release Notes

  Describes any discrepancies between the actual product and the information in the documentation set.

- DECserver 500 On-Line Help

  Provides two forms of server help: tutorial help and command reference help. Tutorial help provides basic information about logging in and using the server. Command reference help provides detailed information about using all the server commands available at your privilege level.

- Terminal Server Configurator (TSC) On-Line Help

  Provides reference information for all commands used to customize the server image in the permanent database on the load host.

Ethernet Communications Server Terminal Server Documents

NOTE

> Throughout this guide the Ethernet Communications Server Terminal Server is called the Ethernet Terminal Server.

- *Ethernet Communications Server Site Preparation and Planning Guide*

  Discusses the issues involved in choosing a site for the Ethernet Communications Server hardware.

- *Ethernet Communications Server Installation Guide*

  Describes the procedures for installing and setting up the Ethernet Communications Server hardware.

- *Ethernet Communications Server Operations and Maintenance Guide*

  Describes maintenance procedures for the Ethernet Communications Server.

- *Ethernet Communications Server Terminal Server Operations Guide*

  Describes the operation of the Ethernet Terminal Server software, which is down-line loaded into the Ethernet Communications Server.

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

  Describes 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*

  Briefly describes 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*

  Used to record the server's Ethernet address, DECnet node name, and serial number.

MUXserver 100 Documents

- *MUXserver 100 Network Reference Manual*

  Describes the installation of the MUXserver network, and discusses both the MUXserver 100 and the DECmux II. This manual explains in detail all supervisor and user commands, the theory of operation, diagnostics, error messages, and system and subsystem debugging and fault isolation procedures.

- *MUXserver 100 Network Installation Manual*

  Describes the installation of the MUXserver network and the procedures to isolate problems, thereby ensuring a successful installation of a fully operational network.

- *MUXserver 100 Users' Pocket Guide*

  Provides concise descriptions of the commands available to users.

- *MUXserver 100 Identification Card*

  Used to record the Ethernet address of a MUXserver 100. Additionally, this card acts as a quick reference guide on the various network configurations available.

- *MUXserver 100 Software Installation Guide (op-sys)*

  Describes how to install the MUXserver 100 down-line loadable image on a Digital load host system.

Terminal Server Manager Documents

- *Guide to Terminal Server Manager*

  Contains the information to operate the Terminal Server Manager (TSM) software installed on a VAX/VMS system running DECnet-VAX. The guide describes software procedures to manage a mix of Digital terminal servers connected to the same Ethernet as a VAX computer. TSM is an optional product.

- *Terminal Server Manager Software Installation Guide*

  Contains information for system managers or network managers responsible for maintaining and for configuring terminal servers on a local area network (LAN).

VAX/VMS Systems

- *LATplus/VMS Service Node Management Guide*

  Discusses the management of a VAX/VMS system that is operating as a service node on a LAT network and running the LATplus/VMS software, which is a VMS layered product. LATplus/VMS software implements Version 5.1 of the LAT architecture. The management guide and the VAX/VMS service node software is included in VAX/VMS Version 4.2 software distribution kits for all servers.

NOTE

> For VMS versions after 4.2, the latest implementation of LAT architecture Version 5.1 will replace the LAT software that is currently distributed with all VMS operating systems and will no longer be distributed as a VMS layered product. At that time, the *LATplus/VMS Service Node Management Guide* will also be removed from the software distribution kits of terminal servers and its contents will be absorbed into the regular VMS document set.

Related Documents

- *The Ethernet—A Local Area Network—Data Link Layer and Physical Layer Specifications*

  Contains the specification of the Ethernet local area network. The specification was developed jointly by Digital Equipment Corporation, Intel Corporation, and Xerox Corporation. The specification is intended as a design reference document rather than as an introduction or a tutorial document.

- *VAX/VMS System Messages and Recovery Procedures Reference Manual*

  Explains all NCP information and error messages.

- *Introduction to Network Performance*

  Introduces the basic concepts and terminology of network performance relating to Ethernet networks. It provides information that helps network managers determine what constitutes good performance, determine how to measure performance, determine factors that affect performance, and how to improve network performance.

Conventions Used in This Document

This guide uses standard terminology for all LAT products.

Graphic Conventions

The following graphic convention is used throughout this manual:

Bold text — Identifies an important term that is defined in the text. The term is also listed in the terminal server glossary.

1

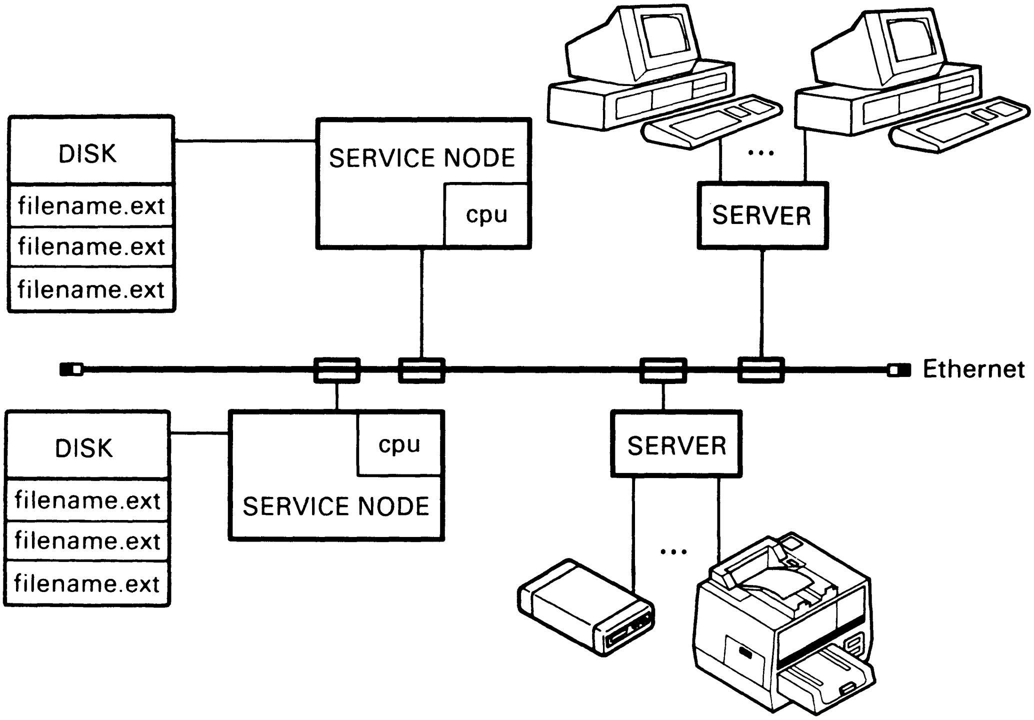
Introduction to LAT Networks

The **Local Area Transport** (LAT) facility operates on an Ethernet local area network (LAN). The systems on an Ethernet LAN that support the LAT facility compose a **LAT network**. LAT networks combine hardware and software components to permit communication between computer systems and other devices, such as terminals, printers, and modems. A LAT network consists of LAT nodes, their network interfaces, and the Ethernet LAN that connects those nodes. By allowing distributed access among many types of computers and applications devices, a LAT network permits network planners and managers to maximize computing resources.

The primary goal of a LAT network is supplying access to computing resources. Resources are offered by some LAT nodes (called service nodes) and are accessed by others (called terminal servers) through logical connections called sessions. A Session provides a two-way communications path between a resource and a server user or applications program. Generally, a resource is defined as a service. To connect to a service, server users request it by a logical service name and need not know the identity of the service node(s) that offer the service.

Figure 1-1 illustrates a simple arrangement of LAT nodes on an Ethernet cable.

Figure 1-1: Arrangement of LAT Nodes on an Ethernet Cable



LKG-1185-87

The LAT architecture and associated LAT protocol distinguish a LAT network from other networks, such as DECnet, that can exist on the same LAN. The **LAT architecture** is a layered network model. It identifies LAT communications functions, assigns distinct functions to distinct layers, and specifies general rules for communications between nodes in network. A general description of the LAT architecture is provided in Chapter 3. The **LAT protocol** is an integral part of the LAT architectural model. It consists of rules that specify the actual format and sequence of the messages used for communication between LAT nodes. Implementations of the LAT protocol are called **LAT**

software. The presence of LAT software is the only feature that defines a LAT node.

This chapter introduces the following aspects of LAT networks:

- Services
- Sessions
- LAT software
- LAT nodes
- LAT network configurations
- Benefits of LAT networks
- LAT management tasks

The remaining chapters provide information about several aspects of LAT networks including terminal servers (Chapter 2), service nodes (Chapter 2), the LAT architecture (Chapter 3), LAT communications (Chapter 4), and an overview of LAT management tasks and tools (Chapter 5).

1.1 Services

A **service** is a named resource that is either specific hardware (such as a dial-out modem), a combination of hardware and software (such as a VAX/VMS or a non-Digital computer), or a logical function (such as an application program). Some services (such as a dial-out modem or a multiuser computer) are useful to users of interactive terminals; other services (such as a printer attached to a terminal server port) are meaningful only to application programs.

Each service has a unique name that identifies it on the LAT network. Service names make resources logically independent of the LAT nodes that offer them: a service can occur on multiple nodes, and a single node can offer multiple services. If two or more service nodes offer the same service name, servers assume that all the services with that name are identical to each other and are interchangeable. Service names are essenially independent of the current physical arrangement of systems on the network. Therefore, the physical network can be reconfigured (for example, by the addition or subtraction of service nodes), while the logical services offered by the network remain unchanged.

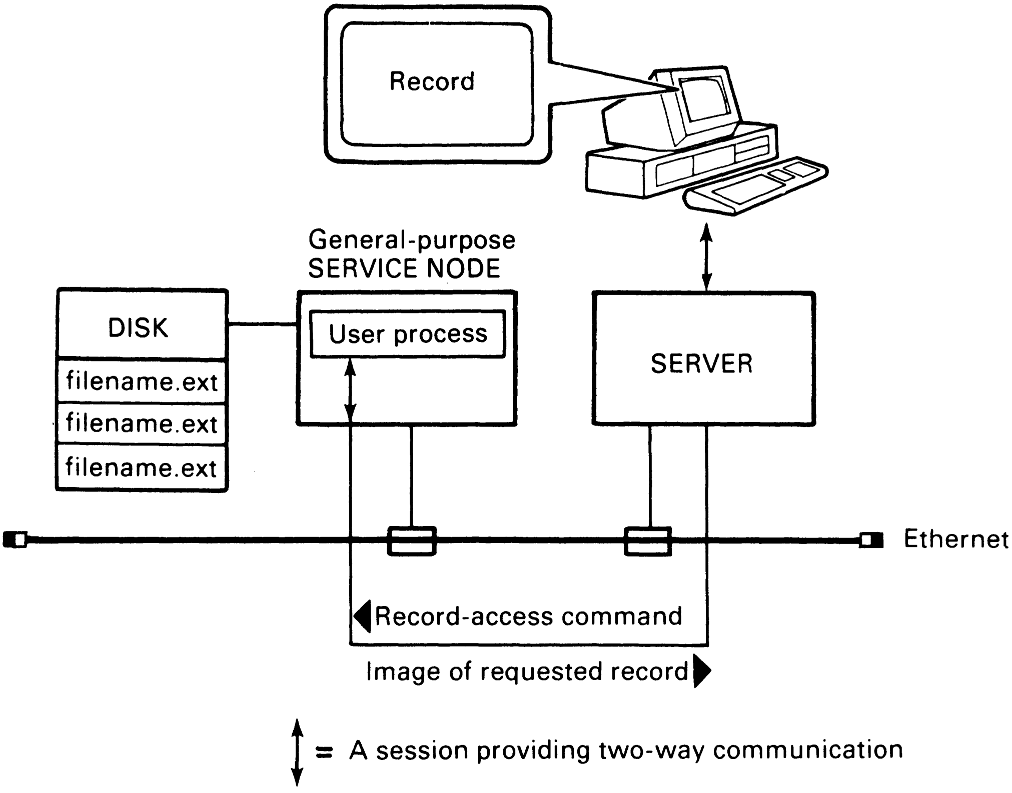
By default, every server user has access to every service on the LAT network. However, server managers can prohibit access to any service node (and its services) from specific server ports. Access is prohibited if the server manager assigns one logical group to all service nodes offering the service and another group to the server ports to be restricted. Groups are discussed in Section 1.6.1.

1.2 Sessions

A **session** is an asynchronous, two-way logical connection between a server port and a service. To a server user, a LAT session appears to be a direct physical connection between a terminal and a computer system.

Figure 1-2 illustrates the two-way flow of information between a terminal on a LAT terminal server and a user's process on a multiuser computer operating as a LAT service node. In this figure, the server user issues a command to access a data record from a particular file that resides on one of the computer's disks. The command passes from the terminal server to the user's process on the service node. The user's process accesses the requested record and passes an image of the record over the session to the terminal screen. To the user, this two-way flow of information in a session appears instantaneous.

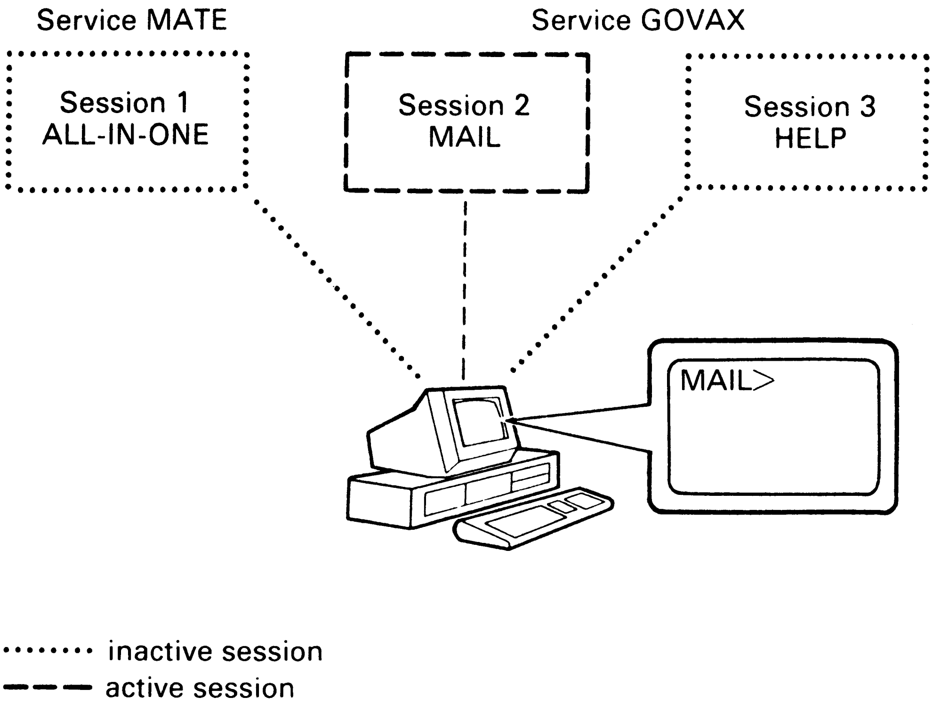
Figure 1-2: Session between Terminal Server and Service Node



LKG-1184-87

A server user can simultaneously maintain multiple sessions and move among them freely and quickly. Note that only the currently selected session is active. Having **multiple sessions** allows users to intermingle tasks such as using an ALL-IN-ONE spreadsheet for accessing a database, reading and sending mail, and using on-line help. Figure 1-3 provides an example of such a three-session arrangement. Session 2, the active session, is currently accessing VAX/VMS MAIL. Sessions 1 and 3 are inactive.

Figure 1-3: Multiple Sessions



LKG-1189-87

A server user can have multiple sessions with the same service or with different services. For example, the sample sessions in Figure 1-3 are with two services. Session 1 is with a service named MATE, and sessions 2 and 3 are with a service named GOVAX, which is a multiuser VAX/VMS system.

1.3 LAT Software

LAT software implements the LAT architecture and protocol on LAT nodes. The most fundamental components of LAT software are service-node software and server software. These two components cooperate to enable communications and data exchange in LAT networks. The primary function of service-node software is to provide services on the LAT network; the primary function of server software is to access those services.

Service-node and server software address complementary aspects of two basic communications functions:

- Generating and maintaining information about services:
  - The service-node software offers named services and announces them over the LAN.
  - The server software keeps a directory of announced services and of the nodes that announced them.
- Establishing and managing sessions between services and server users:
  - The server software responds to a connection request by initiating a session with the requested service.
  - The service-node software recognizes and processes a session-initiation request by accepting or rejecting it.

LAT Protocol Version 5.1 defines an additional communications capability for service-node and server software, which allows the following functions:

- Service-node software can request server software to initiate a session between an applications program on the service node and an applications device on a terminal server. These connection requests are called host-initiated requests to distinguish them from connection requests of interactive users on terminal servers (user-initiated requests).
- Server software can recognize host-initiated requests, process them on a first-come-first-served basis, and initiate a session between the requested applications device and the applications program.

1.4 LAT Nodes

LAT software resides on systems called LAT nodes, which are computer systems that contain service-node software, server software, or both. There are two types of LAT nodes: terminal servers and service nodes.

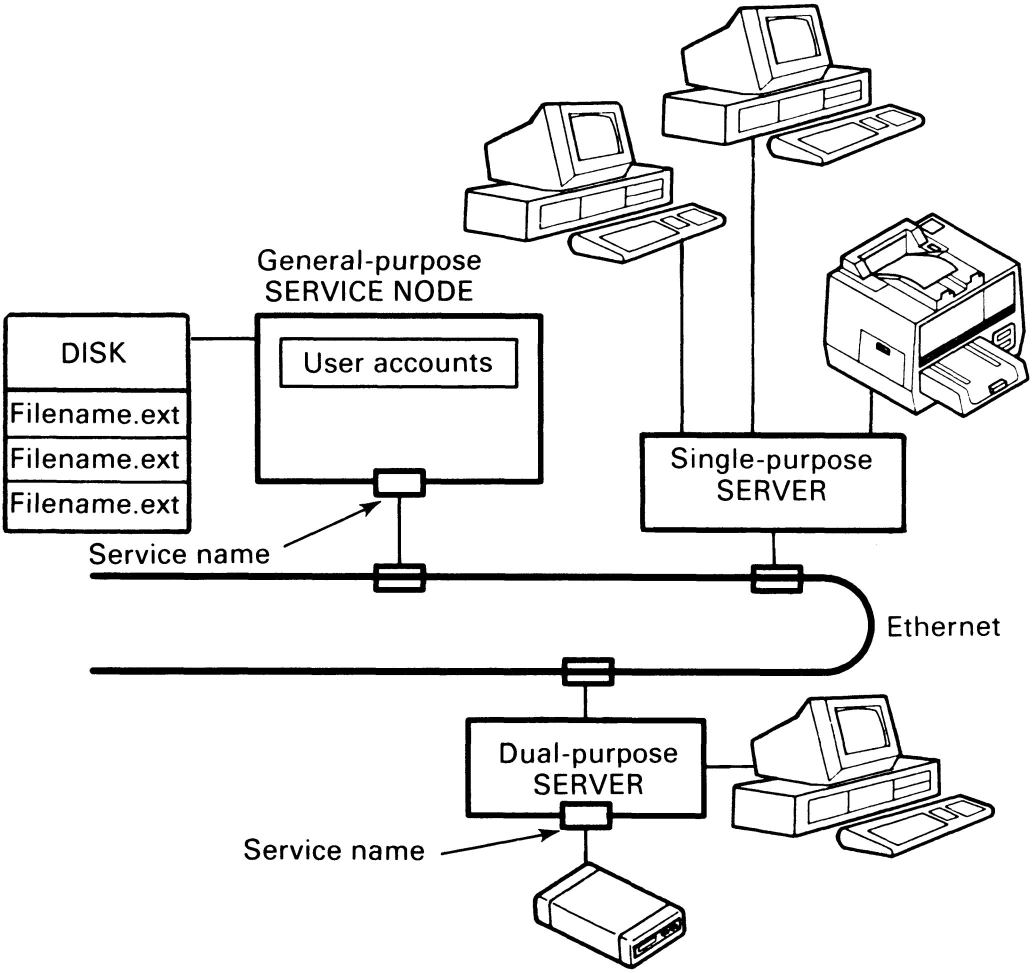
A **terminal server** is an Ethernet communications device that is a special-purpose computer that, minimally, runs server software.

A **service node** is any LAT node that runs LAT service-node software. Two types of service nodes are possible: a multiuser computer or a terminal server that runs service-node software as well as service software (a dual-purpose server).

- The first type of LAT service node comprises multiuser Digital computer systems that contain service-node software. These service nodes are called **general-purpose service nodes**.
- The second type of LAT service node comprises dual-purpose Digital terminal servers, which contain both service-node software and server software.

Figure 1-4 shows a sample three-node LAT network that contains a general-purpose service node offering itself as a service and two terminal servers. One terminal server is a single-purpose terminal server that runs only server software; the other is a dual-purpose server that runs both types of LAT software, supports a personal computer in terminal-emulation mode, and offers a modem as a service.

Figure 1-4: Sample Three-Node LAT Network



1.4.1 Overview of Terminal Servers

Terminal servers provide a cost-effective means of connecting many terminal users to computers and other computing resources. The terminal server reduces the hardware interrupt load on CPUs, enabling service nodes to devote more computing resources to managing users' applications.

Terminal servers have from 8 to 128 ports that support asynchronous ASCII devices. Table 1–1 lists and briefly describes Digital's terminal servers.

Table 1–1: LAT Terminal Servers

Terminal Server	Description
DECserver 100	An 8-line server with RS-232-C interfaces without modem-control software.
DECserver 200	An 8-line server with two models: a modem-control model (200/MC) using RS-232-C interfaces; a data-leads-only model (200/DL) using DEC423 interfaces.
DECserver 500	A server with from 2 to 8 line cards. The line cards can be any combination of 8-line modem-control line cards (with RS-232-C interfaces) and 16-line data-lead-only line cards (with DEC423 interfaces). A server with all data-leads-only lines would have a minimum of 32 and a maximum of 128 lines; a server with all modem-controlled lines would have a minimum of 16 and a maximum of 64 lines.
Ethernet Terminal Server†	A 32-line server with RS-232-C interfaces and modem control.

†Throughout this guide the Ethernet Communications Server Terminal Server is called the Ethernet Terminal Server.

(Continued on next page)

Table 1-1 (Cont.): LAT Terminal Servers

Terminal Server	Description
MUXserver 100	An 8- to 16-line server with RS-232-C interfaces or RS-422-A interfaces. It provides the same functions as the DECserver 100. The MUXserver 100 extends a LAT local area network by allowing remotely located asynchronous terminals to access LAT services. These terminals are attached to DECmux II statistical multiplexers at remote sites from the LAN. Each multiplexer provides 8 terminals with a connection to a MUXserver 100.

All terminal servers support Digital interactive terminals. (People using interactive terminals on terminal servers are called **server users**.) All terminal servers also support Digital personal computers in both terminal-emulation and file-transfer modes. In addition, all servers support Digital printers that are used by applications programs on general-purpose service nodes (which must be suitably configured to support this function). Servers with modem-control capability also support multiuser computers that lack LAT software (**non-LAT hosts**) and Digital-compatible modems. Dual-purpose servers act as front-end processors by supplying LAT service-node software for non-LAT hosts and other devices.

Modem control is very important for the security of non-LAT hosts. However, some ports on servers containing service software lack modem control and should never be used with non-LAT hosts. Table 1-2 indicates whether each server offers modem control on a given type of port and whether the server can function as a service node.

Table 1-2: Modem Control and Service Software on Terminal Servers

Terminal Server	Functions	
	Modem Control	Service Software
DECserver 100	No	No
DECserver 200/MC (RS-232-C ports)	Yes	Yes
DECserver 200/DL (DEC423 ports)	No	Yes
DECserver 500		
—RS-232-C ports	Yes	Yes
—DEC423 ports	No	Yes
Ethernet Terminal Server	Yes	Yes
MUXserver 100	No	No

Server software always defines a number of characteristics for a terminal server as a whole and for its ports. Initially, server software contains a default value for each of these characteristics. Terminal server commands enable showing and modifying the values of these characteristics.

Terminal servers can initiate and maintain sessions between their own ports and service nodes. Servers initiate sessions in response to either of two types of connection requests: user-initiated requests or host-initiated requests.

- A **user-initiated request** is a connection request from a user on a server asking that server to initiate a session between the user's terminal and a specific service. The server CONNECT command allows server users to make these user-initiated requests.
- A **host-initiated request** is a connection request from a suitably configured general-purpose service node asking a server to initiate a session with the service node. The session connects an applications device on a server port to an application such as a print queue on the the service node. The need for host-initiated requests reflects the fact that service-node software lacks the capacity to establish sessions.

1.4.1.1 Devices Supported by All Terminal Servers

The variety of possible configurations for server ports permits them to support a number of devices. All terminal servers support terminals, personal computers, and printers. Table 1-3 lists generally supported models of these universally supported devices. Note that dual-purpose servers support additional types of devices, which are described in Table 1-6.

Table 1-3: Devices Supported on All Terminal Servers

Device	Generally Supported Models†
Terminals	The VT100, VT101, VT102, VT125, and VT131; the VT220, VT240, and VT241 terminals; and the LA12, LA34, LA35, and LA38 keyboard printers
Personal computers	The Rainbow 100A, 100B, 100+, and 190; the DECmate II and III; and the Professional 325, 350, and 380 personal computers
Printers	Asynchronous RS-232-C printers, including the LA50 and LA210; the LQP02 and LQP03; and the LN01S, LN03S, and LCP01 printers

†Most servers support additional terminals (such as the VT52 or VT300 series), personal computers (such as the VAXmate), or printers (such as the LXY12-DA). For complete information, see the software product description (SPD) for each server.

Terminal servers treat attached devices as "dumb" devices; that is, a terminal server provides only the data path to an attached device. Applications connecting to the devices must manage the device-specific features.

For a detailed description of terminal servers, see Chapter 2.

1.4.2 Overview of Service Nodes

Service nodes contain service-node software. They can provide services, and they can support sessions between terminal servers and those services. A service node announces its services to terminal servers using a service-announcement message, which the service node multicasts over the network at regular intervals.

1.4.2.1 General-Purpose Service Nodes

A number of Digital operating systems contain service-node software and can function as general-purpose service nodes on LAT networks. The service-software of general-purpose service nodes is generally packaged with operating system software. Therefore, the manner in which the LAT protocol is implemented depends to some extent on system-specific considerations. Table 1-4 lists these systems.

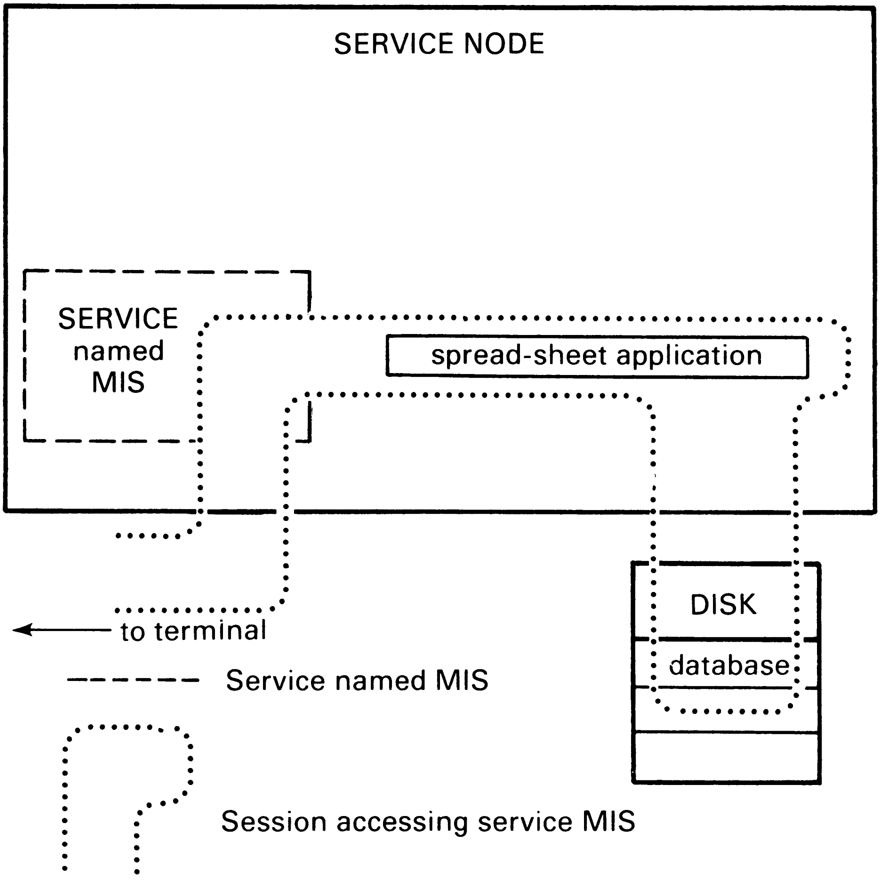
Table 1-4: General-Purpose Service Nodes

General-Purpose Service Nodes	Comments
VMS and MicroVMS systems	Supports host-initiated requests
ULTRIX-32 and ULTRIX-32m systems	
RSX-11M-PLUS system	Packaged with DECnet/RSX
Micro/RSX system	Packaged with DECnet/RSX
TOPS-10 system	
TOPS-20 system	

The services of general-purpose service nodes can be the system itself or one of its logical functions, such as accessing databases, processing text, performing calculations, accessing DECnet and other networks, and programming. When the full system is offered as a service, the server user logs into an account just as does a terminal attached directly to the service node or to a terminal switch. The server user can access the files in that account and any logical computing function of the system for which that account has privileges.

Figure 1-5 shows a database management program and associated database that are offered as a service named MIS. This service is being accessed from a terminal on a terminal server. The figure illustrates the computing resources that an MIS session accesses. These resources include a spread sheet and a database.

Figure 1-5: Applications Program Accessed as a Service



LKG-1221-87

Applications programs on some general-purpose service nodes can use specialized services offered by servers, such as modems or printers. This ability depends on the capacity of the service nodes to make host-initiated requests of a terminal server. A host-initiated request specifies that a server establish a session between two ports: a server port that is configured for an applications device and a specified applications port on the requesting service node. During the resulting session, an applications program running on the service node controls the applications device for an operating-system user. Note that an applications device offered by a server is called a **remote device** in some service-node documentation.

A VAX/VMS service node (VMS V4.2 and later) can be configured to make host-initiated requests for applications devices on behalf of applications programs. Note however, that for VMS V4.2 systems, this configuration requires installation of new service-node software called LATplus, which replaces the form of service-node software that is packaged with VMS V4.2 systems. LATplus is a VMS layered product that comes in server software distribution kits for VMS V4.2 load hosts. For later VMS versions, this new service-node software will replace the old version of VMS service-node software and will be packaged with the operating system.

NOTE

For other general-purpose service nodes, see the appropriate Software Product Description (SPD) to learn whether their applications programs can access applications devices on terminal servers.

1.4.2.2 Dual-Purpose Servers

Several Digital terminal servers contain service-node software as well as server software. The servers, called dual-purpose servers, can function as service nodes. However, some of these servers lack modem-control capability on some or all of their ports. Ports without modem-control are limited to offering terminals and printers as named services. Table 1-5 lists the dual-purpose servers and indicates which ports provide modem control.

NOTE

Note that a dual-purpose server is often called the **local service node** in server-specific documentation. This usage allows server managers to distinguish their own service-node management tasks from those of system managers managing general-purpose service nodes.

Table 1-5: Dual-purpose Servers

Dual-purpose servers	Limitations on modem control
Ethernet Terminal Server	No limitations — all ports provide modem control.
DECserver 200 system	Modem control requires the 200/MC model.
DECserver 500 system	Modem control requires the CXY08 line card.

Like any service nodes, dual-purpose servers offer services, announce them, and support connections to them. Dual-purpose servers can also accept a host-initiated request that specifies a service name representing one or more server ports configured for applications devices. A dual-purpose server can offer a service at some ports and support interactive server users at others.

The services on terminal servers include applications devices and hosts that lack LAT service-node software (non-LAT hosts). A dual-purpose server allows server users and service-node applications to access these devices and hosts. The server provides the necessary service-node software and Ethernet interface hardware (which physically connects nodes to an Ethernet). In this way the server opens the LAT network to many devices that are otherwise incapable of operating on the network.

Figure 1-6 illustrates the sorts of services offered by dual-purpose servers functioning as service nodes. The typical devices offered as services are modems, personal computers, printers, and non-LAT hosts.

Figure 1-6: Devices Offered as Services by Dual-Purpose Servers

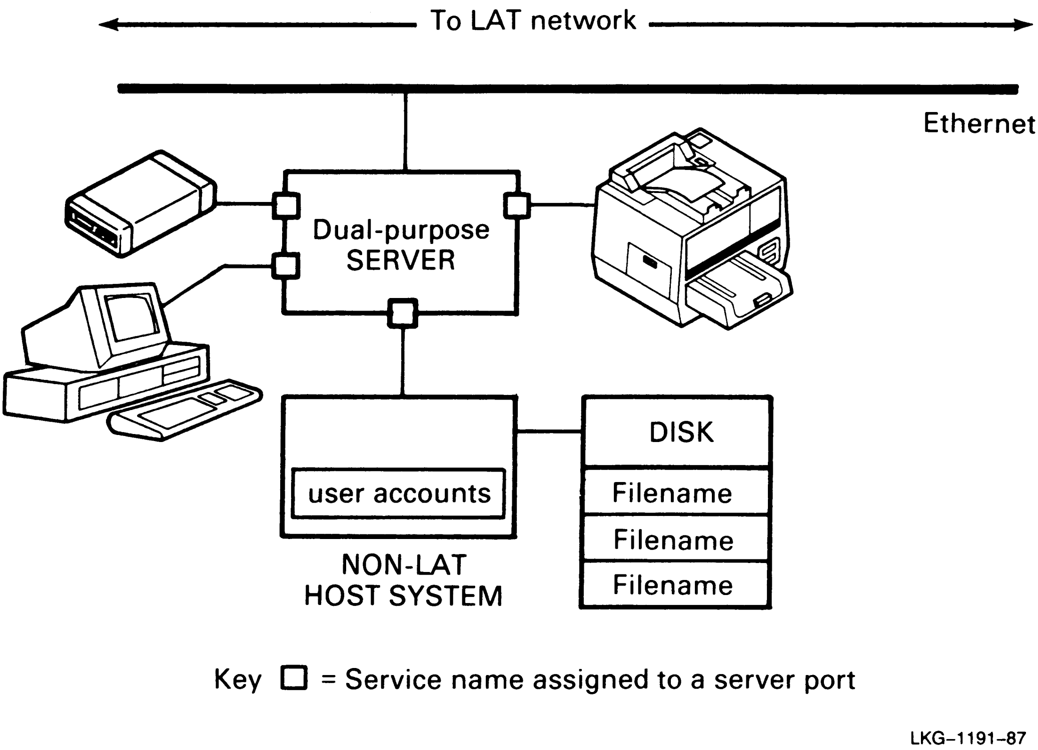


Table 1-6 indicates the sorts of devices supported only by dual-purpose servers.

Table 1-6: Devices Supported by Dual-Purpose Servers

Device	Models Supported
Non-LAT hosts	RT-11; RSX-11M; possibly non-Digital hosts that support XON/XOFF signals and RS-232-C interfaces, unless some communications requirement of such a host prevents it from communicating with a terminal server
Modems	DF02, DF03, DF112, DF124, and DF224 full-duplex asynchronous modems for either dial-in or dial-out use

Note that dual-purpose servers also support the types of devices described in Table 1-3.

For a dual-purpose server to offer one or more devices as services, the server manager must assign values to the server's service characteristics. These include serverwide characteristics that control the service-node communications functions of the server (such as announcing services) and service-specific characteristics (such as service name and which server ports offer it). There are no default services on a terminal server, and, thus, no service characteristics until a service is created.

For more information on service nodes, see Chapter 2.

1.5 Benefits of LAT Networks

LAT networks offer a wide range of benefits. Service nodes benefit from the management of terminals and sessions by servers. Server users benefit from an extensive access to computing resources. Network managers benefit from simplified hardware requirements for service nodes and non-LAT hosts and increased utilization of computing resources. Also, the performance of Ethernet communications is enhanced. Specific benefits to each of these groups are listed in this section.

Benefits to Service Nodes

- Processing of character interrupts is off-loaded from the operating systems to the terminal server, reducing the operating-system overhead.
- Flow control of terminal output is handled by terminal servers rather than by CPU's.

- At regular intervals, the server software places the accumulated user data for all the sessions with a given service node into a single message. This process is called multiplexing.

  Multiplexing has several benefits, including:

  - Eliminating blocking of ports on service nodes. For server users, sessions are independent of on the number of physical ports or the capacity of terminal switches to physically route connections. The number of possible sessions depends only on the service node's internal resources, such as its process limits or available memory.
  - Reducing traffic on the LAN, thus using Ethernet bandwidth efficiently.
  - Usually, reducing the I/O processing required of service nodes.
- By having more than one character per interrupt, terminal servers help reduce interrupts for service nodes and for non-DMA (direct memory access) devices, such as the DZ-11 asynchronous UNIBUS multiplexer,
- Through a process called load balancing, terminal servers establish sessions with the least busy service node that offers a service. Load balancing is especially useful on a VAXcluster for balancing the terminal load at log-in time among the individual processors of the cluster members.

Benefits to Interactive Users

- A server user can connect to a range of services (although the server manager can restrict the access of any terminal to services). The server user appears to be directly connected to the service being used.
- Once connected to a service, a server user can, by default, suspend a current session, create another session, and return to the original session at any time.
- If a service node supporting a session goes off-line, the server attempts to establish another session with the same service on another service node through a process called automatic failover. Automatic failover is especially important in VAXcluster configurations where server users can regain access to their accounts.

Benefits to Network Management

- Terminal servers handle communications for all of their ports over a single Ethernet controller rather than over multiple interfaces. This results in a significant cost-per-port reduction and simplifies cabling.
- Distributing terminal servers around a building, for example, by placing them in communications closets, can simplify wiring and reduce costs. There is only one Ethernet wire from the closet to the LAN and a number of short asynchronous lines from the closet to offices.
- Using terminal servers and service names reduces the impact of changing the physical configuration of a network.

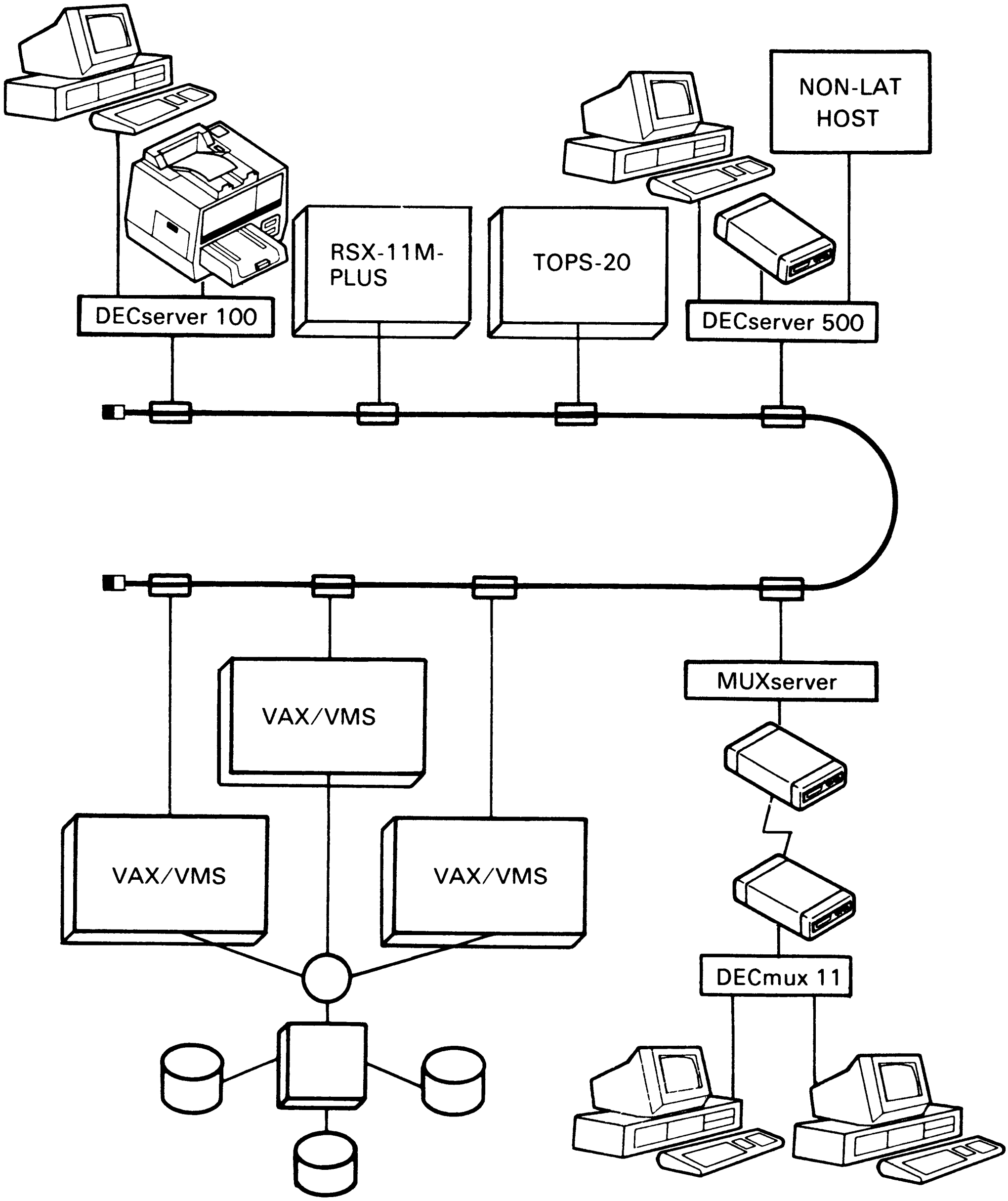
Performance Benefits

- The LAT protocol uses Ethernet bandwidth efficiently. Bandwidth, in this context, is the amount of data that can be transmitted in a specific period of time. For example, the maximum bandwidth of an Ethernet LAN is 10 Mbps (megabits per second).
- The LAT protocol provides similar performance to that provided by direct memory access (DMA) communications devices, such as the DMF32.
- The LAT protocol provides maximum full-duplex communications.

1.6 LAT Network Configurations

An Ethernet LAN can have up to 1024 nodes. A variety of communications protocols can coexist on an Ethernet LAN and, in many cases, within a given node. For example, DECnet and LAT normally coexist on a single Ethernet LAN, and service nodes are often also DECnet nodes. Figure 1-7 shows a LAT network containing several different types of LAT nodes. In this figure, all the nodes are LAT nodes. The DECserver 100 and MUXserver 100 are single-purpose service nodes. The DECserver 500 is a dual-purpose service node. The RSX-11M-PLUS, TOPS-20, and VAX/VMS systems are LAT service nodes and might also be DECnet nodes.

Figure 1-7: Sample LAT Network with Several Types of LAT Nodes



LKG-1192-87

LAT has its own transport mechanism and does not use the DECnet transport mechanism. For physical communications, the LAT protocol runs directly over the Ethernet, which consists of an Ethernet physical channel and the Ethernet protocol. The physical channel includes a coaxial cable, connectors, transceivers, transceiver cables, and so forth. The Ethernet protocol transmits messages over an Ethernet LAN.

1.6.1 Logical Partitioning of LAT Networks Using Groups

Server users on any terminal server on the LAT network potentially have access to every computer on the network. However, the LAT protocol defines a method of logically partitioning terminals and services into manageable subnetworks. These subnetworks are independent of the physical distribution of their members. **Groups** are logical subnetworks that are identified by decimal integers from 0 to 255.

Groups control communications between the following:

- Server ports being used with interactive devices or with applications devices that receive host-initiated requests. The groups assigned specifically to a server port are called **port groups**. Minimally, a server participates in all of its port groups; the sum of the port groups available on a server is called **server groups**.

  NOTE

  For most types of terminal servers, port groups determine server groups; however, for Ethernet Terminal Server 2.2 (and earlier) server groups determine the possible port groups.

- Service nodes (including the dual-purpose servers when functioning as service nodes)

As part of service-announcement messages, service nodes indicate the groups to which they belong. In processing these messages, a terminal server compares the service-node groups to its own port groups. If the server finds any common group, it places the service node and service information in the appropriate server directories.

If a service node and a server port share at least one group, the server can initiate sessions with all services offered by that service node. However, only the server ports that share a group with that service node can connect to its services. If a service node and a server port lack any common group, the port has no access to the service node or to information about it.

The group selection on a port-by-port basis allows different ports to see the network differently. This view of the network is established on on a need-to-know basis. A server user can establish sessions only with services offered by service nodes that share at least one enabled group with the user's port. Therefore, connection requests from that port for other services are rejected by the server. Host-initiated connections are also constrained by port groups: the port(s) being requested must share at least one group with the requesting service node.

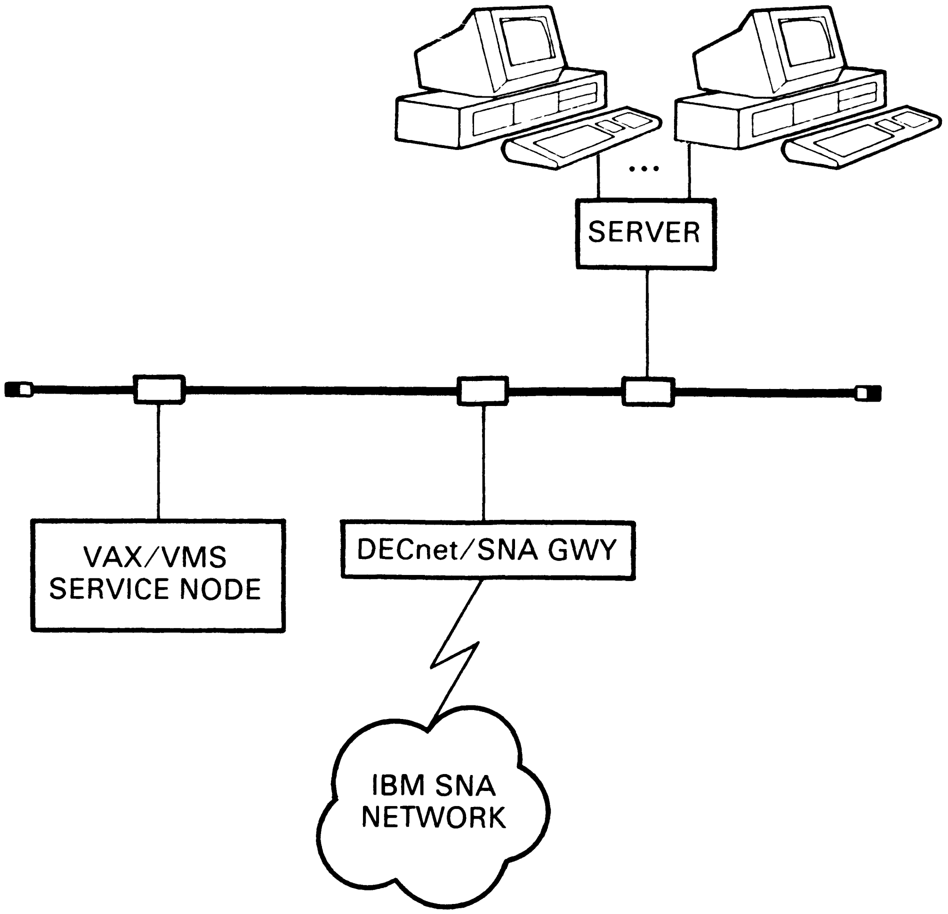
Assigning groups is discussed in Section 5.8.1.3. Figures 5-2 through 5-6 in that section show how a sample LAT network might be divided into four logical networks by assigning four groups—1, 6, 8, and 9—to service nodes and to server ports (or servers).

1.6.2 Access to Other Networks from a LAT Network

If the LAN supports other network protocols such as DECnet, LAT server users can access other networks. Many general-purpose service nodes offer their own computing resources as LAT services. A service node that is also a DECnet node gives server users access via a LAT service to other DECnet nodes on the local area network. Furthermore, a DECnet node potentially can provide a service that permits server users to access a wide area network, such as a PSI network or an SNA network. For example, a DECnet VAX service node that contains DECnet/SNA VMS access software allows server users to access an DECnet/SNA gateway. The gateway, in turn, allows server users that are using the access software to connect to an SNA network. Figure 1-8 illustrates an Ethernet LAN that includes a DECnet/SNA gateway connecting to an SNA network.

Furthermore, if any of these DECnet nodes contains gateway software, the server user can access non-Digital networks. For example, a VAX/VMS system containing DECnet/SNA VMS access software allows DECnet users to access a DECnet/SNA Gateway. Since the VAX/VMS system contains LAT service-node software, it can offer itself as a service to LAT server users. By logging in to the VAX/VMS service, a LAT server user with suitable privileges can use the DECnet/SNA VMS access software of the service to access an IBM host. Having gained access to the IBM host, the server user can request resources on the IBM system.

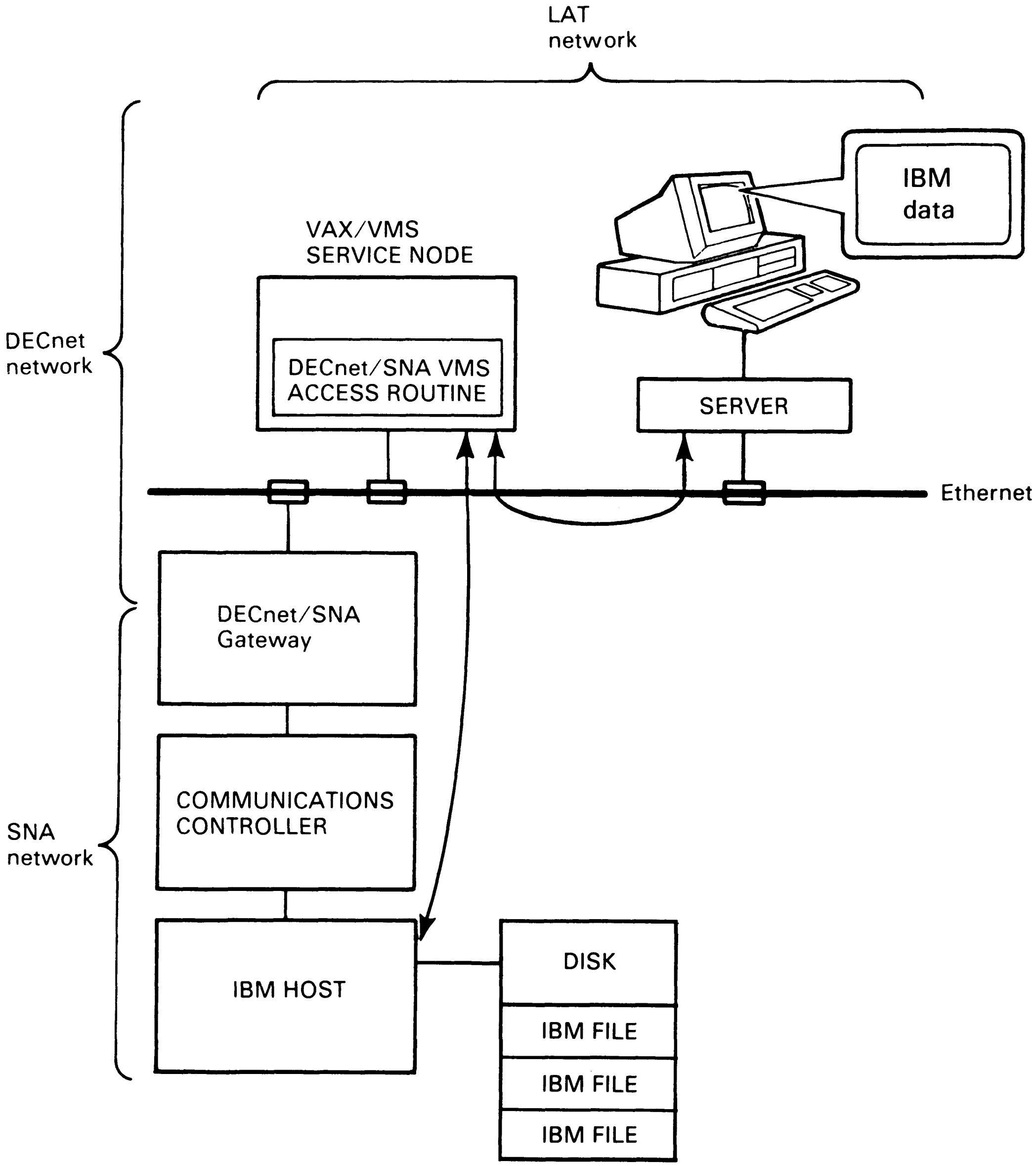
Figure 1-8: Example of an Ethernet LAN with a DECnet/SNA Gateway



LKG-0671

Figure 1-9 illustrates this series of communications events. In this figure, a LAT server user has accessed an IBM system from a VT220 terminal attached to a LAT terminal server. The user is accessing a file stored on an IBM system disk and is displaying data from that IBM file on the VT200 screen.

Figure 1-9: LAT Server User Accessing an IBM file



LKG-1193-87

1.7 LAT Management Tasks

LAT network documentation divides management tasks into three job responsibilities. A specific job title is used for each role: LAT network manager, system manager, and server manager. A **LAT network manager** coordinates the overall performance of the LAT network. A **system manager** manages a multiuser computer that functions as a general-purpose service node; in this role, a system manager functions as a service-node manager. A **server manager** manages one or more terminal servers. A server manager that manages a dual-purpose server is also a service-node manager.

Since job titles and job responsibilities vary widely in practice, the responsibilities used in LAT documentation are summarized in Table 1-7. In reading these job responsibilities, keep in mind that for small networks, one person often performs two or all of these jobs; for large networks, a team of people may share the job responsibilities of a single job title.

Table 1-7: Job Titles and Job Responsibilities in LAT Documentation

Job Titles	Job Responsibilities
Server Manager	Customizing and maintaining one or more terminal servers Configuring ports Creating services on dual-purpose servers Diagnosing and resolving server problems
System Manager†	Installing server distribution software onto load hosts Customizing the load host's node database Creating services on general-purpose service nodes Setting up service nodes to make host-initiated requests
LAT Network Manager	Planning the LAT network components and configuration Preparing for server hardware and software installations Coordinating LAT node names, service names, and group assignments

†Some organizations use the term system programmer or systems analyst.

Chapter 5 introduces the activities that are involved in setting up and managing a LAT network. For information on how to do each activity, see the installation and management documentation of specific LAT nodes. Routine management activities for server, system, and LAT network managers include:

- Preparing for server hardware installation
- Setting up load hosts and down-line loading each server's image to the server
- Customizing a server and some or all of its ports for interactive devices
- Setting up and managing service nodes and their services
- Setting up servers and general-purpose service nodes for host-initiated requests
- Managing communications on a LAT network

Chapter 5 also introduces the Terminal Server Manager (TSM) product, which is a centralized management facility for terminal servers. TSM is an essential tool for server managers and LAT network managers. TSM simplifies the management process, while also making it flexible and efficient.

1.8 Summary

This chapter has introduced LAT terms and concepts that are essential for understanding the discussions in the remainder of this guide. This section recaps the most significant terms and concepts.

- Architecture and protocol

  A LAT network consists of LAT nodes, their network interfaces, and the Ethernet LAN. LAT networks implement the LAT architectural model and associated LAT protocol. Together these describe the communications functions of LAT and the specific message sequences used in communication among LAT nodes.

- Service

  A service is a named resource that is either specific hardware (such as a dial-out modem), a combination of hardware and software (such as a VAX/VMS or a non-Digital computer), or a logical function (such as an application program).

- Session

  A session is an asynchronous, two-way logical connection between a service node and a terminal server.

- LAT software

  The software that implements the LAT protocol is termed LAT software. The presence of LAT software is the only feature that defines a LAT node.

- LAT nodes

  There are two types of LAT nodes: terminal servers and service nodes. A terminal server is an Ethernet communications device based on a special-purpose computer that, minimally, runs server software. A service node is any LAT node that runs LAT service-node software. Two types of service nodes are possible: a multiuser computer called a general-purpose service node or a terminal server called a dual-purpose server that contains both server and service-node software. Note that a dual-purpose server is often called the local service node in server-specific documentation.

- Connection requests

  People using interactive terminals on terminal servers are called server users. A user-initiated request is a connection request from a user on a server asking that server to initiate a session between the user's terminal and a specific service. A host-initiated request is a connection request from a suitably configured general-purpose service node asking a server to initiate a session with the service node.

- LAT groups

  Groups are logical subnetworks that are defined by decimal integers from 0 to 255. Groups control communications between service nodes (and their services) and server ports. Groups assigned to a service node are called service-node groups. Groups assigned specifically to a server port are called port groups. Minimally, a server participates in all of its port groups; the sum of the groups available on a server is called server groups.

- LAT management

  LAT network documentation divides management tasks into three job responsibilities. A specific job title is used for each role: LAT network manager, server manager, and system manager.

2 LAT Nodes

This chapter discusses the nonarchitectual features of LAT nodes. These features combine to support LAT communications functions and to give managers control over those functions and to give users access to them.

This chapter first discusses terminal servers in terms of features related to server-software functions. The chapter then discusses some of the features that distinguish general-purpose service nodes and dual-purpose servers. Terminal servers are dedicated to implementing LAT software, and general-service nodes are fundamentally multiuser systems. Therefore, the focus of this chapter is on the functions of terminal servers.

2.1 Terminal Servers

As discussed in Chapter 1, a terminal server is an Ethernet communications device based on a special-purpose computer that, minimally, runs server software. Server software enables a terminal server to establish and manage sessions between itself and a service node. The mechanics of session establishment and management are invisible to the server user. Server users trigger these processes by requesting and disconnecting sessions and shifting among multiple sessions.

This section discusses the following items:

- Loadable server images
- Modifiable characteristics of the servers
- The databases where the values of characteristics reside
- The basic configurations of server ports
- Security features
- The server command interface and on-line help
- Support for file transfer applications

Part or all of the server software for each type of terminal server is distributed separately from server hardware in software distribution kits. Software distribution kits are installed onto one or more multiuser computers that function as load hosts for one or more servers. The file that contains server software is called the server image file. In addition to server software, which implements LAT protocol functions for a server, the server image also contains additional software that is used by servers; for example, the server's on-line help. A server image provides part or all of the server software to the running server. Note that its contents vary among different types of servers. A server remains inoperative until its server image is down-line loaded from one of its load hosts. For a description of load hosts, see Section 5.3.

The loadable server image of terminal servers permits software updates without requiring hardware changes.

NOTE

> Servers also have ROM-based software (also called firmware) that implements a hardware self-test and enough of the Maintenance Operations Protocol (MOP) to communicate with the load host for down-line loading the server image.

2.1.1 Modifiable Characteristics

Servers have three types of characteristics, whose values determine server operation. These characteristics include those for the server as a whole (server characteristics), those of server ports (port characteristics), and those of a service offered locally by a dual-purpose server (local-service characteristics).

2.1.1.1 Server Characteristics

Server characteristics include characteristics that perform the following sorts of functions:

- Identify a server (for example, server name, identification string, and number)
- Regulate performance of LAT communications functions (for example, timer values, node limit, and retransmit limit)
- Control remote maintenance activities (for example, a maintenance password)
- Provide security on interactive ports (for example, log-in and privileged passwords and the capacity for inter-port broadcasting)
- Control access by interactive ports to LAT services (for example, the maximum number of sessions and, for some servers, server groups)
- Manage a dual-purpose server as a service node (for example, service-node groups and capacity to multicast service announcements)

2.1.1.2 Port Characteristics

Port characteristics include characteristics that perform the following sorts of functions:

- Identify a port (user name and, for some servers, port name)
- Specify physical characteristics (for example, speed, parity, character size, and modem control)
- Control access by an interactive port to LAT services (for example, the maximum number of sessions, port groups, and preferred or dedicated services)
- Customize user environment (for example, switches for changing between sessions and modes, display of broadcasted and other intraserver information, and security status)
- Manage flow control of any session

2.1.1.3 Local-Service Characteristics

Local-service characteristics include characteristics that perform the following sorts of functions:

- Create a local service (by specifying a service name, identification string, and port assignments)
- Manage a local service (for example, a service password and the capacity to queue connection requests)

2.1.1.4 Server Databases for Characteristics

A server manager can change the values of the server, port, and local-service characteristics on a long- or short-term basis. The duration of a setting for a value depends on the database in which the change occurs. A terminal server has three databases that contain values for characteristics that control the actual operation of a terminal server. These databases consist of a permanent database, an operational database, and a log-in database.

- **Permanent database:** This database contains permanent values for server, port, and local-service characteristics. A **permanent value** functions as a system-specific default. By changing the original permanent value of a characteristic, a server manager can override the standard defaults for a particular server. To implement changed permanent values, the server manager loads the server. Permanent values are copied to the operational database whenever the server experiences a down-line load.

  The location of the permanent database and, therefore, the customization procedures varies among different types of terminal servers. There are two distinct approaches to storing permanent databases. Each type of server uses only one approach.

  Ethernet Terminal Server and DECserver 500 systems: Each of these servers has its own individual server image, which resides in a server image file on one or more load hosts. The permanent database resides in a server's server image(s). Customization of the permanent database must occur on a load host.

  The **Terminal Server Configurator** (TSC) is a utility that operates on the individual server image files of Ethernet Terminal Server and DECserver 500 systems. The TSC utility resides and executes on one or more load hosts. By implementing the server DEFINE commands, the TSC lets a server manager customize the permanent database of those servers.

After using TSC on one load host to revise the permanent database for a specific server, a server manager must copy the server's image file to each of its load hosts.

When a server's newly customized server image is down-line loaded, the new permanent values become the operational values of the server. Note that TSC does not operate on a server's operational database, which can be changed only by using SET commands issued on the server.

The user interface for the TSC is standard across operating systems. It is documented in the *Terminal Server Commands and Messages Reference* or in a server's *Operations Guide*, depending on the server or software version. However, the command(s) for starting it are system-specific. They are documented in a server's software installation guide for an operating system.

Note that the Terminal Server Manager (TSM) product provides an alternative to the TSC. The TSM provides a number of additional features and operates on all terminal servers. For an overview of the TSM, see Section 5.15.

If one of these server units is swapped with a different unit, customizing the new unit is simple. The server manager:

1. Attaches each device to the new unit at the same port number it had on the original unit. Note that using the same port number is necessary only for a port with a device-specific (or user-specific) configuration.
2. Assigns the name of the original unit to the new unit.
3. Changes the hardware address associated with that name in each load host's node database (whose contents include DECnet addressing information).
4. Down-line loads the original unit's server image to the new unit.

DECserver 100, MUXserver 100, and DECserver 200 systems: On these servers, the permanent database resides in the server hardware, within nonvolatile random-access memory (NVRAM). Customizing these servers occurs on the running server, after the initial down-line loading of the server image. The server command sets of these servers contain DEFINE commands, which are used to change permanent values in their permanent databases.

Newly specified permanent values for port characteristics take effect the next time the port logs in to the server. Logging in a server port causes the server to place the port's newly defined permanent values into the permanent database. However, newly specified permanent values for server or local-service characteristics take effect only after the server is reloaded. Loading the server causes it to place the newly defined permanent values for server and local-service characteristics into the permanent database.

If one of these servers is swapped with a different unit, the new unit must be customized, including having its ports reconfigured from scratch. For this reason, the Terminal Server Manager product is particularly useful for any server whose permanent database resides within the server (see Section 5.15 for an overview of this product).

- **Operational database:** When the server image is down-line loaded to the server, it supplies start-up values to a modifiable database called the operational database. This database stores operational values of server, port, and local-service characteristics. An **operational value** controls how a characteristic operates on a running server. Unless the operational values of a newly loaded server are reset, they match permanent values.

  The operational database resides on a running server in dynamic memory, and newly set operational values take effect immediately. However, they are temporary. When a port logs out of the server, the operational values of its characteristics are replaced by permanent values. The operational values of server and service characteristics are replaced at downline loading by the values of the permanent database.

- **Log-in database:** This database exists only on a server whose permanent database resides on a load host. Using the server SAVE PORT command, a server manager makes long-term changes to the port values on a running server. These saved values replace the permanent values of port characteristics until the server is reloaded. At that time saved values are overwritten by permanent values.

2.1.2 Software Port Configurations

A server manager can configure a server port for a range of devices by changing one or more of the values of the port's characteristics. Server managers configure each server port to meet the needs of its attached device. Note, however, that there are hardware constraints on server ports; for example, modem control requires that port hardware be an RS-232-C interface. Thus, although the two DECserver 200 models use identical software, only the model with RS-232-C interfaces (the DECserver 200/MC) supports modem control. (Note that RS-232-C ports do not require modem control software to operate; for example, the DECserver 100 has RS-232-C interfaces but lacks modem control hardware and software.) Even where available, modem control can always be disabled when the port configuration does not require it.

The basic port configurations for terminal servers include:

- Fully interactive ports: This port configuration involves a terminal that has access to the local mode of the server as well as to services. **Local mode** is the environment in which a server user (including a server manager) interacts directly with the server by entering commands at the local prompt (Local>). Generally, server users enter local mode when they log in to a server. They leave local mode whenever they connect to a service, but they can return to local mode at any time.

  The fully interactive port configuration allows terminal server users to enter some or all server commands. At the least, they can connect to one or more services, move among their sessions, personalize their server port, and show information about their port and the available services.

- Dedicated ports: This configuration dedicates an interactive terminal to access only one service called a dedicated service. A **dedicated service** is a service that has been designated by a server manager to be the only service accessible by an interactive server port. A server user connects automatically to a dedicated service by pressing the return key on the terminal. No server commands are available on the dedicated port.

- Dynamic access:† This configuration allows a device (such as a hardcopy terminal or a personal computer) to shift between functioning as an interactive terminal and an applications device.

† Host-initiated requests work with this configuration, regardless of whether the port belongs to a service.

- Remote access:† This configuration allows any terminal server to accept host-initiated requests for a particular port (as long as it is configured for an applications device such as a printer). Remote access also allows a dual-purpose server to accept user-initiated requests to use a device (such as a non-LAT host or modem) offered as a service by the server.

2.1.3 Security Features

Using the features discussed in this section, a server manager can ensure the security of a server and its resources by using passwords and by providing different levels of security.

2.1.3.1 Passwords

The server manager can improve security by specifying passwords for the server. The following passwords exist on servers:

- **Lock password:** This password prevents unauthorized access to an unattended local-access port. There is no default lock password. Rather, a lock password is designated for any given port by a server user entering the LOCK command. The server manager can disable the use of the LOCK command on a serverwide basis.
- **Log-in password:** This password controls log-in access to a server's local-access ports. There is a default log-in password, which the server manager can change. The server manager can enable this password on a port-by-port basis. However, only one log-in password exists on given a server; all of its ports that require the log-in password use the same one. There is a default log-in password, which can be changed by a server manager.
- **Maintenance password** ‡ : This password controls remote maintenance activities. Remote· maintenance activities include using the Remote Console Facility (RCF) and loading a server from a load host. There is no default maintenance password. By specifying one, a server manager enables the maintenance password feature.

† Host-initiated requests work with this configuration, regardless of whether the port belongs to a service.

‡ DECserver 100 and MUXserver 100 have no maintenance or service password.

- **Privileged password:** This password protects the server from unauthorized access to its privileged commands, which are discussed in Section 2.1.4.1. The privileged password always remains enabled. There is a default privileged password, which can be changed by a server manager.
- **Service password** ‡ : This password controls access to a particular locally offered service. There are no default service passwords. A server manager can specify a service password for each service individually.

2.1.3.2 Three Levels of Security for Interactive Ports

Security is provided in part by a three-level security feature that controls access to a server's command interface. The three-level security feature enables server managers to control access to the server command interface by modifying the security status of any interactive port. At a given moment, the status of a port is privileged, nonprivileged, or secure. Status reflects the segmentation of server users into three groups: server managers (privileged), advanced users (nonprivileged), and casual users (secure). The default status of a port is nonprivileged.

Note that a server manager can permanently configure a port for secure status but not for privileged status. Privileged status is created only by issuing the SET PRIVILEGED command at the port and entering in the correct privileged password. Privileged status disappears when the port is logged out (or the SET NOPRIVILEGED command is issued).

2.1.4 The Server Command Interface

The command interface of the server, which is described further in the *Terminal Server Commands and Messages Reference*, allows the server manager to control the server and its ports. In addition, the command interface allows users to connect to LAT services, to display information, to access on-line help, and to control some port functions. There are two types of server commands: privileged commands and user commands.

‡ DECserver 100 and MUXserver 100 have no maintenance or service password.

2.1.4.1 Privileged Commands

Privileged commands are intended for managing the server. By using privileged commands, a server manager can perform all the management functions available on the running server. These functions include:

- Customizing server characteristics
- Configuring server ports
- Establishing and managing local services (if any)
- Observing the status of the server, its ports, its services (if any), and its LAT network environment
- Performing various testing functions

To use privileged server commands, the server manager must enter the correct privileged password.

The server manager has access to both privileged and nonprivileged server commands. Nonprivileged commands constitute the commands using interactive ports (user commands). Even if a port has secure status, setting privilege on the port gives access to the full set of nonprivileged commands (which includes secure commands) as well as to privileged commands.

2.1.4.2 User Commands

The commands for server users permit them to access and use available services. These commands also permit users to control their server ports to some extent. On ports with either nonprivileged or secure status, the user commands allow a user to modify some of a port's characteristics, to obtain a display of available services, to establish sessions with services, to switch between established sessions, and to show the current state of the port and its sessions. In addition, nonprivileged status permits users to interact (to a limited extent) with other users and with other ports on the server. Such interaction includes showing information on the server and other ports and broadcasting messages to other ports.

2.1.4.3 On-Line Help

Terminal servers provide on-line help. Some sort of command reference exists on all servers. Reference help varies according to the security status of a port. Only commands that are currently available on a port appear in help screens.

The extent of on-line help depends on the memory capacity of a server. A small server such as the DECserver 100 has only a few screens of information, which is useful mainly to remind experienced server users of command keywords. In contrast, a server with ample memory such as the DECserver 500 has many screens of reference information with examples and default values.

Servers with sufficient memory space, such as the DECserver 200 and DECserver 500 systems, also provide tutorial help. This help provides new server users with an overview of basic nonprivileged commands.

2.1.5 Personal Computers on LAT Networks

LAT networks provide personal computers with a way to access computing resources. Personal computers can participate in LAT networks by using either of two methods:

- Operating through a terminal server port

  This method allows the personal computer to function interactively in terminal emulation or file transfer modes. With an appropriate applications program, a personal computer can use a printer offered as a service by a terminal server. In addition, on dual-purpose servers, the personal computer can be offered as a service to other personal computers on servers.

- Implementing LAT server software internally and operating directly on the Ethernet as a specialized single-port terminal server

  This method allows the personal computer to function interactively in terminal emulation or file transfer modes. The method requires a network terminal services (NTS) product that implements the LAT server protocol.

2.1.5.1 Personal Computers Attached to Terminal Servers

Personal computers that act like (emulate) Digital terminals (such as a VT100, VT220, or VT240) can access general-purpose service nodes and non-LAT hosts that are available as services on the same LAT network.

The server supports file transfers between personal computers on a local-access port and any compatible computer whose resources are offered as a service on the LAT network. The user of a personal computer attached to a terminal server can transfer files from and to another computer. The other computer, which can be a personal computer, a general-purpose service node, or a non-LAT host, must be offered as a service on the LAT network. Note that file transfer requires that the participating computers share a common file-transfer application. For example, Pro 350 systems can use the Pro File Transfer (PFT) utility.

File transfers occur within the context of a single session. Other sessions on the server port are unaffected, which allows a personal-computer user to move back and forth easily between terminal emulation and file transfer functions. If the user switches to another session during a file transfer, the file transfer is suspended until the user resumes the file-transfer session.

Data transparency during file transfers is often handled automatically by service nodes that implement the LAT V5.1 architecture (such as VAX/VMS 4.2 systems using the LATplus layered product as the service-node software). **Data transparency** refers to the manner in which flow control and other special characters are handled by the server during a particular session on an interactive port. Sometimes, automatic handling of data transparency is unavailable. In this case, server users specify the data transparency mode of a session using server commands.

2.1.5.2 Server Software on Personal Computers

An implementation of server software exists for some Digital personal computers. To distinguish the server software of personal computers from that of terminal servers, the server-software of personal computers is called **PC LAT** in this guide. PC LAT resides in a driver (which is often called LAT.EXE) that implements the server software. PC LAT is used by NTS terminal emulators.

Like other server software, PC LAT operates on an Ethernet local area network and accesses resources offered as LAT services by service nodes. The server software of PC LAT maintains service and service-node directories. If conditions permit, PC LAT can establish sessions with any LAT service whose service name appears in the personal computer's service directory.

The conditions under which session establishment fail include the absence of the PC LAT driver, the absence of the service name from the service directory, or the failure of a connection attempt.

Assigning groups is essential for effective use of PC LAT. Groups permit efficient use of the personal computer's memory and ensure that its user(s) can access the necessary services.

Digital NTS products that use the PC LAT driver are available for VAXmates and for IBM PCs. In addition, a LAT server-software implementation exists as part of the Pro/Comm software option for the Pro 350 and Pro 380 personal computers. There are several network terminal services (NTS) emulators one or more of which is supplied with an NTS product. These emulators are as follows:

- VT220 emulator

  This emulator supports multiple sessions outside of the terminal emulator by using the windowing capabilities of the Digital MS-Windows.

- VT240 emulator

  This emulator supports four multiple sessions within the terminal emulator. It is available for the VAXmate personal computer only.

- SETHOST utility

  This utility provides terminal emulators that support both LAT sessions and DECnet connections by using two distinct communications protocols: the server portions of the LAT protocol (PC LAT) and the CTERM protocol. Multiple sessions are possible. The **CTERM protocol** is a wide area virtual-terminal protocol that is layered on top of DECnet. The CTERM protocol is unrelated to the LAT protocol and is not considered here. When connecting a personal computer to another node on a LAN, the default communications method of the SETHOST utility is the PC LAT driver.

2.2 Service Nodes

As discussed in Chapter 1, a service node is any LAT node that runs LAT service-node software. Service software allows a service node to offer computing resources as named services to server users and to accept connections from server to these services.

Two types of service nodes are possible:

- General-purpose service nodes: Multiuser Digital computers, such as a VAX/VMS system, whose operating systems include LAT service software. See Table 1-4 for a list of systems that operate as general-purpose service nodes.
- Dual-purpose servers: Terminal servers, such as the DECserver 500, that contain both server and service-node software. See Table 1-5 for a list of dual-purpose service nodes.

This section discusses both types of service nodes.

2.2.1 General-Purpose Service Nodes

General-purpose service nodes offer their own computing resources as LAT services. These resources may include log-in access to user accounts or access to specific logical functions, such as a spread-sheet application.

Defining a logical function as a LAT service provides one method of giving terminal servers access to wide-area-network systems, such as DECnet nodes or DECnet/SNA gateways. For example, a system manager can setup a generic DECnet set-host account. A set-host account limits users to the DECnet SET HOST command but requires no log-in password. It is accessible to server users whose ports share a group with the service node. These server users can access DECnet nodes on the same local area or wide area network as the service node. A similar type of generic account is possible for other networking services, such as for a DECnet/SNA VMS access routine.

2.2.1.1 The LAT Control Program (LCP)

General-purpose service nodes contain management software called the **LAT Control Program** (LCP). LCP provides a command interface for setting up and managing the service nodes. Using LCP, system and network managers customize and monitor the LAT protocol on service nodes that are general-purpose computers.

LCP allows managers to set up and manage an operating system as a service node. Minimally, LCP controls the following activities:

- Customizing the characteristics of a service node
- Setting up, managing, and deleting services
- Starting and stopping the LAT service-node software (sometimes called the LAT driver)
- Displaying the current values of service node and service characteristics
- Showing and zeroing protocol-related counters

2.2.2 Dual-Purpose Servers

The second type of LAT service node consists of Digital terminal servers that contain service-node software. These terminal servers, which are called dual-purpose servers, contain both service-node software and server software. Note that, in terminal server documentation, a dual-purpose server is often called the **local service node**, and the services it offers are called **local services**.

The server privileged command set contains a number of service-node commands with which a server manager manages the server as a service node. Setting up a server as a service node minimally involves assigning service-node groups and defining services.

The server privileged command set also contains a number of service-node commands with which a server manager creates and manages services. Setting up a service involves configuring a server port using server commands and assigning the port to a service name. Note that a port configured to offer a service is capable of offering only a single session.

By default, as soon as a server manager establishes one service on a dual-purpose server, the server functions as a service node by issuing multicast service announcements. These announcements describe the server' s available services and contain identification information such as the server' s name and identification string.

Maintaining a service as a service node requires ensuring that the devices offered as services continue to operate. The server manager can prevent access to all local services or can control access to a specific local service when devices or server ports require maintenance.

The following sections discuss the types of devices that are offered as services on dual-purpose servers. Modems and non-LAT hosts (including personal computers) require modem-controlled server ports.

2.2.2.1 Printers as LAT Services

Terminal servers can support the use of serial printers (such as an LA120, LA75, or LN03). Both nonkeyboard and keyboard printers are supported. A server can also support printerlike devices such as plotters. See Table 1-3 for information on what sorts of printers function on terminal servers.

Printers offered as a service on a server can be accessed in two ways: host-initiated requests from properly configured service nodes or requests from a personal computer on servers. To access a printer service on a server, a personal computer must contain an applications program to drive the printer. The personal computer must be attached to another server port or contain the PC LAT driver. The personal-computer user enters terminal-emulation mode and then connects to the printer by requesting the server to connect to its service name.

2.2.2.2 Personal Computers as LAT Services

On a remote- or dynamic-access server port, a personal computer can be assigned to a service. When offered as a service, a personal computer is a non-LAT host. Offering a personal computer as a service allows other personal computers to connect to it for transferring files. See Table 1-3 for information on what sorts of personal computers function on terminal servers.

2.2.2.3 Modems as LAT Services

Servers with modem control allow any of their modem-controlled ports to be used with full-duplex asynchronous modems. The modem control sequences conform to the RS-232-C and CCITT V.24 standards. A modem connection can be to a leased line or to a switched line. The server supports modems that receive calls (dial-in modems), modems that initiate calls (dial-out modems), and modems that alternate between receiving and initiating calls (dial-in/dial-out modems). The physical location of the other modem is irrelevant. See Table 1-6 for information on what sorts of modems function on terminal servers.

2.2.2.4 Non-LAT Hosts as LAT Services

Dual-purpose servers can provide access for users to a non-LAT host, which is a general-purpose host that lacks LAT service-node software. A non-LAT host is usually connected to a server at an RS-232-C port using a null modem cable or a modem cable. The server port is enabled for modem control. See Table 1-6 for information on what sorts of non-LAT hosts function on terminal servers.

Offering a non-LAT host as a service allows a server user to request a connection to the non-LAT host. After connecting, the user's terminal appears to be connected directly to the non-LAT host. The server provides the service-node software for the non-LAT host. The functions provided by the server include sending data to the user's terminal and demultiplexing user data sent to the non-LAT host.

Security for Non-LAT Hosts

Modem control provides security for non-LAT hosts offered as services by a terminal server. When a server port is used with a non-LAT host, a modem cable or a null modem cable should connect them, and the server and host ports should have modem control enabled. This configuration allows the server and the non-LAT host to use modem signals to monitor each other.

Using modem control, both the server and the non-LAT host send data-terminal-ready (DTR) or data-set-ready (DSR) RS-232 modem signals as long as they are functioning. During modem-controlled communications, each device monitors incoming signals constantly. If either device drops DTR (or DSR), the other device detects the resulting signal loss and recognizes that the other device has ceased functioning.

The server and (it is assumed) a non-LAT host terminate any session for which the DTR or DSR signal ceases for a designated time interval. The process is as follows:

- On the server

  When a session is terminated at the host (such as when a user logs out), the host drops DTR for a null modem cable or DSR for a modem cable. The server detects a signal loss and terminates the connection. Without modem control, the dual-purpose server continues to maintain the session(s) (and virtual circuits to the port(s) at which the non-LAT host is attached).

- On the non-LAT host

  When a session is terminated at the server (such as by a user issuing the DISCONNECT command or by the server powering down), the server drops DTR. The host detects a signal loss and logs out the user. Without modem control, a non-LAT host would not detect that the session had been disconnected. It would, therefore, leave the user's process(es) and data accessible to the next user that creates a session to that non-LAT host port.

NOTE

Modem control involves much more that asserting DTR or DSR and watching DSR signals. It involves a modem-signal handshaking protocol for establishing and disconnecting point-to-point physical links over a switched network. Server implementations of the RS-232 modem protocol are described in terminal management or operations guides.

3

LAT Architecture

The **LAT architecture** is a communications model for resource sharing on an Ethernet local area network (LAN). LAT architecture reflects a distinct communications strategy for exploiting the Ethernet LAN environment. The architecture defines basic communications functions such as locating resources, controlling sessions, and transporting messages. It then assigns these function to specific architectural layers and provides specialized messages for each layer. Finally, LAT architecture defines features that support its communications functions; for example, service names, directories, buffers for storing user data, message timers, and so forth.

Implementation of the LAT architecture depends on the LAT protocol. The **LAT protocol** is the portion of the LAT architecture that specifies rules for formatting and sequencing the various messages defined by the architecture.

This chapter discusses the following general topics:

- Fundamental features of LAT architecture
- An overview of the architectural layers used by LAT networks
- The functions, messages, and flow-control mechanisms of LAT layers
- A VAX/VMS implementation of LAT service-node software

3.1 Fundamental Features of LAT Architecture

LAT architecture systematically divides its communications functions (along with their supporting features) between server software and service-node software. Because server software and service-node software implement complementary sides of the LAT architecture, these distinct types of LAT software operate as communications partners.

LAT communications involves partners exchanging messages containing protocol information and data. Each layer processes only information that originates from the corresponding layer of a partner (this is called **peer-to-peer communication**).

The LAT architecture exploits the following elements of Ethernet LAN environments:

- The bandwidth of the Ethernet cable is much greater than the bandwidth required by a single communications path. Therefore, simultaneous logical channels can exist.
- On a LAN or an extended LAN, names and addresses of LAT services and LAT nodes can be distributed and accessed in a decentralized manner. Routing messages is unnecessary.

The basic features of the LAT architecture reflect its layered nature and the elements of the Ethernet LAN environment. The following are some of the fundamental architectural features:

- LAT nodes dynamically configure their LAT network. Dynamic configuration means that LAT communications is not limited by hardware configurations. A number of factors contribute to the configuration process:
  - Server users request resources by logical service name without having to specify a particular service node to supply the resources.
  - Service nodes routinely multicast the names of their services and related information in service-announcement messages.
  - Servers automatically configure logical subnetworks by accepting service-announcement messages only from service nodes that share at least one group with the server.
  - Servers maintain current service and service-node directories containing this information.

  - Some LAT nodes can use a server name to get a server's Ethernet hardware address by multicasting a specialized message to solicit that information.

- LAT communications use logical communications channels between two LAT communications partners (virtual circuits). Any potential communications partners can create a virtual circuit to carry one or more sessions. Once initiated, a virtual circuit is controlled primarily by one of the virtual-circuit partners, so that no central controlling node is required.

- The LAT architecture ensures predictable data flow by defining the following features:

  - Special timers that regulate multicasts of service-announcement information and exchanges of session information

  - Maximum message sizes

- When simultaneous sessions occur between a server and the services of a particular service node, the LAT protocol ensures efficient data flow. A single virtual circuit carries all of those sessions, and a node transmits all of its accumulated session information for those sessions simultaneously.

3.2 Overview of the Architectural Layers of LAT Networks

LAT nodes implement three LAT architectural layers and two DNA architectural layers. The LAT layers comprise the Service Class, Slot, and Virtual Circuit layers. The DNA layers used by LAT comprise the Data Link and Physical Link layers. These DNA layers are part of the general model specified by the Ethernet specifications.

The LAT layers are (from upper to lower layers):

- Service Class layer: This layer provides a range of tasks relating to offering and requesting computing resources, such as service names, service-announcement messages, directory functions, and logical groups.

- Slot layer: This layer provides session control. **Session control** involves establishing and managing sessions and controlling the flow of information over them.

- Virtual Circuit layer: This layer provides the LAT transport mechanism. The **LAT transport mechanism** consists of virtual circuits between two LAT partners. Virtual-circuit partners multiplex their sessions over their shared virtual circuit.

To transmit LAT messages, the LAT architecture uses the Ethernet protocol defined by Digital Network Architecture (DNA). On DECnet systems, the same Ethernet protocol implementation transmits messages for both DECnet upper layers and LAT layers. The Ethernet protocol can target messages to a specific Ethernet hardware address or multicast messages. **Multicasting** is the process by which a message is addressed and sent to a set of logically related nodes, which, for LAT networks, comprises only servers. Note that a server discards any message that does not concern it.

The Ethernet protocol reflects the following two DNA layers (from upper layer to lower layer):

- Data Link layer: This general DNA architectural layer implements a communications facility on top of the medium provided by the underlying Physical Link layer. The Data Link layer places outgoing messages into discrete data units called **datagrams**, and it removes incoming messages from datagrams. Datagrams have a standard format that is determined by the Data Link protocol.
- Physical Link layer: This general DNA architectural layer provides a specific medium for transmitting the datagrams. The Physical Link insulates the Data Link layer from medium-dependent physical characteristics of the Ethernet network.

These five LAT and DNA architectural layers permit peer-to-peer communications between terminal servers and service nodes. For each type of LAT node, Figure 3-1 shows the LAT layers and the underlying DNA layers. In this figure, the term HOST refers to LAT service nodes; the terms USERS and SERVICES represent the Service Class layer.

Figure 3-1: LAT Layers and the Underlying DNA Layers

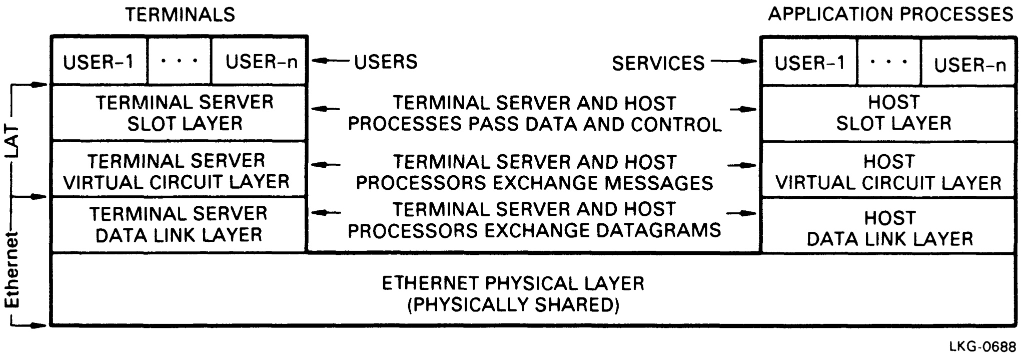
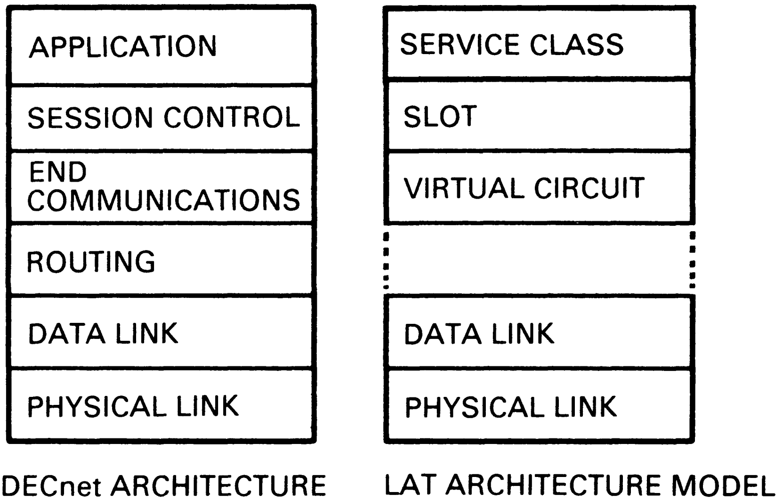

TERMINALS
APPLICATION PROCESSES
USER-1
. . .
USER-n
USERS
SERVICES
USER-1
. . .
USER-n
TERMINAL SERVER SLOT LAYER
TERMINAL SERVER AND HOST PROCESSES PASS DATA AND CONTROL
HOST SLOT LAYER
TERMINAL SERVER VIRTUAL CIRCUIT LAYER
TERMINAL SERVER AND HOST PROCESSORS EXCHANGE MESSAGES
HOST VIRTUAL CIRCUIT LAYER
TERMINAL SERVER DATA LINK LAYER
TERMINAL SERVER AND HOST PROCESSORS EXCHANGE DATAGRAMS
HOST DATA LINK LAYER
ETHERNET PHYSICAL LAYER (PHYSICALLY SHARED)
LAT
Ethernet
LKG-0688

Figure 3-2 compares the DNA architectural model and the LAT architectural model. Note that the DNA architectural model has a Routing layer that is unnecessary for LAT communications.

Figure 3-2: Comparison of DECnet and LAT Architectures



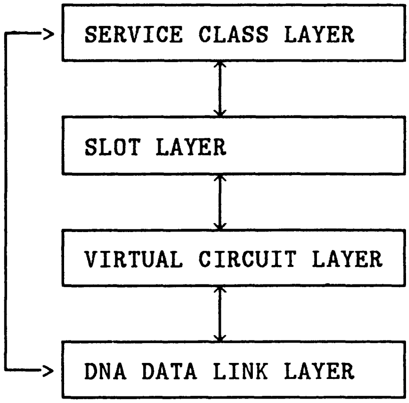
LKG-1200-87

Within a partner, the LAT layers and the underlying DNA layers are interdependent. Each layer uses functions provided by other layers and, in turn, provides functions to other layers. The interdependencies between layers permits passing information (such as connection requests, user data, or status information) upwards and downwards between layers. Figure 3-3 illustrates the LAT interlayer interfaces as bidirectional arrows. Each LAT layer interfaces with two other layers to perform the following functions:

- Interfacing of the Service Class layer with the Slot layer allows users to establish and manage sessions, exchange information with services, and disconnect sessions.
- Interfacing of the Service Class layer with the Data Link layer allows the Service Class layer to send and receive multicast messages.

- Interfacing of the Slot layer with the Virtual Circuit layer allows the Virtual Circuit layer to multiplex outgoing session information being sent to a particular LAT node and to demultiplex incoming session information received from a particular LAT node.
- Interfacing of the Virtual Circuit layer and the Data Link layer allows the Data Link layer to place outgoing virtual-circuit messages into datagrams and to pass upwards the incoming virtual-circuit messages that it removes from datagrams.

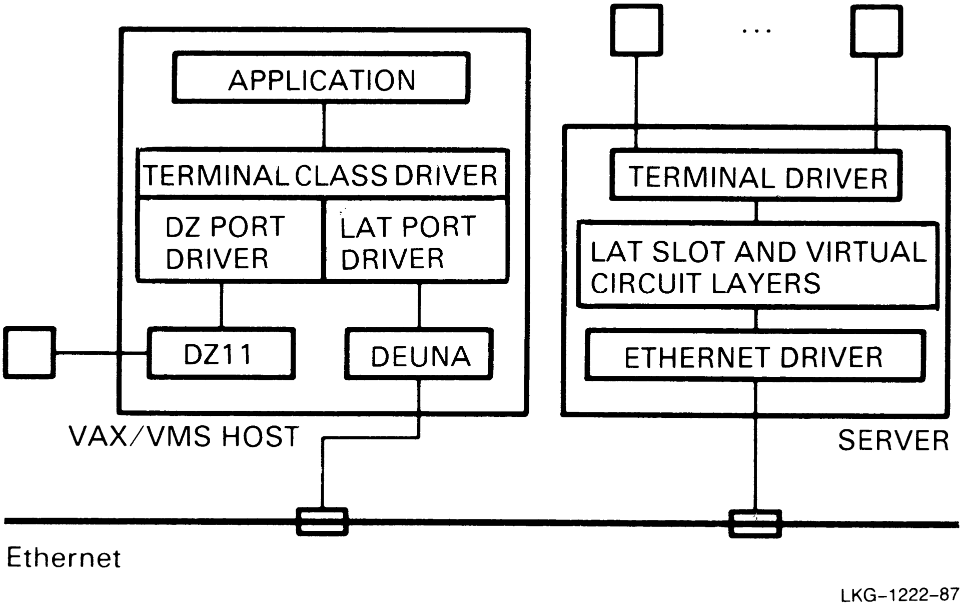
Figure 3-3: Interlayer Interfaces of LAT Architecture



Outgoing information is packed into messages; the messages are then placed into datagrams, which are transmitted over the Ethernet. When a datagram is received, its data-link control information is separated from the message within it; the message is then passed to an upper layer, which processes the LAT information as dictated by the LAT architecture.

Figure 3-4 shows the LAT protocol implementation on a VAX/VMS service node and on a terminal server on an Ethernet LAN. Note that on general-purpose service nodes such as VAX/VMS systems, the LAT service-node software is intricately meshed with the operating system software.

Figure 3-4: LAT Protocol Implementations



3.3 Service Class Layer

The Service Class layer is the highest layer of the LAT protocol. Currently the Service Class layer comprises a single module called the Service Class 1 module, which is the service class for interactive terminals. The Service Class 1 module is used by both terminal servers and service nodes for all their service-class functions. Digital has reserved all other modules for its future use.

3.3.1 Service Class 1 Functions

The Service Class 1 module provides the following functions:

- Offering services
- Assigning service ratings
- Identifying the groups that logically partition a LAT network
- Sending service announcements and building service and service-node directories using information from these announcements
- On service nodes, maintaining a multicast timer that determines when service-announcement messages are multicasted by a service node
- Providing flow control for sessions
- Queuing connection requests sent to servers under certain circumstances

3.3.2 Service Class 1 Messages

The Service Class 1 module uses the interface provided by the Slot layer to transmit and to receive user data. However, the Service Class 1 module also uses specialized messages. Service-class messages are processed directly by the Data Link layer. No virtual circuit is involved. The service-class messages are as follows:

- Service-announcement message
- Solicit-information message
- Response-information message
- Command message
- Status message

3.3.2.1 Service-Announcement Messages

The Service-announcement message is a multicast message sent exclusively by service nodes. It contains information about the sending service node and its services. This information includes:

- Node name and identification string
- Service-node groups
- Service names and identification strings
- Service ratings

The service names, groups, and service ratings are used by the Slot layer to select a service node if none is specified in the connection request. The Slot layer provides the node name and Ethernet address of the selected service node to the Virtual Circuit layer.

3.3.2.2 Solicit-Information and Response-Information Messages

Under some conditions, a server uses its connection queue to hold connection requests for unavailable ports (and services). The solicit-information message and the response-information message are complementary messages that allow a LAT node to access a server's connection queue. Note that a dual-purpose server queues connection requests only for services that are enabled for queuing.

- **Solicit-information message**

  Accessing the connection queue depends on a requesting node using a specialized message called a **solicit-information message**, which allows the requesting node to identify a target server. A **solicit-information message** is a multicast message that designates a specific server and requests the server's Ethernet address.

  Solicit-information messages are sent by only some service nodes and servers. Service nodes with host-initiated request capability use solicit-information messages to locate a specified target server. All host-initiated requests are potentially eligible for queuing. A service node can, however, decline queuing for any of its host-initiated requests. Some servers send solicit-information messages. These messages solicit the information required to access connection queues on behalf of user-initiated requests from the server's users. The target server must be a dual-purpose server.

- **Response-information message**

  The server that is designated in a solicit-information message responds using a **response-information message**. The response-information message is sent to the Ethernet hardware address of the service node. This message contains the server's address and other identifying information that is required for queuing a connection request. The information in the response-information message is always required for a service node to make host-initiated requests, whether the requests are eligible for queuing or not.

3.3.2.3 Command and Status Messages

The Service Class 1 module defines a second pair of complementary messages called the command message and the status message. This pair of messages are used for making host-initiated requests and queued user-initiated requests and for managing the connection queue.

Every host-initiated request and any user-initiated request that might be queued receives a specific request number from the requesting node. A **request number** is a number that allows the communications partners to keep track of a connection request during a series of command and status messages.

- **Command messages**

  These messages are sent by a node requesting the connection. A command message contains the physical address of the targeted terminal server. A command message either makes a host-initiated request or manages a queued connection request.

- **Status messages**

  These messages are sent by a node that receives a queuable connection request. A status message contains the physical address of the targeted node. Status messages either respond to command messages or routinely report the status of a queued connection request.

For Host-Initiated Requests

The functions of command and status messages relating to making host-initiated requests are as follows:

- A command message carries a host-initiated request to a server whose address was just received by the requesting service node in a response-information message.
- A status message indicates whether a server has accepted, rejected, or queued a host-initiated request.

For Managing Queued Connection Requests

The queue-related functions of command and status messages are as follows:

- Command messages can remove a queued entry or inquire about the service status and the position of a request in the queue.
- Status messages inform a node of the status of its queued connection requests.

3.3.3 Service-Class Flow Control

The Service Class 1 module handles flow control for outgoing user data being accumulated for a session. This overview of service-class flow control discusses terminal servers only, since service-class flow control on a general-purpose service node depends on the operating system.

Terminal servers use flow-control signals to control the flow of user data between the user device and the server port on a session-by-session basis. Port flow control prevents a transmit buffer from overflowing and prevents a receive buffer from being lost, as, for example, when a server user switches out of a session.

All servers support XON/XOFF in-band flow control. As a transmit buffer, which holds outgoing user data for a given session, nears its capacity, the Service Class 1 module of a server sends an XOFF signal to the device attached to the server port of the session. After successful transmission of the data from the transmit buffer, the module sends XON to the attached device. On modem-controlled ports DSR/DTR modem signals and, sometimes, RTS/CTS modem signals can provide out-of-band flow control as an alternative to XON /XOFF flow control.

3.4 Slot Layer

The Slot layer is the middle layer of the LAT architecture. This layer resides between the Virtual Circuit layer and the Service Class layer. The Slot layer establishes and maintains sessions for the Service Class layer. The Slot layer uses the Virtual Circuit layer to send and to receive session information over virtual circuits.

3.4.1 Slot Layer Functions

The Slot layer provides the following functions:

- Establishing sessions between server ports and service nodes
- Providing two full-duplex flow-controlled data channels and one nonflow-controlled data channel for each session
- Providing flow control for session data
- Terminating sessions when requested to by the Service Class layer

3.4.2 Types of Message Slots

Information for a given session is stored and transmitted within one or more slots. A **slot** is a message segment containing information corresponding to a single session. To perform its communications function, the Slot layer generates start, reject, data, and stop slots.

3.4.2.1 Start Slot

The Slot layer of a server requests a session with a service node by producing a start slot for the session. The Slot layer of a service node either accepts the connection request by responding with another start slot or rejects it by responding with a reject slot. If the service node accepts the start slot, the two nodes become session partners for the duration of the session.

3.4.2.2 Reject Slot

This slot is used to reject a start slot. Only a service node can transmit a reject slot.

3.4.2.3 Data Slots

By accepting a start slot, a service node completes session establishment. While a session lasts, session partners exchange user data using data slots. The available types of data slots are:

- **Data-A slot:** This slot carries user data.
- **Data-B slot:** This slot carries port and session characteristics and status indicators, such as break, parity, and framing errors
- **Attention slot:** This slot carries out-of-band data that signals special conditions such as the termination of output to a terminal.

Each slot is usually processed in the order in which it appears. However, the processing priority can vary. An attention slot takes priority over data-A and data-B slots so that the out-of-band data of an attention slot is processed as soon as it is received. For example, when a service node receives a CTRL/O character, the service node sometimes sends an attention slot to the server, causing the server to abort further output.

3.4.2.4 Stop slot

Data exchange can continue until one of the session partners terminates the session by sending a stop slot. Normally, a stop slot occurs after a user logs out of a service node or types a DISCONNECT command at the terminal. If the session is the last session on a virtual circuit, the virtual circuit is also terminated.

3.4.3 Slot Flow Control

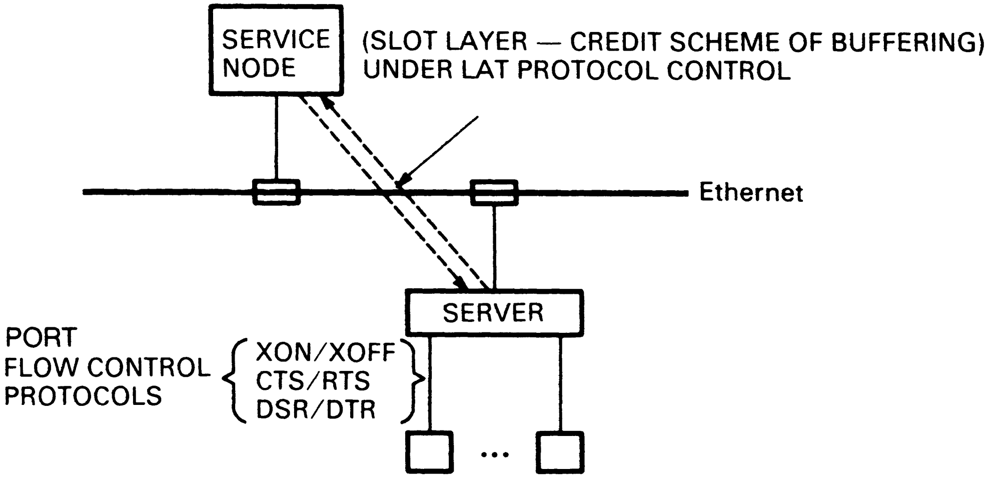
The Slot layer temporarily stores incoming data for each session in receive buffers. If a session is active, its receive buffers are filled and emptied regularly. However, if the session becomes inactive (as when a server user switches out of the session), its receive buffers might remain full for an extended time. Slot flow control ensures that the receive buffers get no new data until the Slot layer finishes processing the buffer's existing data.

The slot flow-control mechanism is based on a credit-exchange scheme. When establishing a session, two LAT nodes negotiate the number of receive buffers (usually two) that each node has for the session. For each receive buffer, each node also provides its partner with a credit. A **credit** is a marker that allows the node that owns it to send one slot (up to 255 bytes) of user data to the other node. When sending a data-A or data-B slot, a node must include a credit. The credit represents an available receive buffer on the other node.

Credit exchanges ensure that a receive buffer is available for every data-A or data-B slot that a node receives from a session partner.

Figure 3-5 illustrates the distinction between the internode credit-based flow control of the Slot layer and intranode flow control of the Service Class at server ports. The set of dashed lines between the terminal server and the service node represents the movement of credits back and forth between the nodes.

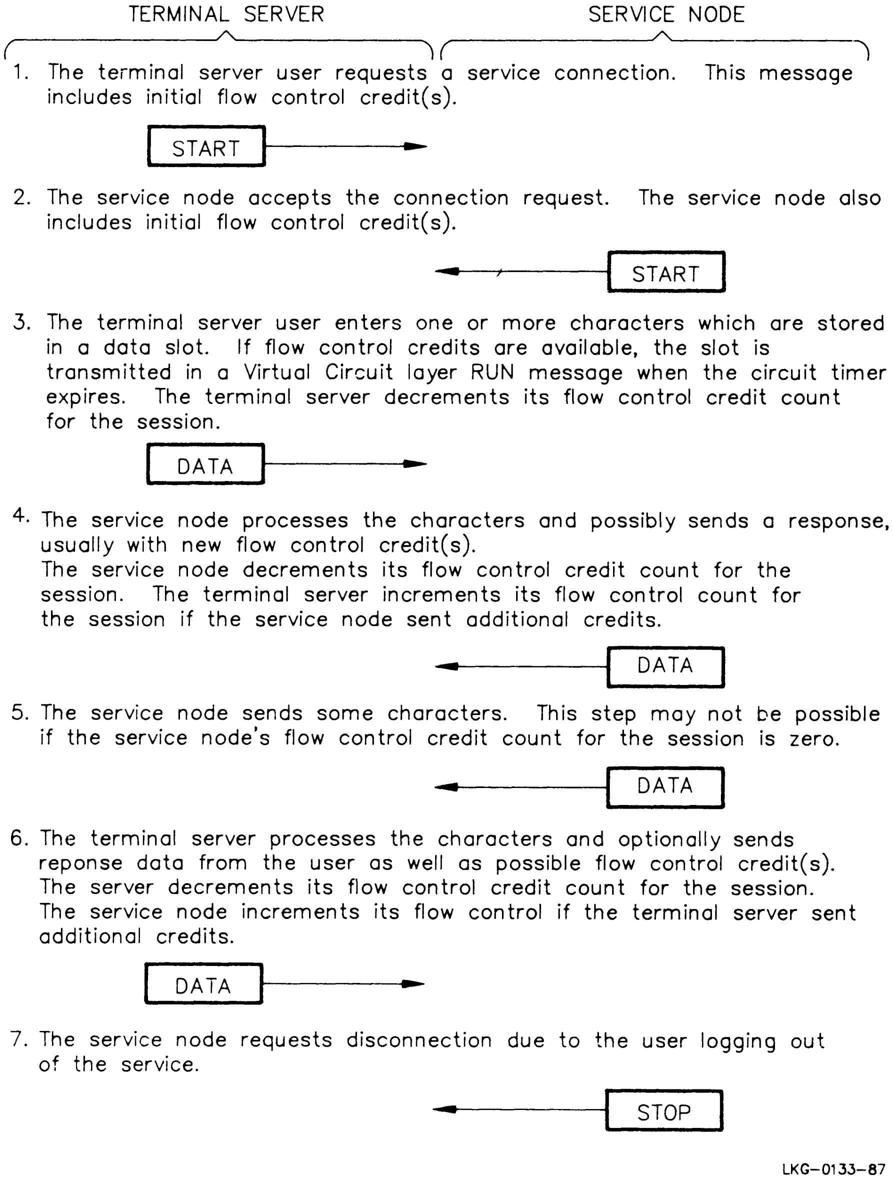
Figure 3-5: Slot Flow Control on a LAT Network


SERVICE NODE
(SLOT LAYER — CREDIT SCHEME OF BUFFERING) UNDER LAT PROTOCOL CONTROL
Ethernet
SERVER
PORT FLOW CONTROL PROTOCOLS
XON/XOFF
CTS/RTS
DSR/DTR
...

LKG-1183-87

Each session partner maintains a flow-control credit count. When a partner's credit count is greater than zero, the partner can send data. After expending its last credit, a partner decreases (decrements) its credit count to zero. A zero credit count causes the partner to wait to send more data until one or more of its credits are returned by its session partner. Whenever a partner processes the data in one of its receive buffers, it always returns the credit. This allows the other partner to increase (increment) its flow-control credit count. The operation of slot flow control is summarized in Figure 3-6.

Figure 3-6: Operation of Slot Flow Control



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

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

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.

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.

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.

6. The terminal server processes the characters and optionally sends reponse 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.

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

3.5 Virtual Circuit Layer

The **Virtual Circuit layer** is the lowest layer of the LAT architecture. This layer establishes and maintains virtual circuits. A **virtual circuit** is an independent logical communications channel between a terminal server and a service node. LAT virtual circuits are multiplexed over a single physical circuit. The physical circuit hardware includes the network interface of the terminal server, the network hardware, and the network interface of the service node.

A terminal server, which always initiates virtual circuits, can initiate simultaneous virtual circuits to different service nodes. During its life span, a single virtual circuit supports all sessions between a server's ports and the service(s) of a single service node. A virtual circuit lasts as long as any session remains.

The Virtual Circuit layer provides an interface to the Slot layer for the transmission and reception of data and session-control information. The Virtual Circuit layer is responsible for transporting messages between two nodes and guaranteeing delivery of error-free, correctly sequenced data.

The Virtual Circuit layer uses the interface provided by the DNA Data Link layer to send and receive virtual-circuit messages. The Data Link layer packages the virtual-circuit messages into datagrams.

3.5.1 Virtual Circuit Layer Functions

The Virtual Circuit layer performs the following functions:

- On servers, initiating a virtual circuit whenever the server's Slot layer requests access to a service node without a virtual circuit
- On servers, maintaining a circuit timer that determines when the server sends its virtual-circuit messages
- Transmitting slots and credits for the Slot layer
- Multiplexing and demultiplexing slots over a single virtual circuit between a server and a service node
- Managing connections, sequencing message, detects errors, and corrects problems created by duplicated and lost messages
- Providing a keepalive function that maintains a virtual circuit if no user data or credits are sent by the Slot layer for a period of time
- Stopping a virtual circuit when all sessions cease

3.5.2 Virtual Circuit Layer Messages

Virtual-circuit messages are messages used by LAT nodes to communicate information about virtual circuits or sessions. Both servers and service node compose virtual-circuit messages. A virtual-circuit message is always addressed to a specific LAT node. The addressed node is always the opposite of the sending node: servers send virtual-circuit messages only to service nodes, and service nodes send them only to servers. A LAT node receives the virtual-circuit messages addressed to it and its Virtual Circuit layer processes them to extract the information that they contain. The maximum size of a virtual circuit message is constrained by the LAT protocol, which is itself constrained by the Ethernet Data Link protocol.

Virtual-circuit messages comprise three message types, which are named for the type of function they perform. The types of messages are start, virtual-circuit, and stop messages.

3.5.2.1 Start Messages

A **start message** is a virtual-circuit message that starts a virtual circuit. Start messages are exchanged between a server and a service node to start a virtual circuit. The Virtual Circuit layer of a server initiates the exchange in response to a request from its Slot layer. The terminal server sends a start message to a specific service node. The service node acknowledges the start message with a corresponding start message that accepts the start request or with a stop message that rejects the start message.

Start messages contain information needed by the LAT nodes to start the virtual circuit. The information includes the data-link frame size for the LAT node, which determines the size of virtual-circuit messages. The information also includes the size of slots, the size of transmit and receive buffers, and the number of receive buffers and credits available for each session.

3.5.2.2 Run Messages

Virtual-circuit partners routinely exchange run messages A **run message** is a virtual-circuit message that contains session information. The run message establishes, maintains, and disconnects sessions. Run messages sent by the server usually require an acknowledgment from the service node. Virtual-circuit partners continue to exchange run messages until one or the other sends a stop message.

Run messages carry the following types of information for each session:

- Start session requests
- Credits for Slot layer flow control
- User data
- Changes in how a session is handling data transparency and changes in port characteristics
- Port status (such as parity errors)
- Stop session requests

Slots are passed by the Slot layer to the Virtual Circuit layer, which transmits them for the Slot layer in a virtual-circuit run message. All sessions on the same virtual circuit share the same run message. The different types of message slots can be intermingled.

In certain situations, run messages contain no slots. Such empty run messages maintain a virtual circuit when all of its sessions are inactive for a specified period of time.

3.5.2.3 Stop Messages

A **stop message** is a virtual-circuit message that ends a virtual circuit. Either virtual-circuit partner can terminate a virtual circuit by sending a stop message. However, the server usually sends the stop message after the last session on a virtual circuit is disconnected. Stop messages are unacknowledged.

3.5.3 Virtual-Circuit Flow Control

Virtual circuit flow control is simple. A node cannot transmit its next run message until its previous run message has been acknowledged. If the previous run message is not acknowledged, the server retransmits the message until either the service node acknowledges the message or the server passes its retransmit limit.

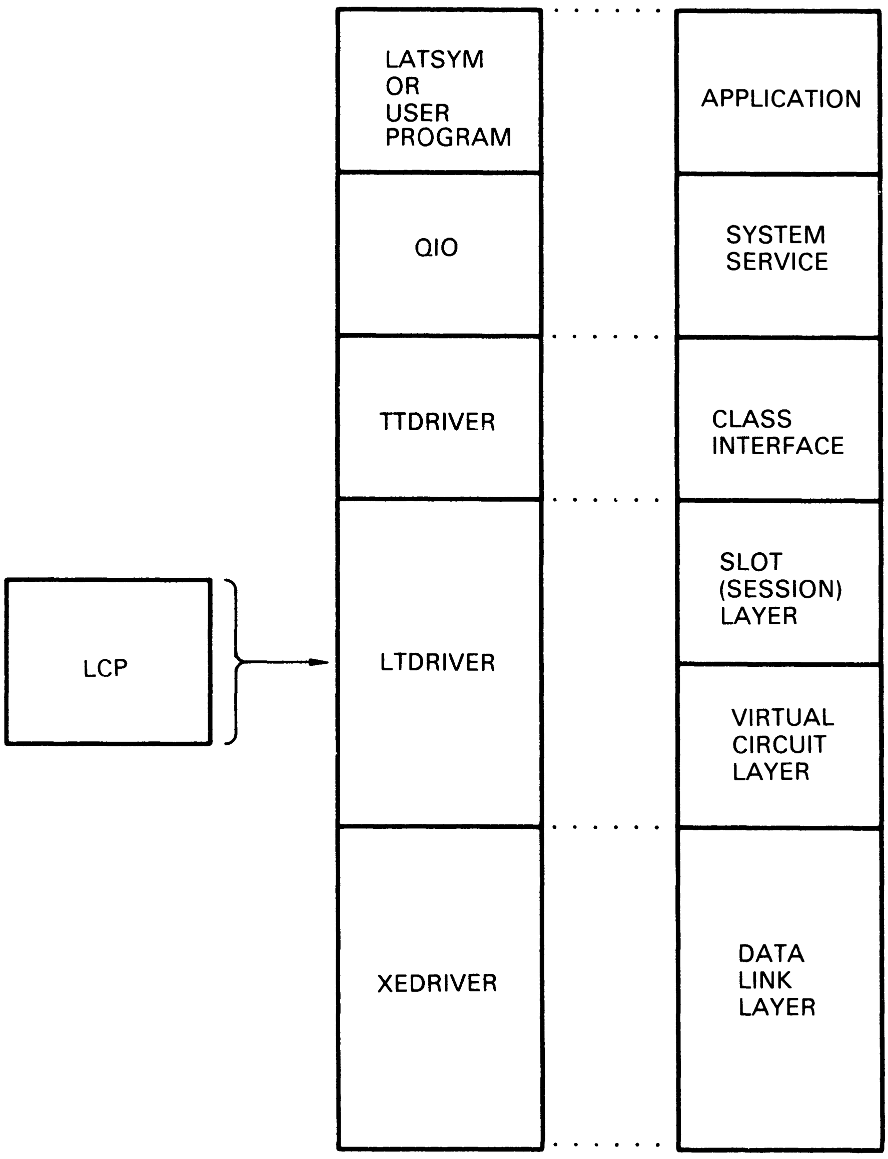
3.6 A VAX/VMS Implementation of LAT Layers

The VMS implementation of service-node software integrates LAT protocol and management features with VMS device drivers and queues. Figure 3-7 illustrates the relationship between LATplus/VMS layers and VAX/VMS device drivers. The components of LATplus are explained in the following table:

Table 3-1: Components of LATplus/VMS

Component	Explanation
LATSYM	An image that directs output from a print queue to a printer on a terminal server. The symbiont transfers data disk to the printer.
QIO	A VMS system service that prepares an I/O request for processing by the driver and performs device-independent preprocessing of the request.
TTDRIVER	A VMS driver that provides a common interface to different terminal I/O devices.
LTDRIVER	A driver that implements the LAT protocol and which interacts with the LCP and TTDRIVER to prepare LAT data for the XEDRIVER.
LCP	The LAT Control Program, which provides the command interface for controlling LTDRIVER and for displaying information about LTDRIVER (its counters, characteristics, and so forth).
XEDRIVER	A VMS-software device driver that controls the DEUNA and DELUA UNIBUS-to-Ethernet network adapters. XEDRIVER transmits and receives messages for higher-level drivers such as LTDRIVER.
XQDRIVER	A VMS-software device driver that controls a DEQNA or DELQA QBUS-to-Ethernet network adapters. The XQDRIVER performs similar functions as the XEDRIVER.

Figure 3-7: A VMS Implementation of LAT Network Layers



LKG-1211-87

4

LAT Communications Processes

Drawing on the preceding discussion of LAT architecture, this chapter summarizes LAT communications processes. The basic LAT communications processes are as follows:

- The service-announcement process

  Service nodes offer named services and announces them over the LAN. Servers keep directories of announced services and of the service nodes that announce them.

- Session establishment

  A terminal server is always the initiator of this process. Servers respond to a connection request by selecting a service node for the session, by ensuring that there is a virtual circuit, and by initiating a session with the requested service. The service node cooperates in formation of virtual circuits and sessions.

- Session management

  Both servers and service nodes buffer user data, provide flow control, and disconnect sessions as needed. When multiple nodes offer a common service, servers also provide automatic failover for sessions to the service if they are disrupted by loss of their virtual circuit.

- Virtual-circuit maintenance

  This process keeps virtual circuits alive when their sessions are all inactive and retransmits unacknowledged run messages.

- Host-initiated-request process

  This process begins when an applications program on a general-purpose service node requests an applications device. The service node multicasts a request for the address of the server offering the applications device and then makes the host-initiated request directly to that server. Servers participate in the host-initiated request process by supplying their address and by processing host-initiated requests received from service nodes.

- Connection-queue maintenance

  A server maintains its connection queue by sending intermittent status messages for each queued request to the node that requested it. Before it is completed, a queued connection request can be removed from the queue by a command from the requesting node or from the server manager. Otherwise, when a port becomes available for a queued request, the server removes the request from the connection queue and enters the session-establishment process.

This chapter discusses the dynamics of each of these processes.

4.1 The Service-Announcement Process

The service-node software offers named services and announces them over the LAN. The service-announcement process begins with calculation of service ratings and multicasting of a service-announcement message by a service node. It ends with the building of service and service-node directories by servers.

4.1.1 Maintaining Service Ratings

For each service, service nodes generate a service rating value from 0 to 255. A **service rating** is a value that indicates the relative availability of a service when compared against other service ratings. Service nodes dynamically calculate the service rating for each of their services. For a general-purpose service node, the service rating is based on the overall level of activity of the node, the amount of memory, and the processor type. On some service nodes, system managers can define a static value for a service rating. For services offered by a dual-purpose server, the service rating reflects the number of ports that offer the service and are currently available. Also, in some servers, the service ratings of services with queuing enabled reflect the amount of space in the server's connection queue.

4.1.2 Multicasting Service-Announcement Messages

A service node periodically multicasts a service-announcement message to all servers. This message contains information about the node and its services. All services on a service node and their current service ratings are advertised in every multicast message sent by a service node. All servers receive this multicast message, and those that share a group with the service node accept the message and store its information. This process allows the servers to identify service nodes that offer services requested by server users.

On most general-purpose service nodes, the announcement of services begins as soon as the LAT driver starts. A service node sends multicast messages at a regular interval that is determined by a multicast timer. The **multicast timer** is the timer that controls the interval, in seconds, between multicast messages. The network or system manager can control the frequency of multicast messages by changing the value assigned to the multicast timer. On some service nodes, if the system manager creates any new service while the driver is running, the service node multicasts a service-announcement message immediately.

4.1.3 Maintaining Service and Service-Node Directories

Terminal servers receive service-announcement messages sent by service nodes. When a service-announcement message arrives, the server compares the service-node groups listed in the message with the server's current port groups. If there are one or more common groups, the server processes the service-announcement message.

Using information received in the message, the server builds two directories in its memory: a service directory and a service-node directory. For each service, a server's service directory contains the service's name and identification strings, the node names of service nodes that offer each service, its current status, and the service rating of the service on each service node. The service-node directory contains the node name, identification, and address of each known service node, as well as its current status.

These directories are used to process connection requests during the session-establishment process. This process is discussed in the following section.

NOTE

Service nodes do not maintain service directories, which is one of the reasons that connections to LAT services are possible only for interactive devices attached to terminal servers.

4.2 The Session-Establishment Process

Establishing a session occurs after a terminal server receives a connection request, which can be a host-initiated request or a user-initiated request.

Establishing a session involves a multistage process called the **session-establishment process**. Failure at any stage ends the session-establishment process. The stages, which always occur in the same order, consist of the following:

1. Selecting a service node

   This stage can involve translating the requested service name into a node name, selecting a service node, checking whether it shares a group with the server, and getting its address.

2. Ensuring that a virtual circuit is available

   If no virtual circuit between the server and the selected service node exists, then the server tries to start one.

3. Starting a session

   When a virtual circuit is available, the server tries to start a session.

4.2.1 Selecting a Suitable Service Node

Before initiating a session, a server must select a service node that is reachable and that shares at least one common group with the server. The possible steps for selecting a service node are identifying service nodes, comparing service ratings, checking groups, and getting a service-node address.

Since some connection requests supply some of the preceding information, the number of steps actually required to identify a suitable service node vary according to the type of connection request. There are several possible situations:

- **A user-initiated request that specifies only a service name**

  The server must complete all the steps listed below to locate a node.

- **A user-initiated request that specifies a service node**

  The server immediately checks service-node groups of designated service node (step 3, below) and then gets the node's address (step 4, below).

- **A host-initiated request**

  The command message sent by the service node supplies the server with the service node's node name and address so that only checking groups (step 3) is necessary.

The possible steps for identifying a service node involve the following events:

1. Identifying service nodes that offer a service

   On receiving a user-initiated request that specifies only a service name, the server searches its service directory for the requested service name. If the service name is absent from the service directory, the server rejects the connection request itself. If the service name is present, the server finds the names of the service node(s) offering the service (if any). The server then performs step 2.

2. Comparing service ratings

   If multiple service nodes offer the service, the server performs load balancing. **Load balancing** is a process that compares the the activity levels of two or more service nodes that offer a requested service to select the least busy service node. A high level of system activity gives a low service rating and inhibits new connections. After comparing a service's service rating on the service nodes that offer the service, a server selects the service node with the highest (that is, the most favorable) service rating. Server users requesting the common service name benefit from load balancing, which places interactive users in the best possible computing environment.

   Figure 4-1 illustrates how the members of a VAXcluster can each have an individual service name and a common service name. The figure shows three service-node-specific services called A, B, and C, and a common service called COMMON.

3. Checking groups

   The server checks the groups of the service node. The service nodes must share at least one group with the server port for which a session has been requested. If the service node lacks any common group with the server port, the server rejects the connection request.

4. Getting the service node's address

For user-initiated requests, the server gets the address of the selected service node. At this point, the service node ensures that there is a virtual circuit with the service node (see Section 4.2.2). For host-initiated requests, the address of the service node is supplied in advance, so the server proceeds directly to ensure that there is a virtual circuit.

4.2.2 Ensuring that a Virtual Circuit Exists

The next step of the session-establishment process is ensuring that there is a virtual circuit between the server and the service node. In establishing a session, a server determines whether a virtual circuit already exists with the targeted service node. In a virtual circuit does not exist, the server creates one before initiating the session.

To create a virtual circuit, the server sends a virtual-circuit start message to the selected service node. Generally, conditions permit a service node to accept the request by responding with an equivalent virtual-circuit start message. However, some conditions cause a service node to reject virtual-circuit start messages from a terminal server. The likely reasons are (in order of their probability):

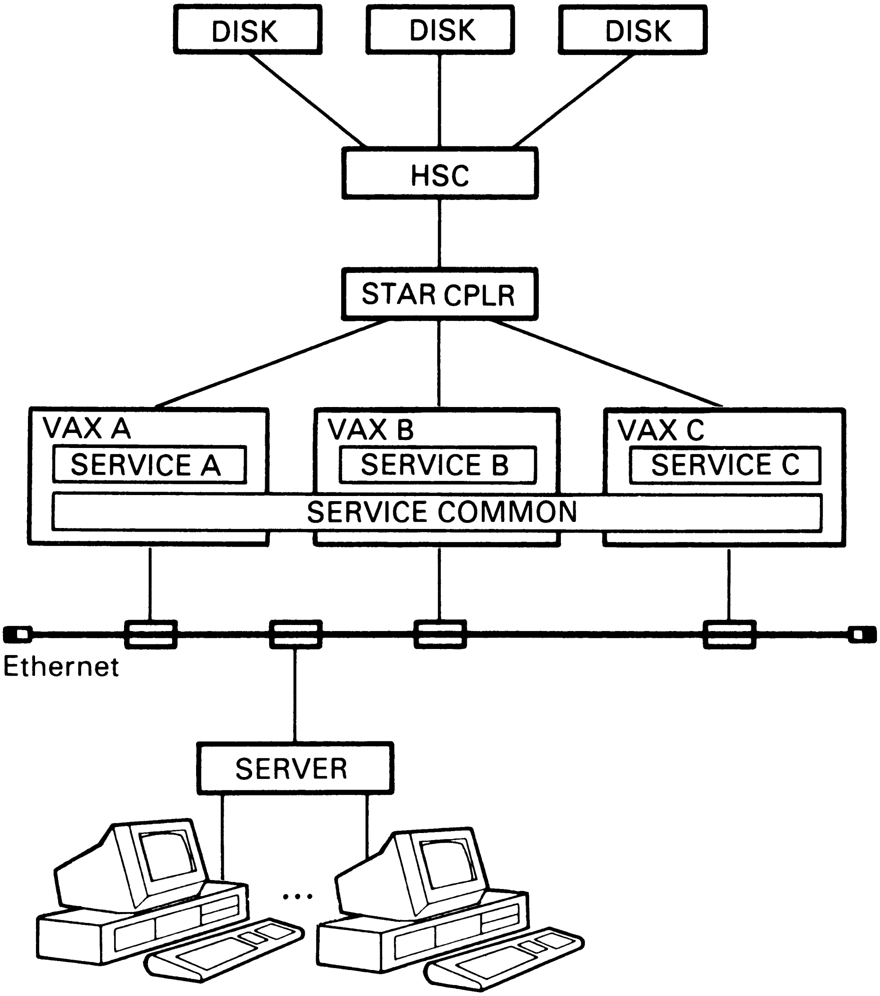
1. The service node has insufficient resources to satisfy the request.
2. The maximum number of virtual circuits is exceeded.
3. An illegal or invalid message is received.

4.2.2.1 Exchanging Session Information over a Virtual Circuit

After being established, a virtual circuit enters a run state during which it can pass run messages. At this point, the Slot layer of a server initiates sessions and begins exchanging data and session-control information. When preparing to send a run message, the Virtual Circuit layer usually packs whatever slots are pending for a virtual-circuit partner into a single run message. However, if the number of slots exceeds the maximum number permitted for a run message, excess data is held for the next run message. In this case, the sessions with unread user data go to the top of a round-robin queue so that they are processed before other sessions for the next run message.

The transmission of run messages is not governed by the amount of data that is pending for each virtual circuit. Instead, the server initiates an exchange of virtual-circuit run messages at regular intervals. To determine this interval, the server maintains a circuit timer. The **circuit timer** is a timer that determines the interval at which a server transmits run messages containing user data and/or

Figure 4-1: VAXcluster with a Common Service Name



LKG-1207-87

credits. A server hold virtual-circuit messages until the circuit timer expires. If the previous run message has been acknowledged by the service node, the server can send a new run message. After sending any pending information, the server resets the circuit timer.

Implementation of the circuit timer is server-specific. The DECserver 100 and 200 systems create a separate circuit timer for each virtual circuit and maintain it only until that circuit stops. The Ethernet Terminal Server and DECserver 500 maintain one circuit timer for all virtual circuits, and they restart it each time a server unit is loaded.

Digital provides a default value of 80 milliseconds as the interval between virtual-circuit messages. A transmit buffer seldom gets filled in that interval, and,generally, the default value is optimal. However, it is possible for server and LAT network managers to change the interval. For information on setting circuit-timer values see Section 5.14.2.

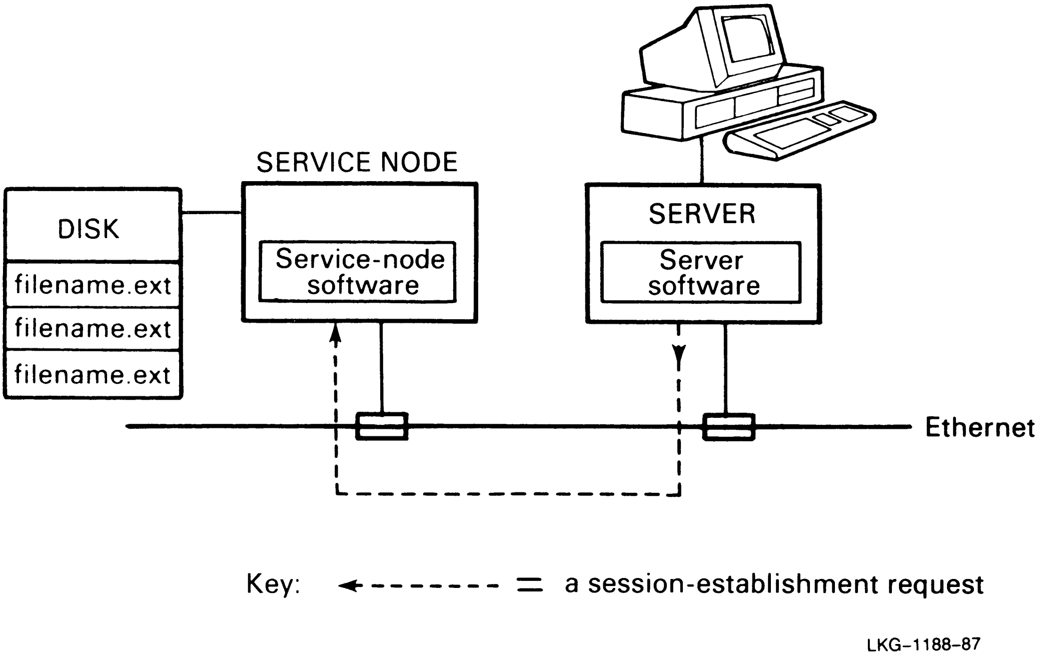
4.2.3 Starting Sessions

After ensuring that there is a virtual circuit to the selected service node, the server attempts to create a session. Creating a session is equivalent for user-initiated requests and host-initiated requests. All sessions are identical regardless of who requests them. Figure 4-2 shows a terminal server requesting a session with a service node. The dashed line represents a request on route from the terminal server to the service node.

4.2.3.1 Sending a Start Slot in a Run Message

The terminal server sends a start slot in a virtual-circuit run message to the selected service node. For a user-initiated request, the information in the start slot identifies a requested service by name and (if designated) a port identifier. For a host-initiated request, the start slot includes the request number that the service node specified in its command message to identify the particular host-initiated request.

Figure 4-2: Terminal Server Establishing a Session

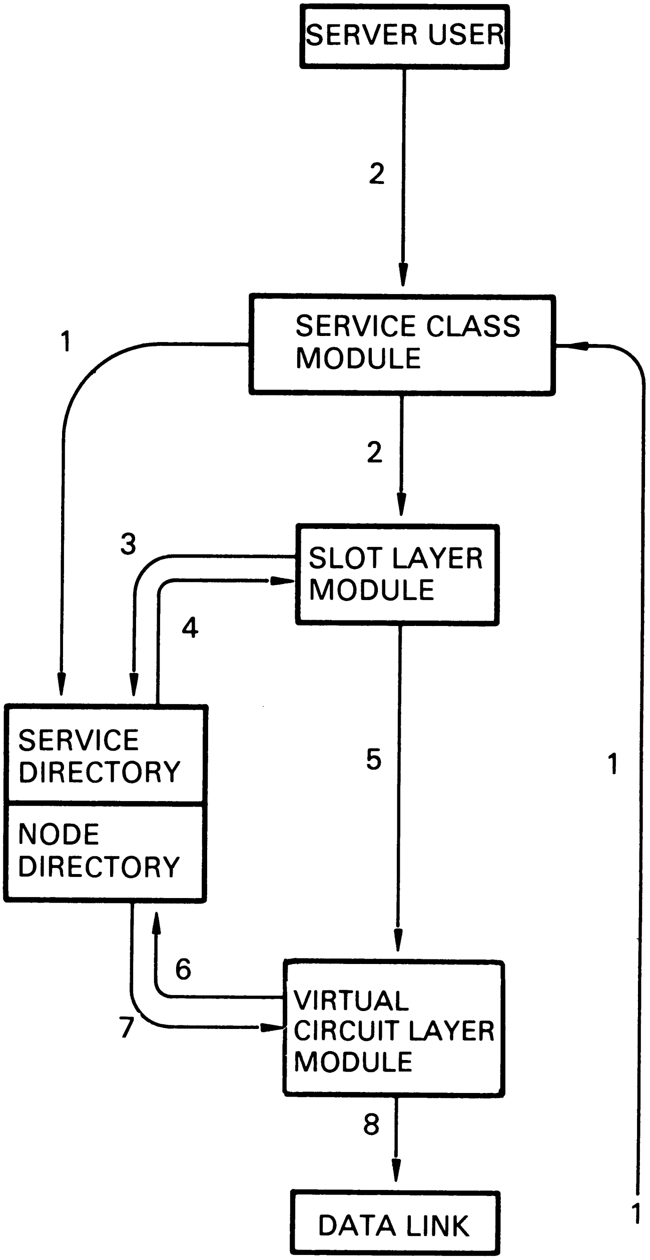


4.2.3.2 Responding to Start Slots

After receiving a virtual-circuit run message containing a start slot for a session, a service node processes the start slot by accepting or rejecting it. Generally, a service node responds favorably to a start slot by acknowledging it with an equivalent start slot in its next run message. This acknowledgment informs the server that the session has been started successfully. The main reason for a service node to reject a start slot is that the service node has insufficient resources for the requested session.

The session-establishment functions of a terminal server are shown in Figure 4-3.

Figure 4-3: Session-Establishment Functions of Servers



1. Service node multicast message received—directory updated.
2. User requests service connection.
3. Slot layer looks up service name.
4. Node offering service is found.
5. Slot layer requests Virtual Circuit layer to set up virtual circuit.
6. Virtual Circuit layer looks up node.
7. Node address is found.
8. Virtual Circuit layer transmits the START message.

LKG-1202-87

4.3 Session Management

This section considers several functions related to maintaining sessions. These functions include:

- Passing user data for sessions (performed by both servers and service nodes)
- Managing flow control for sessions (performed by both servers and service nodes)
- Providing automatic failover for interactive sessions (performed by servers only)
- Disconnecting sessions (performed by both servers and service nodes)

4.3.1 Passing User Data for Sessions

LAT nodes pass user data within virtual-circuit run messages. Passing user data for a session requires that a node have a credit for every slot. When building the next run message for the virtual circuit that carries the session, a node puts one credit and up to 255 bytes of user data within each data slot.

LAT nodes accumulate user data from every session in one or more session-specific buffers. Usually, a session uses one transmit buffer, where it stores user data to be transmitted, and two receive buffers, where it stores user data received from its virtual-circuit partner.

Each time the circuit timer expires, a server places buffered user data into a slot within a run message for the virtual circuit that carries the session. After building a run message, a node transmits the message over that circuit.

A server sends run messages containing user data when its circuit timer expires only. Service nodes respond to incoming run messages by building and transmitting a run message that contains any user data generated at the service node since the last run message.

If a server lacks user data or other information to send over a given virtual circuit, the server sends no run message when its circuit timer expires. Generally, this causes the virtual circuit to enter a balanced state. Section 4.4 discusses how LAT nodes manage a balanced virtual circuit.

4.3.2 Performing Automatic Failover

Automatic failover is a failure-recovery function of terminal servers that takes over if a session is disrupted because a virtual circuit fails. When such a failure occurs, the server automatically searches for other service nodes that offer the same service. The server attempts to connect to the service on the alternative node with the highest service rating. In the case of disruption of a session with a service, a server can attempt to reconnect a terminal to another service node offering the same service. Note that the server user must log in again on the new service node.

Automatic failover is useful with VAXclusters. By assigning a common service name to two or more cluster members, a system managers enables terminal servers to provide automatic failover among the members. Figure 4-1 illustrates how the members of a VAXcluster can each have an individual service name and can share a common service name. Server users requesting the common service name benefit from automatic failover.

4.3.3 Disconnecting Sessions

Interactive sessions can be disconnected from a server in various ways:

- Often, interactive sessions are disconnected at the service node by a server user logging out of the service node. The service node then disconnects the session with the user's server.
- A server user can disconnect an interactive session by using the server DISCONNECT command or logging out of the server. Also, a server manager can disconnect any session with any server port by logging out the port. In these cases, the server at which the disconnection occurs sends a stop slot to its virtual-circuit partner.
- When a server user disconnects from a non-LAT host or a non-LAT host crashes, the host drops its DTR or DSR modem signal. The resulting signal loss causes the dual-purpose server to disconnect the session with the server user.
- Generally, sessions end between a service node and an applications device on a server port when the service node has sent the last of the user data and received confirmation from the server. At that point, the service node disconnects the session by sending a stop slot to the server.

4.4 Virtual Circuit Maintenance

The Virtual Circuit layer is responsible for maintaining virtual circuits until the last session ends. Maintaining virtual circuits involves retransmitting unacknowledged run messages and keeping virtual circuits active when sessions are unused. When a virtual circuit fails, the Virtual Circuit layer can attempt automatic failover for sessions using services that are offered by multiple service nodes.

4.4.1 Retransmitting Messages

Retransmission of messages ensures reliable communications on the LAN. The retransmit limit specifies the number of times a run message is retransmitted without acknowledgment. If a server reaches this limit for a virtual circuit, it considers the circuit to have timed out and declares the service node to be unreachable. The server then times out all sessions on that virtual circuit. If another service node offers a service used by a timed-out session, the server attempts automatic failover.

4.4.2 Managing Balanced Virtual Circuits

When all sessions are unused on a virtual circuit, the circuit is said to be **balanced**. Normally, a virtual circuit enters a balanced state only after each node has emptied its receive buffers and returned the associated credits to the other node. Thus, while a virtual circuit is balanced, each node has one or more credits that allow it to transmit data, and either node can end the balanced state.

Servers are responsible for monitoring the service nodes when a virtual circuit is balanced. A keepalive timer is maintained by every server. Each time the server sends out a series of virtual-circuit messages, it resets its keepalive timer as well as its circuit timer. The **keepalive timer** is a timer that determines the amount of time that a balanced circuit remains inactive.

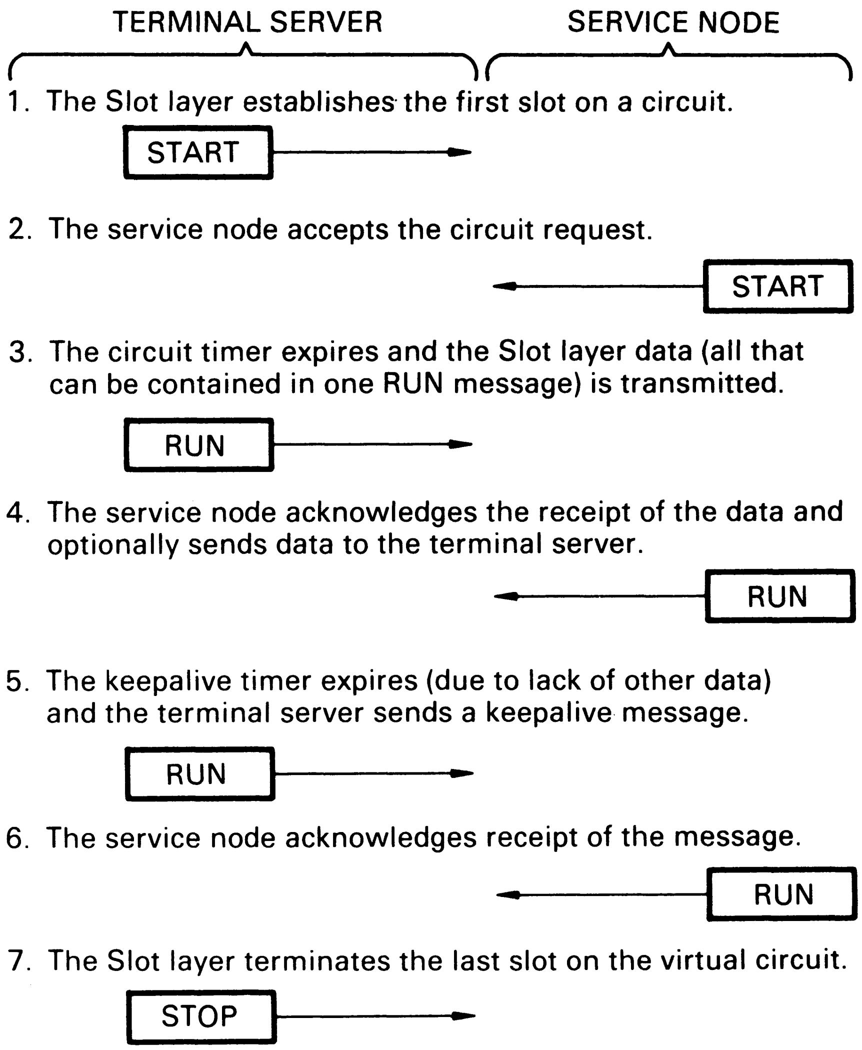
If the server has no information to transmit when the circuit timer expires, the server allows the keepalive timer to continue its current countdown. If the keepalive timer expires before any virtual-circuit message is sent by either node, the server sends an empty run message to monitor the status of a balanced virtual circuit. The server also informs a service node of its keepalive timer value within a virtual-circuit start message. If the service node receives no run messages for that amount of time, it can time out a virtual circuit.

An **empty run message** contains no slots. It is used to maintain a virtual circuit when no sessions are passing data or credits. An empty run message solicits acknowledgment from another node that shares a balanced virtual circuit. A service node acknowledges an empty run message by sending either a run message containing user data or an empty run message. Either form of acknowledgment confirms that the virtual circuit is still intact. As for any run message,when an empty run message is unacknowledged, the server retransmits the message until it has reached its retransmit limit. Thereafter, the server treats a lack of acknowledgment as a suspected circuit down event.

If a service node sends an empty run message, the server is not required to answer.Instead, the service node sets a flag indicating a possible balanced virtual circuit. If the server has no information to transmit, the circuit becomes balanced, and the service node experiences a delay before receiving the next run message from the server. However, the balanced-circuit flag enables the service node to send one unsolicited run message, if it accumulates one or more transmit buffers of data before the server sends its next run message. Being able to send an unsolicited run message permits a service node to end a balanced state.

Figure 4-4 summarizes the overall operation of the Virtual Circuit layer.

Figure 4-4: Operation of the Virtual Circuit Layer



1. The Slot layer establishes the first slot on a circuit.

2. The service node accepts the circuit request.

3. The circuit timer expires and the Slot layer data (all that can be contained in one RUN message) is transmitted.

4. The service node acknowledges the receipt of the data and optionally sends data to the terminal server.

5. The keepalive timer expires (due to lack of other data) and the terminal server sends a keepalive message.

6. The service node acknowledges receipt of the message.

7. The Slot layer terminates the last slot on the virtual circuit.

4.5 The Host-Initiated-Request Process

General-purpose service nodes using software based on Version 5.1 of the LAT architecture can be configured to make host-initiated requests. Configuring a service node for making host-initiated requests is specific to the operating system of a service node. For information on what steps are involved in setting up a VMS service node for making host-initiated requests, see Section 5.11.

The **host-initiated-request process** is a process that involves two message exchanges between a service node and a server.

1. Translating a server's node name to its Ethernet address
2. Making the actual host-initiated request asking the server to establish a connection between a specific server port or service and the service node

4.5.1 Translating a Server Name

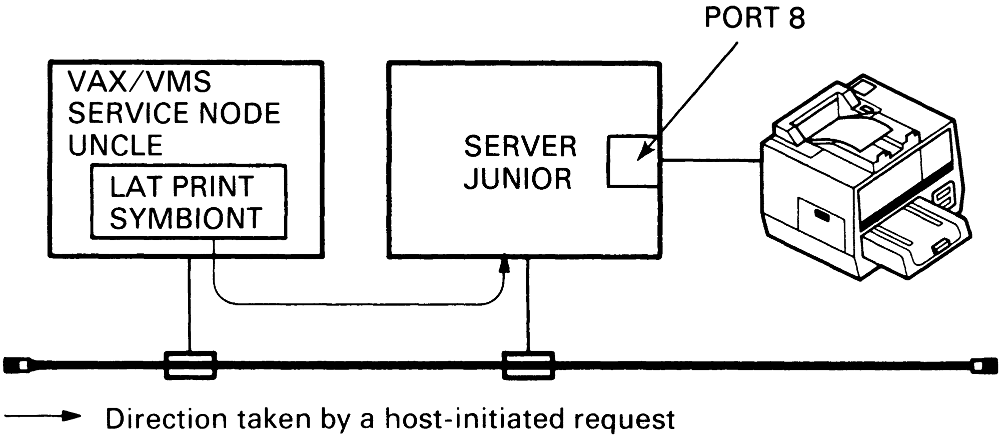
When a service node needs an applications device on a server for one of its applications, the service node multicasts a solicit-information that designates a particular server's name. Other servers discard the message. If the designated server shares at least one group with the service node, it accepts the solicit-information message. The server then sends a response-information message that provides the server's Ethernet address and other information to the requesting service node. If there is no shared group, the server discards the solicit-information message.

4.5.2 Making Host-Initiated Requests

After translating a server's name to its Ethernet address, the service node sends a command message that contains the host-initiated request. The service node addresses this command message to the server whose address was solicited. The command message requests a particular server port or (with dual-purpose servers only) service consisting of one or more server ports. For the server to accept a host-initiated request, the server port and service node must share at least one group, and the targeted port(s) must be configured for an applications device. Otherwise, the server rejects the request. Note that the command message supplies a number that identifies the service node's connection request. This request number is used during subsequent communication about that request.

Figure 4–5 illustrates the path of a command message carrying a host-initiated request from the VAX/VMS service node UNCLE to the server JUNIOR for port 8.

Figure 4-5: The Path of a Host-initiated Request



LKG-1190-87

4.5.3 Processing Host-initiated Requests

A server processes host-initiated requests on a first-come-first-served basis. When a command message arrives, the server immediately checks the availability of the appropriate port or service. There are three possible outcomes: the server can accept, queue, or reject a host-initiated request. The actual outcome is determined by the following considerations:

1. If a port or service is available, the server ensures that there is a virtual circuit with the requesting service node, creating a new one if none exists. The server then sends a start slot on a run message.
2. If for any reason a host-initiated request is invalid, the server rejects the request by sending an error status message to the service node. The error status indicates the reason for the rejection. Likely reasons that a request is invalid include:

On any server,

- The requested port is unknown.
- The server port lacks any common group with the service node.
- The requested port is not configured for an applications device.

On dual-purpose servers,

- A requested service name is nonexistent.
- Connections are disabled for a service.
- A service has no assigned port.
- A service is not offered on a requested port.

3. If an appropriate port or service is unavailable but does exist, the server responds in one of two ways. It either queues the request or rejects it.

   Queuing a request requires several conditions:

   - The host-initiated request must indicate that queuing is acceptable.
   - If a service name is being requested, queuing must be enabled for that service.
   - There must be space on the connection queue.

   If all of these conditions exist, the request for the busy port or service is queued. Otherwise, the server rejects the request and sends an error status message to the service node indicating that the service is in use. Depending on the application, the service node might attempt the host-initiated request again later.

Figure 4-6 illustrates the functions of each layer of service-node software in the host-initiated-request process. It is assumed that the service node declines queuing in its command message. The figure illustrates the following service-node functions (the following numbered list corresponds to the numbers on the figure):

1. An application process on a service node requests a connections to a port or service on a specific server.
2. The Service Class 1 module of the service node multicasts a solicit-information message to get the address of the server.
3. The response-information message received from the server goes directly from the service nodes's Data Link layer to its Service Class 1 module.

4. The Service Class 1 module of the service node sends a command message directly through the Data Link layer using the server's Ethernet address. This command message contains the host-initiated request.)
5. If the requested port or service is available, the server responds with a virtual-circuit start or run message.
6. If the port or service is unavailable, the server responds with an error status message rejecting the request. An error status message goes directly from the Data Link layer of the service node to its Service Class 1 module. If queuing were permitted by the service node, the server might queue a request rather than reject it.

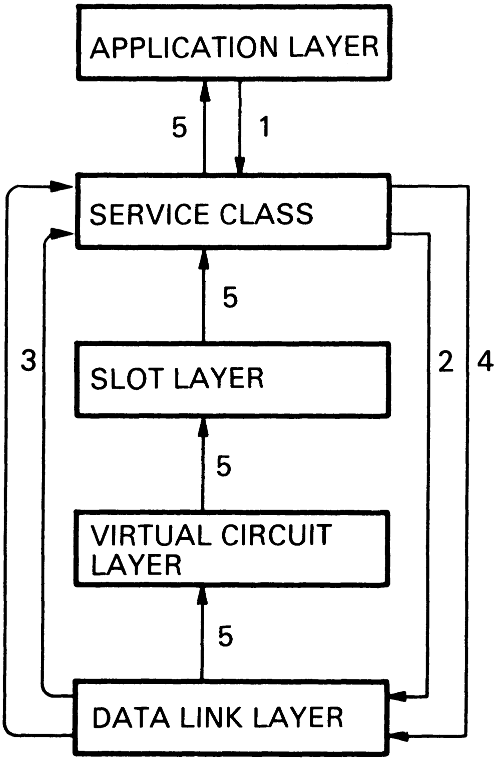
4.6 Connection-Queue Maintenance

When a connection request is queued by a server, the server and requesting node can exchange command and status messages about the server port or the position of the request in the connection queue. A server sends periodic status messages to the requesting node. The messages indicate the status of the connection request in relation to other requests in the queue. The messages also note the queue entry identifier. The rate at which these status messages are broadcast varies among servers. On some dual-purpose servers such as the DECserver 500, the rate is determined by the multicast timer. On other servers such as the DECserver 100 and DECserver 200, status messages are always sent once a minute.

The connection queue is is a first-in-first-out (FIFO) queue with no special priorities. The server begins every search of the queue with the earliest entry, which is the entry that has currently been queued for the longest period of time. However, the server logically divides queued entries according to the resources they request. This division is achieved by selective searching through the queue (beginning with the earliest entry) until the server finds the first entry requesting a specific newly available port or service. Therefore, the single queue provides a resource-sensitive environment. The process for searching queues is explained further later in this section.

Queuing connection requests permits the server to give equal access to devices, such as printers, when connection requests overlap. See Figure 4-7 for an illustration of the processing of multiple connection requests for the same port. The first request for port 8 is connected immediately. The second request for port 8 is received by the server while the port is occupied. With queuing enabled for the port, the server can accept the second request and

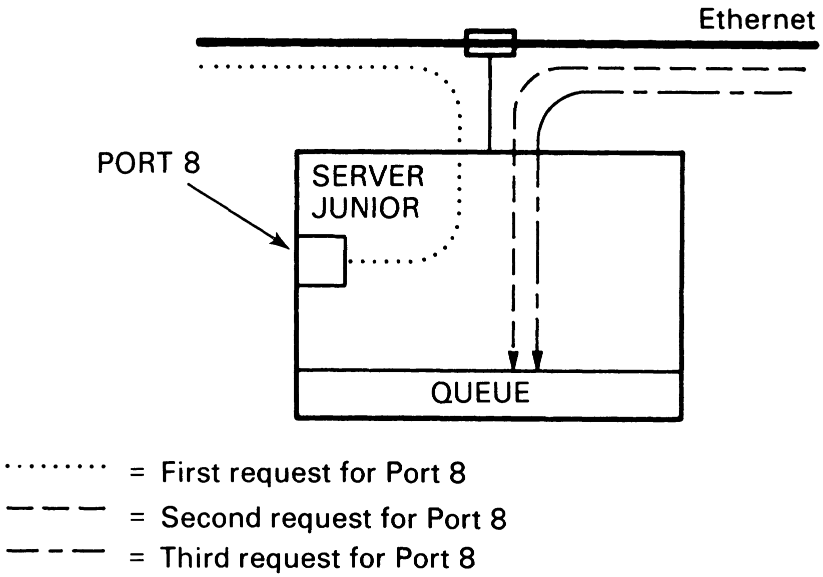
Figure 4-6: Service-Node Functions in Host-Initiated-Request Process



LKG-1199-87

place it in the queue behind all existing entries. In this manner, queuing connection requests allows the server to temporarily postpone establishing a session until the proper resources become available. The same basic procedure occurs when the connection request specifies a service name. The only difference is that when more than one server port offers the service, an entry requesting that service might be dequeued after any of those ports becomes available.

Figure 4-7: Processing Queued Connection Requests for Same Port


Ethernet
PORT 8
SERVER JUNIOR
QUEUE
········ = First request for Port 8
— — — = Second request for Port 8
— - — = Third request for Port 8

LKG-1206-87

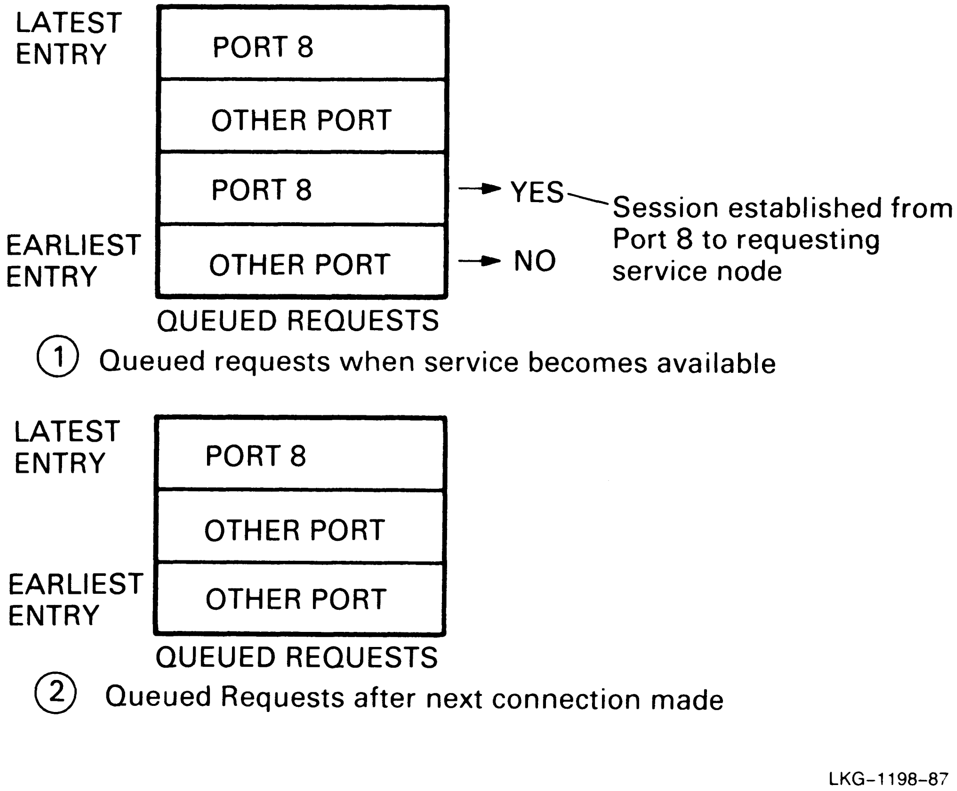
A server manager can remove an entry from the server queue. When this happens, the server sends a status message to the requesting node. The message reports that the request is being rejected. Also, at any time, a requesting node can instruct a server to delete any of its connection requests from the server's connection queue.

When a given remote-access port becomes available, a server searches its connection queue to see if a connection request is pending for the port. The server dequeues the earliest request for the port in question. After dequeuing an entry, the server attempts to establish the requested session.

Figure 4-8 illustrates how a server processes its connection queue after a remote-access port (in this instance, port 8) becomes available. While looking for requests that specify port 8, the server bypasses requests for all other ports (regardless of the current availability of the other ports). Note that port

8 might be the only port mentioned in the connection request, or it might be one of a series of ports listed for a requested service. In either case, the server considers the request as being for port 8. After removing the earliest request for port 8 from the connection queue, the server enters the session-establishment process for port 8.

Figure 4-8: Processing the Queue After Port 8 Becomes Available



5

Managing a LAT Network

This chapter introduces the activities involved in setting up and managing a LAT network. How to do each activity is explained in the installation and management documentation of specific LAT nodes.

This chapter considers the following activities:

- Preparing for server hardware installation
- Installing server software
- Customizing serverwide characteristics
- Configuring server ports
- Setting up service nodes for offering services
- Creating services
- Managing services
- Setting up host-initiated request capability
- Setting up service nodes for host-initiated requests
- Setting up server ports For host-initiated requests
- Managing the connection queue
- Managing communications on a LAT network
- Using the Terminal Server Manager (TSM) product

5.1 Planning the Physical Configuration of a LAT Network

Network planners and managers have a considerable degree of freedom in designing a LAT network. However, there are a few constraints on the configuration of a LAT network. The major constraints are as follows:

- A LAT network must reside on an Ethernet LAN.
- A LAT network must contain at least one terminal server to enable access to LAT services. This is necessary since only the server software can initiate sessions.
- A LAT network must be on a LAN with at least one appropriate multiuser computer to serve as a load host. (Digital recommends a minimum of two load hosts for each terminal server and no more than ten terminal servers for a load host.)
- Except for ULTRIX systems, load hosts must be running DECnet Phase IV software to perform load host functions.
- A LAT network must contain at least one service node (either a multiuser computer or a dual-purpose server).
- Not all types of general-purpose service nodes can access devices on servers for applications programs.
- Not all servers can function as service nodes.
- File transfers between a personal computer and another computer depend on the two systems sharing a common file transfer program. Therefore, while non-LAT hosts are readily accessible and usable as services, the data on them is not automatically transferable.

5.2 Preparing for Server Hardware Installation

Server hardware installation varies widely among servers. Some servers, such as the DECserver 200 system, are customer installable. Other servers, such as the DECserver 500 system, must be installed by a Digital field service technician. Installing server hardware, however, always requires that the site is suitable prepared for the server unit. While the actual site requirements of hardware vary considerably, there are some common considerations for preparing the site. This section provides an overview of those site preparation considerations.

Site preparation involves:

- Making sure the installation site meets the requirements identified in the site preparation instructions of the server hardware unit
- Installing and testing any wiring that is required for devices that will be connected to the server
- Installing any prerequisite hardware so that power, network, and port device connections can be made

The hardware site for each server must meet the server's requirements for the following:

Physical Space

The space where a server is installed must accommodate the dimensions of the server. Also, it must allow air flow around the server. Server air vents must remain free once the server is installed.

Environment

The server installation site must be compatible with the server's environmental requirements as listed in the site preparation or hardware installation guide.

Cabling (Power, Network, and Device Connections)

The server installation site must allow the power cord, the transceiver cable, and the device cables to be connected to the server without being strained. Furthermore, the server must be supplied with the power requirements identified in the site preparation or hardware installation guide. Also, servers must connect to a suitable Digital transceiver, which must be installed before server installation.

Preparing to Connect Devices to Server Ports

A variety of wiring strategies are possible, depending on the server. The system or server manager must make appropriate arrangements to connect port devices.

5.3 Installing Server Distribution Software

The set of files contained in the software distribution kits provided by Digital for each type of server is called the **server distribution software**. Every terminal server has system-specific software distribution kits for each type of load host that supports the server. To a large extent, the contents of server distribution software depends on the requirements of both the server and the operating system of the load host.

Some of the distribution software performs functions that are specific to load hosts. For most servers, the server distribution software includes a menu-driven procedure for customizing the load host's node database; for example, the DSVCONFIG procedure, which is used with several servers. Though it is system-specific internally, DSVCONFIG has a standard user interface on all load hosts and for all terminal servers.

Other files appear in the software distributions kits for a specific server on all of its load hosts. Server-specific distribution software always includes the server image source file, whose contents depend on the type of server. If a server's permanent database resides within its server image file(s), the server's distribution software also includes:

- A Terminal Server Configurator utility and associated on-line help (this utility exists only for servers whose permanent database resides on load hosts).
- Command files that restore defaults to permanent databases that are stored in server images on load hosts.

Load hosts perform important maintenance tasks for terminal servers. A **load host** is usually a multiuser computer that stores the server distribution software and the individual server image files of one or more terminal servers on the same LAN.

For most operating systems, the load host must be a DECnet node. However, DECnet is not required for ULTRIX load hosts. The functions of load hosts are independent of LAT network functions, so a load host need not be a LAT service node. VAX/VMS systems support load-host functions for all terminal servers. For other operating systems, see the server's SPD to identify other load-host systems. The following table lists the operating systems that support load-host functions for one or more terminal servers products.

Table 5-1: Operating Systems with Load-Host Capability

Operating system	Comments
RSX-11M-PLUS	
Micro/RSX	
MicroVMS	Available for all servers
TOPS-10	
TOPS-20	
ULTRIX	Unavailable for servers requiring a TSC
VMS	Available for all servers

5.4 Customizing the Load Host's Node Database

After copying the distribution software to a load host, a system manager customizes the load host's node database to support the new server(s). **Customizing of the load host's node database** means defining an entry for each server in the load host's node database.

The **load host's node database** comprises the DECnet permanent database, the DECnet operational database, and the server configuration database. An entry in the load host's node database identifies a server's type (such as, DECserver 100), its DECnet node name and address, its Ethernet hardware address, the name of any server image file, and other identifying information. These entries provide information for down-line loading and up-line dumping.

The DSVCONFIG Procedure

All terminal servers have a configuration procedure for some or all of their load hosts. For most servers, customizing a load host's node database involves using a command procedure called DSVCONFIG, which is part of the server's distribution software.

DSVCONFIG performs the following functions:

- Listing the servers loaded by the load host
- Adding a new server unit
- Swapping an old unit with a new one

- Removing a unit
- Restoring the previous configuration from the local database

After modifying the database, the manager must ensure that every other load host for the server is updated. The server manager must coordinate running DSVCONFIG with the load host system manager because, for most load host's operating systems, the server manager needs privileges to run this procedure.

After installing a server or after changing its permanent database, the system or server manager must down-line the server image to the server. A server manager can initiate a down-line load by using a number of methods, which are described in the documentation of specific terminal servers.

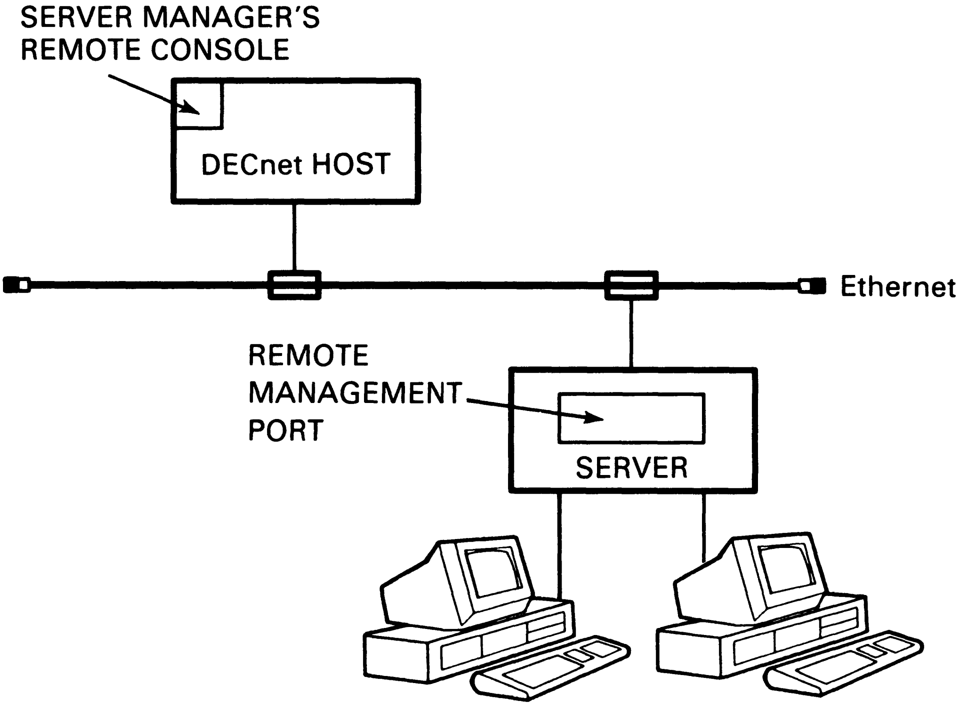
5.5 Using the Remote Console Facility (RCF)

The Remote Console Facility (RCF) provides a useful management tool for LAT networks with a small number of terminal servers. RCF is provided by Phase IV DECnet nodes that implement the DNA Maintenance Operation Protocol (MOP). Note that ULTRIX nodes do not require DECnet to implement MOP. From these nodes, RCF provides access to devices on an Ethernet network that accept RCF connections. Terminal servers are among the types of Ethernet communications servers that RCF accesses. RCF is invoked differently on different types of operating systems. Using RCF, managers can perform diagnostics and customize a server remotely from a single centralized terminal.

An RCF user can work at any terminal that is capable of logging into a DECnet node on the same LAN as the server. RCF establishes a logical connection between that terminal and the remote management port of a specific terminal server. The **remote management port** of a terminal server is a logical software-based port. Note that this port is called the **console port** in some terminal server documentation. A server's remote management port is accessible through RCF or through the Terminal Server Manager (TSM) product, which implements its own RCF to manage servers remotely.

Figure 5-1 illustrates the logical relationship between a remote console and the remote management port of a terminal server.

Figure 5-1: Relationship of Remote Console to a Terminal Server



LKG-1204-87

Password protection curtails general access to servers using RCF. Some servers can require a maintenance password for RCF access. After entering the correct log-in password, the server manager enters local mode on the server's console port. There, the manager can execute most terminal server commands as if the terminal were directly attached to the terminal server hardware.

5.6 Customizing Serverwide Characteristics

This section considers serverwide characteristics that help to identify the server on the LAT network and to protect the server from unauthorized access.

- Server name

  Specifies the server's name, which identifies it as a service node on the LAT network. The name must be unique on the network.

- Server identification string

  Specifies the server ıdentification string, which is included in multicast announcements to describe the local service node to users.

- Server number

  This is a value from 0 to either 32767 or 65535 (depending on the server) that identifies a server. The server number should be unique. In some situations the number is used along with the server name to identify a server.

- Specifying serverwide passwords

  Routinely changing passwords helps maintain server security.

- Server groups (only for Ethernet Terminal Server V2.2 and earlier)

NOTE

Serverwide characteristics (such as service-node groups) that exist only on dual-purpose servers are discussed in Section 5.8.1.

5.7 Configuring Server Ports

Server managers are responsible for configuring each server port for the attached device and the use intended for the device.

5.7.1 Configuring Ports for Interactive Devices

Terminal servers are character-oriented and must know the physical characteristics of any device on a server port. Local-access ports generally can use the autobaud facility to adjust the speed, parity, and character size of the port to the equivalent characteristics of the attached interactive device. For autobaud to function, a device must have either of the following sets of characteristic settings:

- A character size of 8, no parity, and any supported speed
- A character size of 7, even parity, and any supported speed

The autobaud facility does not function with unusual physical characteristics such as mark parity or split-speed operation. Some interactive devices cannot be reset so that autobaud can function. In this case, the server manager sets the server port to autobaud disabled and redefines the physical characteristics of the port as needed.

The final aspect of configuring the port to an interactive device is to ensure that port is compatible with the output format used by the device. This facilitates using the terminal in of local mode.

In addition to configuring a local-access port for the attached device, a server manager can customize a port to a particular user or set of users by redefining the user-oriented characteristics of a server and its ports. A user-oriented characteristic is a characteristic that affects the local-mode environment of a local-access port.

The server manager manages user-oriented characteristics that are controlled by privileged commands; for example, enabling the log-in password on a port-by-port basis, specifying secure status, and authorizing port groups. Server users control user-oriented characteristics that are modified by nonprivileged commands (or the subset of these commands available on ports with secure status).

5.7.1.1 Assigning Port Groups

Assigning the necessary group(s) to each server port is the server manager's responsibility. Assigning groups to a server port requires information about the services to be accessed by the person (or persons) that use the server port.

If the LAT network manager maintains an up-to-date record of service groups and intended users for each service, a server manager can use it to plan groups for server ports. If this information is not available, the server manager must identify the requirements of the user or users of each interactive server port. In some cases, the manager might anticipate that a specific user will use a terminal on a particular port, such as a person with a terminal located in a private office. In other cases, there may be an identifiable set of users for a port, such as a employees accessing a common database. In both cases, the manager should be able to gather information from the users about what services they need. Then, the manger can enable groups for ports accordingly. Port groups specified by a server manager are generally called **authorized groups**.

In other cases, the manager might have a terminal that is available to many kinds of users. In this case, memory usage or security issues might be the best way to decide on the groups to assign for the port.

5.7.1.2 Educating Users

Digital suggests server users be informed about the features of local-access ports, such as multiple sessions and related commands/switches, the types of services available on the LAT network, load balancing among VAXcluster nodes, automatic failover, and file transfer capabilities. In addition, users should be encouraged to use the server's tutorial and reference on-line help and user's guide. Finally, users need to know whom to contact if they have problems.

5.7.2 Configuring Ports for Applications Devices

Port access must be set to either remote or dynamic. and a number of physical characteristics often require new values. These characteristics are:

- Physical characteristics

  On a remote- or dynamic-access port, the autobaud facility cannot normally be used (unless the port device is a terminal and the application is initiated by user input). Therefore, you usually must change settings for unmatched physical characteristics in one of the following ways:

  - Change the character size, parity, and speed on the device itself to match the settings on the server port.
  - Change the character size, parity, and speed of the port to match those of the device.

- Modem control and associated characteristics

  A modem-controlled device such as a non-LAT host or a modem requires the server manager to enable modem control and, sometimes, associated characteristics on the server port.

5.8 Setting Up Service Nodes for Offering Services

Every manager of a service node, whether it is a general-purpose service nodes or a dual-purpose server, is responsible for the following basic tasks:

- Setting up the characteristics of the service node
- Setting up services
- Managing services

The commands used to manage service nodes differ substantially between general-purpose service nodes and dual-purpose servers. However, there are commonalties in service node management. For example, all service nodes provide commands that identify the service node, create and manage services, and display the values of characteristics. Note that when both types of service nodes are discussed together, system managers and server managers are called **service-node managers**.

5.8.1 Setting Up Service-Node Characteristics

Service-node managers must coordinate the settings of service-node characteristics with the server managers. Often such coordination is the responsibility of a LAT network manager.

Any service-node manager can control the following service-node characteristics:

- Node name
- Node identification string
- Service-node groups
- Multicast timer

5.8.1.1 Service-Node Names

All service nodes must have a node name that is unique to the LAT network. For LAT nodes that have DECnet-node names, Digital recommends that the same name be used. The DECnet-node name must be unique within the entire DECnet network.

General-purpose service nodes supply a default name for service nodes. The choice of the default name is system-specific. For example, VAX/VMS systems use their DECnet-node name if they have one. Since RSX systems cannot have LAT without DECnet, they can always use their DECnet-node name as their default service-node name.

For dual-purpose servers, the service-node name is always the same as the server name. A server's name can be changed by its manager.

5.8.1.2 Node Identification String

The node identification string is a description for your node. The node identification string is announced in to servers in service-announcement messages.

Some service nodes have default node identification strings; others do not. If there is no system default and none is specified by the service-node manager, then only the service node's name is sent in the service-announcement messages.

5.8.1.3 Service-Node Groups

A service node's groups affect all of its services equally. Service-node groups determine whether a server can establish sessions with any service on the service node.

NOTE

> On a dual-purpose server, its service-node groups are enabled by using the SET SERVER SERVICE and DEFINE SERVER SERVICE commands. For this reason, service-node groups are generally termed **service groups** in terminal server documentation.

Assigning service-node groups requires coordination among all managers of service nodes (including of dual-purpose servers) on a LAT network. Groups should be allotted before any service-node groups are actually set up. The network manager should use a service-specific approach to assigning service-node groups on the LAT network. The **service-specific approach** involves selecting a group for either an individual service or a set of related services offered on the LAN and assigning that group to all service nodes offering that service.

The service-specific approach is useful when several nodes offer the same service (as with a VAXcluster offered as a single service or several servers offering modems as part of a single service). This approach also helps when multiple ports are configured for host-initiated requests and are offered in common to service nodes.

The service-specific approach to assigning service-node groups has advantages to both service node and servers. These advantages are:

- To service nodes:

  Service-specific service-node groups allow server users to access services independently of what service nodes currently offer them. With service-specific groups, a system manager can add an existing service to a new service node without involving server managers.

- To servers:

  Since the interactive users are oriented to services rather than to service nodes, this approach helps when assigning authorized groups for services offered by multiple service nodes.

Another advantage to servers is that the number of authorized groups remains small for each port. Furthermore, if the relationship changes between a service and the service nodes offering it, it is not necessary to update the groups of every port that requires that service.

A variation of the service-specific approach helps a server manager assign port groups to ports that are configured for host-initiated requests. These ports must share a group with each service node making requests. The server manager must tell system managers of those nodes what service-node group is required to access those server ports.

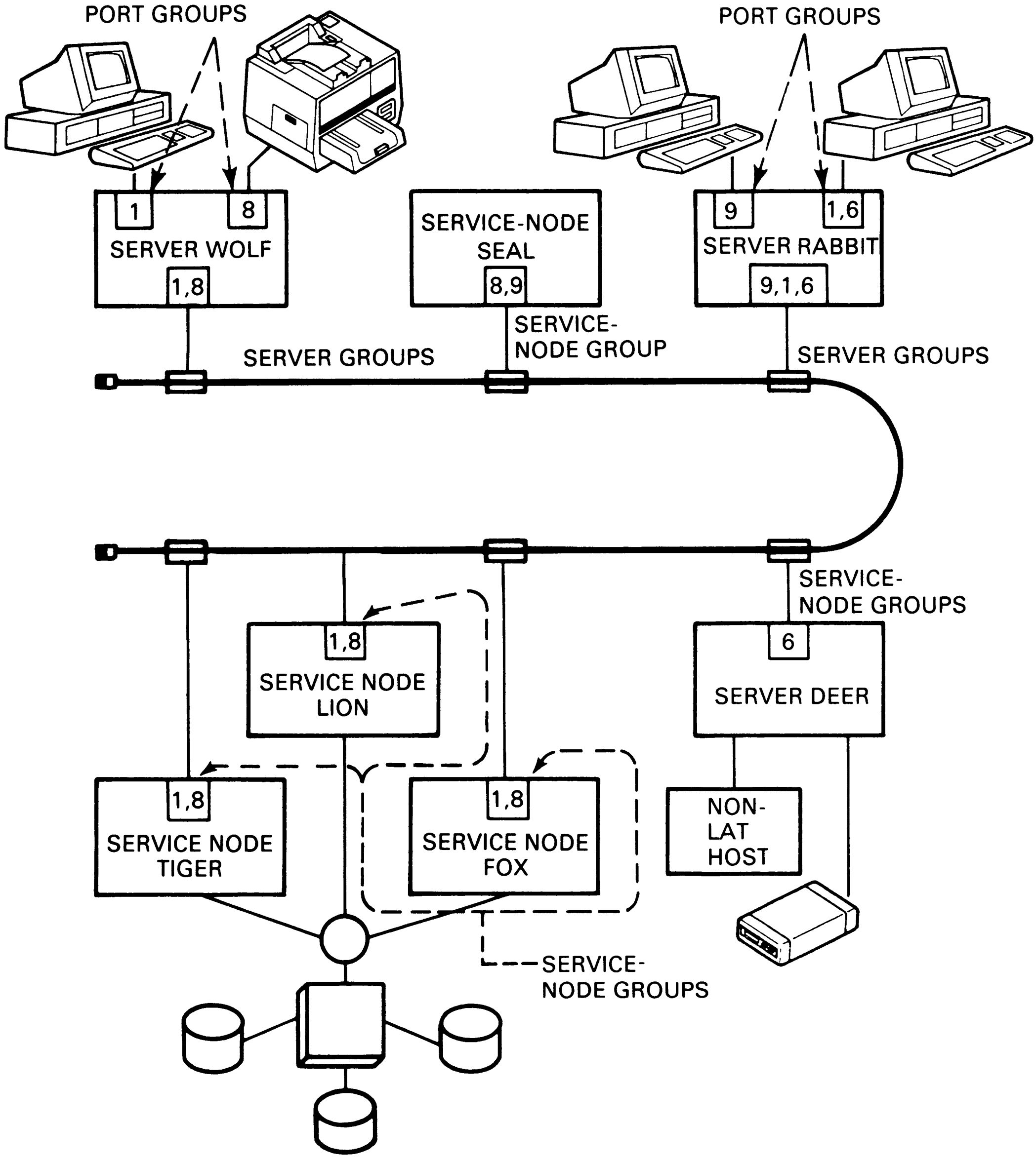
Server managers need to know the service-node groups that are enabled on each service node. They can then determine which users need these services and assign the same groups to their server ports.

Table 5-2 lists shows show how a sample LAT network might be divided into four logical networks by assigning four groups—1, 6, 8, and 9—to service nodes and to server ports (or servers). Note that the default group, 0, is disabled across the network. This ensures that specific groups must be assigned to all service nodes and server ports (or servers). The sample network and the logical subnetworks formed by each group are illustrated in Figures 5-2 through 5-6.

Table 5-2: Groups on a Sample LAT Network

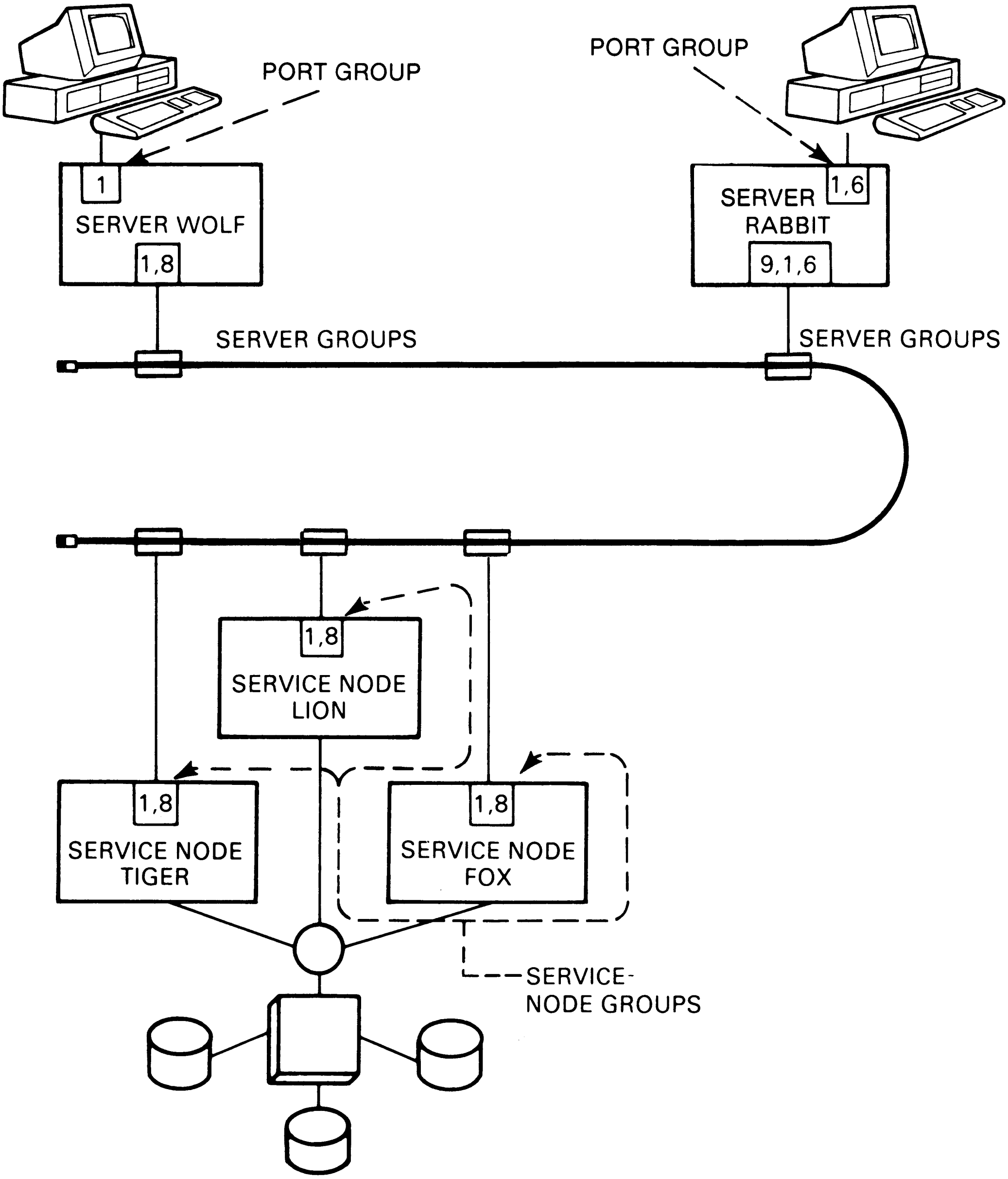
Resource Description	Group	Offering node(s)	Accessing entity
none	Group 0	Disabled	—
Cluster	Group 1	Service nodes Tiger, Lion, and Fox	User(s) on servers Wolf and Rabbit
NonLAT Modem	Group 6	Server Deer	User(s) on server Rabbit
Printer	Group 8	Server Wolf	Service nodes Seal, Tiger, Lion, and Fox
Computer Seal	Group 9	Service node Seal	User(s) on server Rabbit

Figure 5-2: Group Assignments on a Sample LAT Network



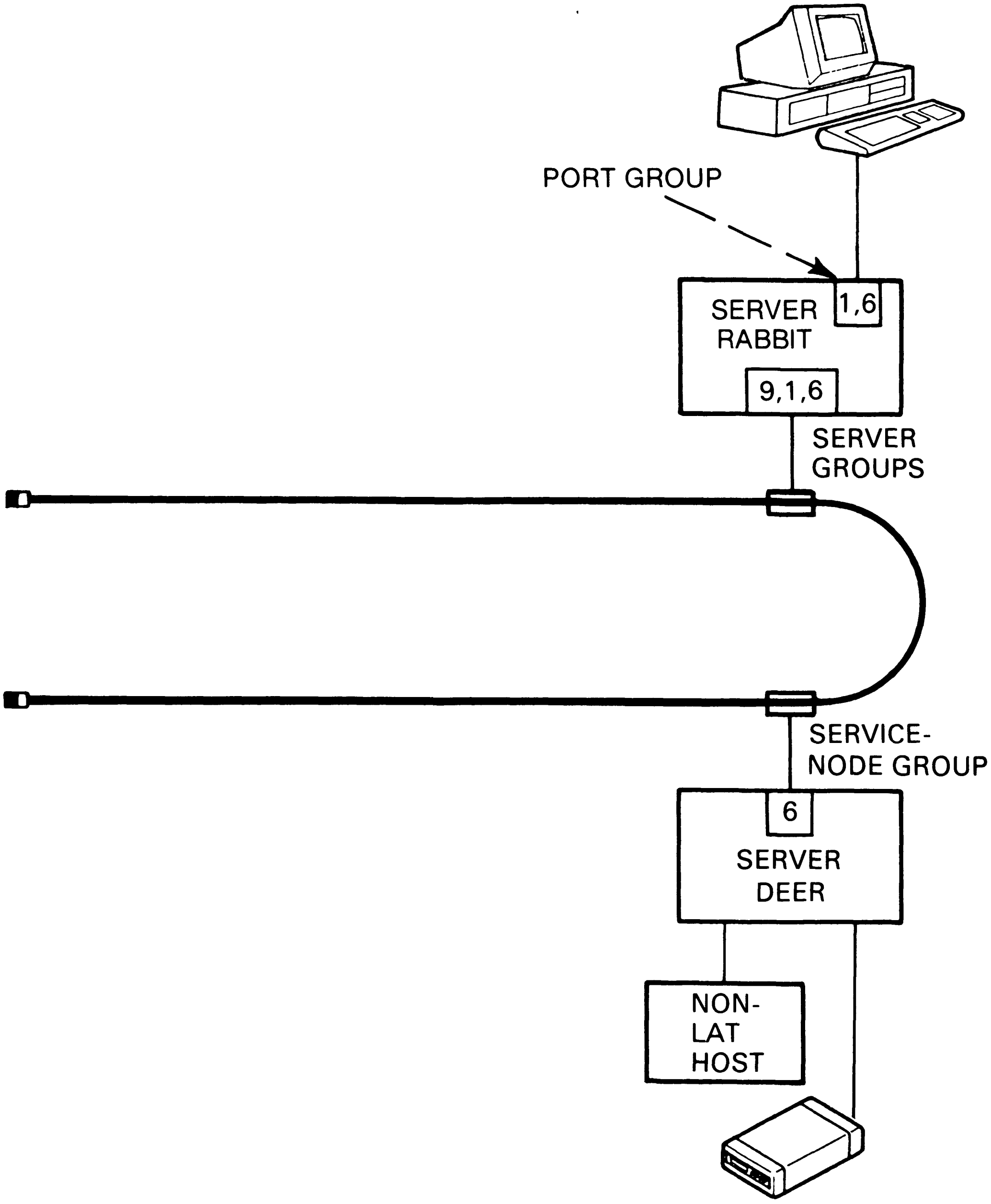
LKG-1194-87

Figure 5-3: Logical Subnetwork Formed by Group 1



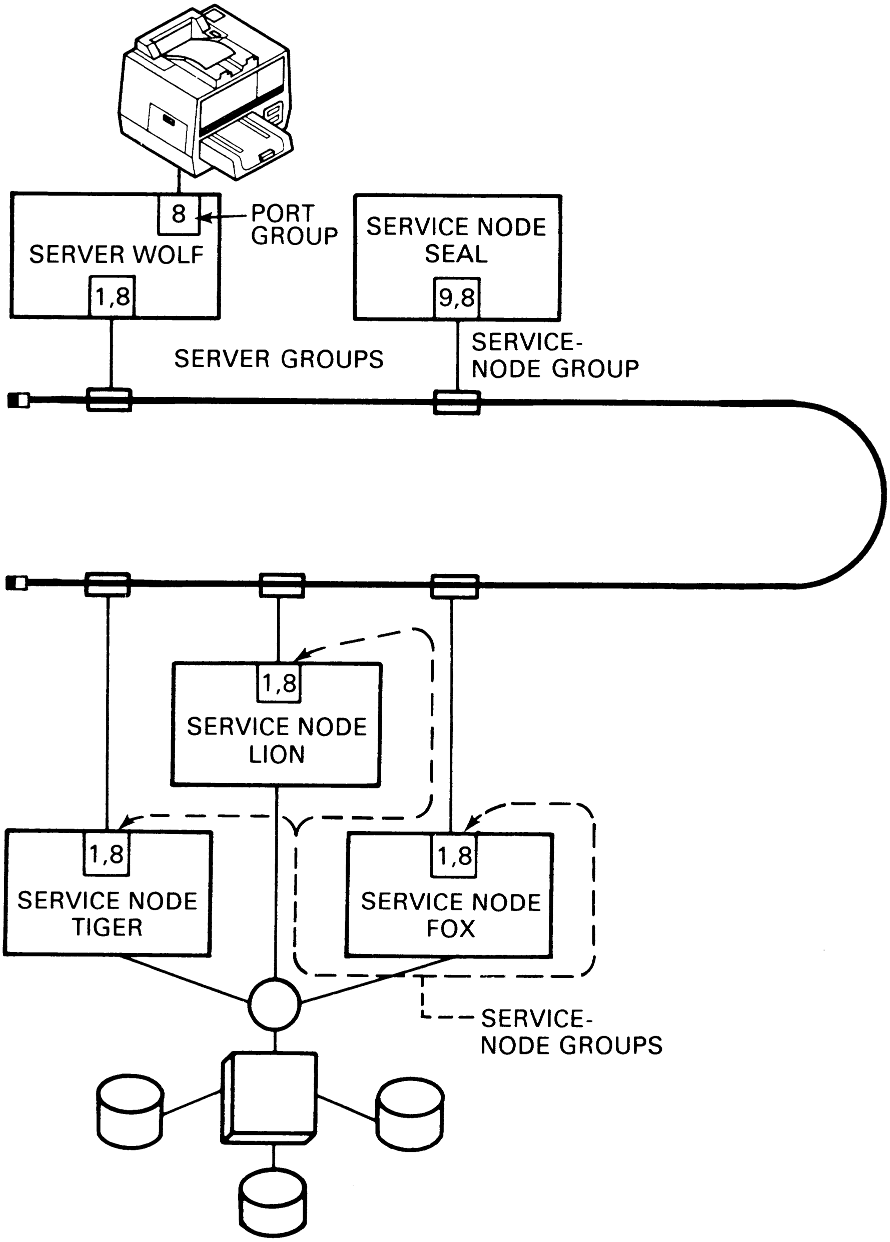
LKG-1196-87

Figure 5-4: Logical Subnetwork Formed by Group 6



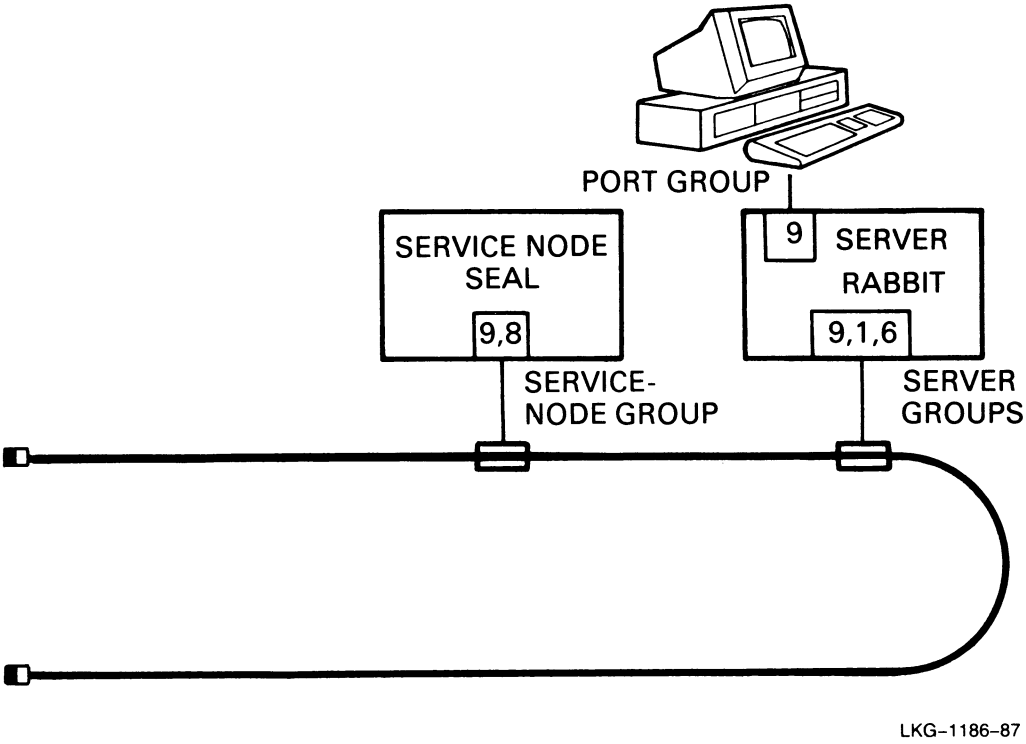
LKG-1197-87

Figure 5-5: Logical Subnetwork Formed by Group 8



LKG-1187-87

Figure 5-6: Logical Subnetwork Formed by Group 9



5.9 Creating Services

This section discusses the factors that concern all service-node managers when establishing a service. These factors are:

- Service names

  A service-node manager should coordinate naming a local service with the system and server managers that manage other service nodes. Several service nodes can share one service name. Sometimes duplicating service names is desirable, as for example, with VAXcluster members.

- Service identification string

  Identifies a service to users. A service-node manager should specify a meaningful service identification string to help people understand the nature of the service.

5.9.1 Assigning Service Names

Service names are assigned by the network manager, by the system manager, and by the server manager, who must all cooperate to coordinate the name assignments. These managers must specify a unique service name for each service created. Service names can be up to 16 alphanumeric characters long. The potential length of service names allows managers to identify services in a manner that allows users to understand the function of each.

5.9.2 Special Considerations for Dual-Purpose Servers

A service without accessible ports would be given a service rating of 0 by the server, which means that no one can connect to this service. There are two general prerequisites for assigning a port to a a service on a dual-purpose server:

- Configuring the port for the applications device
- Assigning one or more server ports to the service.

5.10 Managing Services

Service-management features vary between general-purpose service nodes and dual-purpose servers.

5.10.1 Managing Services on a General-Purpose Service Node

On some general-purpose service nodes such as VAX/VMS systems, a system manager has the option of overriding the dynamic generation of a service rating by assigning a static service rating. Where available, a static service rating value can be used to direct server users temporarily away from or toward using the service on that service node.

5.10.2 Managing Services on Dual-Purpose Servers

The service characteristics of a dual-purpose server allow the server manager to manage local services. Most of the characteristics affect specific services.

On a serverwide basis:

- Announcements characteristic

  Specifies whether the dual-purpose server transmits LAT multicast messages for the services it offers. By default, a server begins multicasting service-announcement messages as soon as a service is defined. Multicasting continues until the server manager either clears the last local service or disables announcements.

On a service-specific basis:

- Connections characteristic

  Specifies whether future connections to the service are allowed. Current connections are not affected. The default is having connections enabled.

  When a server manager disables connections for a service and queued requests for that service exist, the server dequeues them at the appropriate time and then rejects them.

- Password characteristic

  Specifies a service password, which a user must enter when requesting a connection to the service.

- Queue characteristic

  Specifies whether queued connection requests for the service can be held on the connection queue. Managing the queue is discussed in Section 5.13.

5.10.2.1 Using Service Passwords

If a server manager defines a service password for a service, the server prompts a user for the password before completing a connection to the service. A service password is particularly useful for unprotected devices such as modems used for dialing out. Note, however, that attempts to connect to password-protected services fail when the connection request is initiated from some older versions of terminal server software that do not support connections to password-protected services. Any request from such a server to start a session with a password-protected service is denied. The user is not prompted for a password but is notified that the connection failed because of an invalid password.

Service passwords do not affect host-initiated requests for services.

5.11 Setting Up VMS Service Nodes for Host-Initiated Requests

A VAX/VMS service node (VMS V4.2 and later) can be configured to make host-initiated requests for applications devices on behalf of applications programs. However, for VMS V4.2 systems, this configuration requires installation of a new service-node software called LATplus, which replaces the form of service-node software that is bundled with VMS V4.2 systems. LATplus is a VMS layered product that comes in server software distribution kits for VMS V4.2 load hosts. For later VMS versions, this new service-node software will be bundled with the operating system.

An applications program requests a logical device that is mapped to one or more remote devices on a specific server. A VAX/VMS service node uses a special application software module to route a file from a VMS print queue to a terminal server. This software module is called the LAT print symbiont (LATSYM). LATSYM routes the files through a VAX/VMS applications port, over a session, and to a server port with an attached applications device (which is usually a printer). Note that in the LATplus/VMS documentation, applications devices on terminal servers are called **remote devices**. For consistency with VMS LAT documentation, the printer is called a remote printer in this section.

The following series of tasks is involved in setting up a service node for making host-initiated requests for printers:

1. Creating Applications Ports on Service Nodes

   The system manager configures a VAX/VMS service node by creating a logical device on the service node. The logical device used by an application program on a service node is called an **applications port**. After being suitably configured by the system manager, LATSYM undertakes the host-initiated-request process for any print job sent to that applications port.

2. Mapping Applications Ports to Server Ports and Services

   The destination of a host-initiated request is determined by the system manager by mapping an applications port to a specific server and port or service. First, the server manager specifies the server by server name. Using the server name rather than the Ethernet hardware address allows server units to be swapped without affecting the service node.

   Secondly, the server manager specifies either the port identification of a server port or (for dual-purpose servers only) a service name. When the system manager specifies a server port for an application, the server always establishes the session for that port. When the system manager specifies a service name, the server goes to its operational database to translate the service name into a list of server ports. If one or more ports are available, the server selects the lowest-numbered available port in the port list.

3. Specifying a service-node group

   When configuring a service node for making host-initiated requests to a server, the system manager needs to know what group to enable on the service node. The group must be a port group of one or more ports offering the required applications device. Since the operation of port groups varies among servers, see server-specific documentation for details.

4. Setting Up Printer Characteristics

   The system manager customizes the applications port to match the physical characteristics of the remote printer and sets up the applications port as a spooled device. Matching the physical characteristics of the printer ensures that the printer receives user data in the proper format. Setting up spooling allows multiple users to share fairly a single printer or device.

5. Setting up a print queue for the remote printer(s)

   The system manager initializes a print queue for the remote printer(s) on a specific server. Like any print queue, a queue processed by LATSYM requires that the system queue manager be running before the print queue starts. The queue manager provides the link between the applications program and a print queue processed by LATSYM.

5.12 Setting Up Server Ports for Host-Initiated Requests

This section discusses what is involved in managing a port used for receiving host-initiated requests.

5.12.1 Deciding Whether To Assign a Service Name

Except on the DECserver 100 and MUXserver 100 systems, a host-initiated request can specify a service name instead of a port number. A server configured for host-initiated requests can have devices accessed by a port identifier without the use of a service name. Frequently, however, assigning and using a service name offers advantages.

One advantage is that a service name can be associated with equivalent devices on more than one remote-access port. Assuming that queuing is enabled for the service, these ports respond collectively to host-initiated requests on a first-come-first-served basis as they become free. With two or more ports, the delay for users of the service is minimized.

At the same time, the possibility of disabling queuing for any service permits a server manager to control the queuing of host-initiated requests for the service, if necessary. Furthermore, a request for named services can be controlled by disabling connections and/or disabling one or more groups for the service node. These features do not affect host-initiated requests that specify a port without a service name.

5.13 Managing the Connection Queue

The connection queue of the server is a first-in-first-out (FIFO) queue with no special priorities. The basic operation of the queue is explained in Section 4.6.

- Displaying connection queue entries

  The server manager can selectively observe the status of connection requests that have been queued by a server.

- Managing the queue limit

  The queue limit specifies the maximum number of concurrent queue entries. The server manager can change the default queue limit without affecting existing queue entries. The maximum queue limit and default queue limit is server specific. Reducing the queue limit can increase the number of sessions a server supports or the number of nodes it lists in its service-node directory.

- Disabling further queuing

  Disabling queuing prevents new entries for a specified service from being added to the queue but does not affect existing queue entries. With queuing disabled, when all ports offering the service are busy, the server rejects all further connection requests for that service. However, queuing is not disabled for host-initiated requests that specify a port name rather than a service name. Since these requests are not associated with a service, the server continues to place them in the queue. Disabling queuing for a service can help reduce memory use on a server.

- Removing entries from the queue

  A server manager, application user, or system manager can remove an entry for a host-initiated request from the connection queue.

5.14 Managing Communications on a LAT Network

The network manager coordinates the activities of service nodes and servers. Normally, the default values of timers and of retransmit limits are best. However, a LAT network manager can ask system and server managers to modify these defaults. This section provides some guidelines to help the network manager decide whether the defaults need to be changed.

5.14.1 Selecting the Value of the Multicast Timer

The value for a multicast timer ranges from 10 to 255 seconds. Normally, a 60-second timer provides timely service notification to users with minimal overhead. However, if a network has a very large number of service nodes, a service-node manager can increase the value of the multicast timer to reduce network traffic. Increasing this value increases the time before the server can detect that a service node is no longer offering services. Also, server users have to wait longer for services to become available after the server is rebooted or after recovering from a network problem. In addition, the accuracy of the service ratings used in load balancing by servers decreases as the value of the multicast timer increases.

On the other hand, assigning a low value to the multicast timer, consumes more network resources. The server must process multicast messages and update its service and service-node directories more frequently.

5.14.2 Selecting the Value of the Circuit Timer

The value of the circuit timer is important for balancing fast response time and network utilization against optimal service node performance. For normal interactive functions the circuit timer should be set at the default value of 80 milliseconds, which provides a good balance between terminal response time and service node performance. If the value is increased, the LAT protocol overhead decreases on the service node and on the network. However, any gain achieved by setting the circuit timer higher must be weighed against the lengthened response time at terminals.

The circuit timer value also affects file transfers. As the circuit timer is reduced, the port buffers are less likely to be filled between virtual circuit messages.

5.14.3 Selecting the Value of the Retransmit Limit

The retransmit limit defines the number of times a message is retransmitted before the virtual circuit is declared down and any current attempt to establish a session is timed out. Failover to another service node occurs only after a circuit times out.

Retransmission of messages ensures reliable communications on the LAN. Therefore, if traffic load is heavy or if the network is experiencing noise problems, it might help to make the value higher than the default. On the other hand, if rapid error detection is important, a server manager might specify a lower value.

5.14.4 Selecting the Value of the Keepalive Timer

The value a server manager sets for the keepalive timer is a trade-off between fast detection of a downed circuit and unnecessary traffic flow on the network. The default value of 20 seconds represents a good compromise value. However, the value can be increased to reduce traffic on heavily used networks.

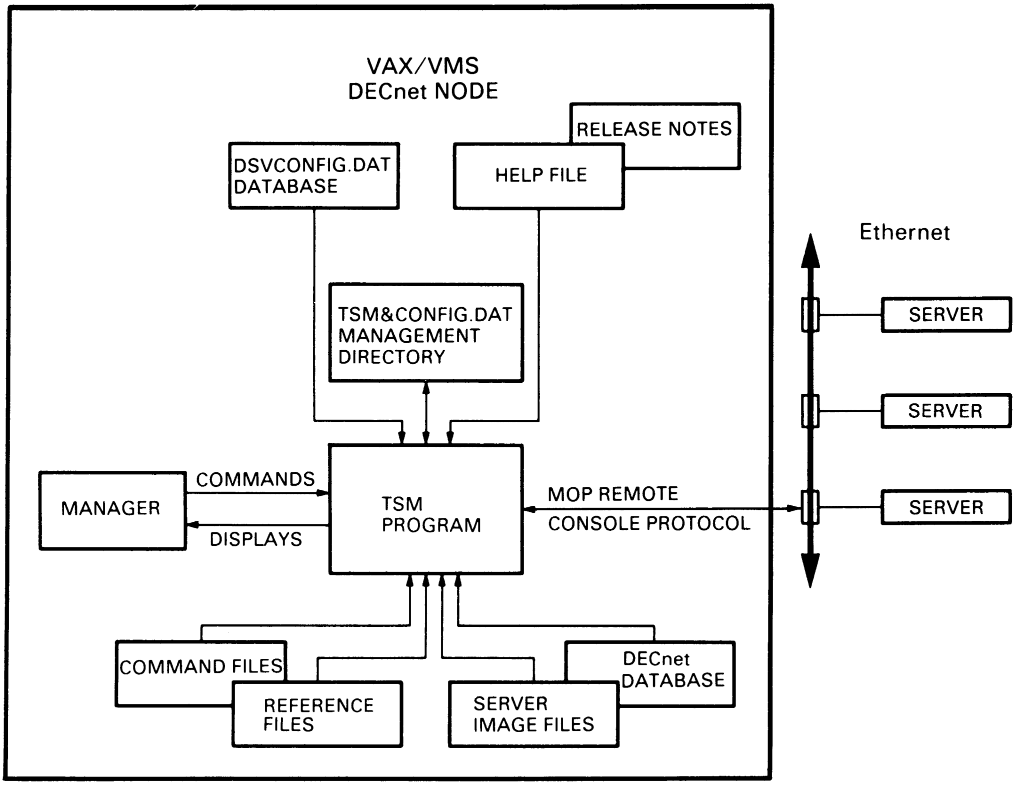
5.15 Using the Terminal Server Manager (TSM) Product

The Terminal Server Manager (TSM) software is an optional network management product that runs on VAX/VMS systems that implement DECnet-VAX software, including VAXcluster systems. TSM provides the capabilities of TSC and RCF and much more. TSM provides centralization and increased flexibility in configuring, monitoring, and controlling Digital's family of terminal servers. TSM utilizes the Maintenance Operations Protocol (MOP) Remote Console Facility to provide communications with terminal servers on the LAT network. Network/system managers can perform the following operations using TSM:

- Management directory access
- Database manipulation
- Server selection
- Initial configuration
- Wildcard operations
- Fault management
- Command file support
- Reference file support

Figure 5-7 illustrates the relationship of TSM to load hosts and terminal servers on the LAT network.

Figure 5-7: TSM Program and Terminal Servers on a LAT Network



LKG-1208-87

5.15.1 Management Directory

TSM provides a directory that contains the name, Ethernet address, terminal server type, and other fields for each server to be managed. By using the TSM directory, the network/system manager needs only to refer to the name of the server in order to gain access.

5.15.2 Database Manipulation

TSM can access and manipulate the database associated with all servers within its jurisdiction including their characteristics and parameters. The software provides an alternative, with additional features, to the Terminal Server Configurator (TSC). TSM manages the volatile characteristics for all terminal server types. TSM manages permanent NVRAM characteristics for the DECserver 100 system, DECserver 200 system, and MUXserver 100 system. For the Ethernet Communications Server Terminal Server and DECserver 500 system, TSM modifies the values contained within the server's permanent image file on the load host.

5.15.3 Server Selection

TSM provides a USE command that allows the selection of a server to be managed or of a directory to be referenced.

5.15.4 Initial Configuration

Command files can be maintained outside TSM (for example, using a VAX /VMS editor). This command file support allows values of characteristics to be stored on the host, effectively creating a copy of the permanent database that can be used to configure new servers. Using TSM command files simplifies swapping one server unit for another.

5.15.5 Wildcard Operations

Through the execution of nested command files, terminal server operations can be completed on more than one server at a time. Provision is made for "wildcard" operations (allowing limited management partitions); the TSM user can activate a string of DEFINE commands with a single "@" command.

5.15.6 Fault Management

Troubleshooting within the terminal server network is enhanced by TSM, which can centrally read traffic status and counters in addition to error status and counters.

A simple connectivity test to a specific server or all servers listed in the directory provides an informational message to be displayed for each server not responding. TSM allows the network/system manager to check periodically for nonfunctioning terminal servers. The connectivity test can be performed from within a batch job, which allows the test to be administered as a specific time; for example, a server manager can set up a batch job that runs automatically every morning before business hours.

5.15.7 Command File Support

TSM provides network managers with the flexibility of using Digital Command Language (DCL) command file procedures. A server's output can be directed to a file. These procedures can be used two ways:

- Command procedures invoked from the DCL prompt
- Command procedures called from within the TSM program

Using command procedures allows the network manager to selectively analyze certain server characteristics. These characteristics can include important day-to-day monitoring of network servers and services.

Index

A

Announcements characteristic
  definition of, 5-21
Applications port
  definition of, 5-23
Architecture
  see DNA architecture
  see LAT architecture
Attention slot
  definition of, 3-14
Authorized groups
  definition of, 5-10
Autobaud facility
  description of, 5-9
Automatic failover
  definition of, 4-13

B

Balanced virtual circuit
  definition of, 4-14
Buffering
  receive buffers
    flow control for, 3-14
    quantity of, 4-12
  transmit buffers
    flow control, 3-12
    quantity of, 4-12
    use of, 4-9
Buffering (cont'd.)
  using credits, 3-14

C

Circuit timer
  definition of, 4-7
  operation of, 4-9
  selecting value for, 5-26
Command interface
  of servers, 2-9
Command message
  and connection queue, 3-12
  and host-initiated requests, 3-12
  definition of, 3-11
Command procedures
  DSVCONFIG.COM, 5-5
Communications processes
  overview of, 4-1
Configuration
  of load host's node database, 5-6
  using the TSC, 2-4
Connection queue
  accessing, 3-10
  definition of, 4-20
  earliest entry, 4-20
  maintenance by server, 4-20
  management of, 5-25
  operation of, 4-20

Connections characteristic
  definition of, 5-21
Console port
  definition of, 5-6
Credits
  definition of, 3-14
  explanation of, 3-14
  operation of, 3-15
CTERM protocol
  and PC LAT, 2-13
Customization
  of load host's node database, 5-6
Customizing load host's node database
  definition of, 5-5

D

Data
  in-band, 3-14
  out-of-band, 3-14
Data-A slot
  definition of, 3-14
Data-B slot
  definition of, 3-14
Datagrams
  definition of, 3-4
Data Link layer
  definition of, 3-4
Data transparency
  definition of, 2-12
Dedicated ports
  explanation of, 2-7
Dedicated service
  definition of, 2-7
DNA architecture
  Data Link layer, 3-4
  layers used, 3-4
  Physical Link layer, 3-4
Down-line loading, 2-2
DSVCONFIG, 5-5
Dual-purpose servers
Dual-purpose servers (cont'd.)
  see also Service nodes
    dual purpose
  local-service characteristics, 5-21
  overview of, 2-15
  services offered by, 2-16
Dynamic access
  explanation of, 2-7

E

Empty run message
  definition of, 4-15
Emulate
  definition of, 2-12
Ethernet hardware
  overview of, 1-23
Ethernet protocol
  Data Link layer, 3-4
  Physical Link layer, 3-4

F

Failover
  definition of, 4-13
File transfer support, 2-12
Flow control
  credits, 3-14
  in Service Class 1 module
    overview of, 3-12
  in Slot layer
    overview of, 3-14
  in Virtual Circuit layer
    overview of, 3-19
Fully interactive ports, 2-7

G

General-purpose service node
  see Service nodes

Groups
assigning, 5-13
definition of, 1-23
example of, 5-14
server ports, 5-10
service groups, 5-13
service-specific approach to, 5-13
setting up, 5-10

H

Help
overview of, 2-11
Host-initiated requests
configuring service node for, 5-22
definition of, 1-12
in session establishment, 4-6
process
definition of, 4-17
making requests, 4-17
processing requests, 4-18
translating a server name, 4-17
using local services for, 5-24

I

In-band transmission
data, 3-14
flow control, 3-12
Interfaces
on general-purpose service nodes, 2-15
on servers, 2-9
Interfacing between LAT layers, 3-6

K

Keepalive timer
definition of, 4-14
selecting value for, 5-27

L

LAT
product support, 1-13
service nodes, 2-14
LAT architecture
definition of, 1-2
explanation of, 3-1
fundamental features of, 3-2
groups
example of, 5-14
interfacing between layers, 3-6
layering in, 3-3
messages
service class, 3-9
Service Class layer, 3-8
definition of, 3-3
Slot layer, 3-13
underlying elements of, 3-2
Virtual Circuit layer, 3-17
definition of, 3-3
LAT Control Program (LCP)
definition of, 2-15
LAT network manager
definition of, 1-27
LAT Networks
benefits, 1-19
LAT networks
accessing other networks from, 1-2
configuration guidelines, 5-25
configurations of, 1-21
definition of, 1-1
Ethernet hardware, 1-23
groups
example of, 5-14
management of
configuring server ports, 5-8
customizing load host's node database, 5-5
customizing server characteristics, 5-8

- LAT networks
  - management of (cont'd.)
    - host-initiated requests, 5-22, 5-24
    - installing server hardware, 5-2
    - installing server software, 5-4
    - managing communications, 5-25
    - planning physical configuration, 5-2
    - services, 5-20, 5-21
    - setting up service nodes, 5-11
    - using the RCF, 5-6
    - using the TSM, 5-27
  - partitioning with groups, 1-23
- LAT nodes
  - definition of, 1-8
- LAT protocol
  - communications processes, 4-1
  - definition of, 1-2, 3-1
  - features, 3-2
  - groups
    - example of, 5-14
  - session control
    - definition of, 3-3
  - transport mechanism
    - explanation of, 3-3
  - VAX/VMS implementation, 3-20
- LAT software
  - definition of, 1-3
  - overview of, 1-7
- LATSYM, 3-20
- LCP
  - definition of, 2-15
  - on VMS systems, 3-20
- Load balancing
  - definition of, 4-6
  - effect of multicast timer on, 5-26
  - for VAXclusters, 1-20
- Load host's node database
  - definition of, 5-5
- Load hosts
  - definition of, 5-4
- Local mode
  - definition of, 2-7
- Local service
  - password, 5-22
- Local-service characteristics
  - overview of, 2-4
- Local service node
  - *See also* Dual-purpose server
  - definition of, 1-16, 2-15
  - use of service rating for, 4-6
- Local services
  - advantages for remote-access ports, 5-24
  - characteristics
    - modifying values of, 5-21
  - definition of, 2-15
  - setting up
    - prerequisites for, 5-20
- Log-in database
  - definition of, 2-6
- Log-in password
  - definition of, 2-8

M

- Maintenance Operations Protocol
  - on servers, 5-6
- Maintenance password
  - definition of, 2-8
- Messages
  - command
    - definition of, 3-11
    - for host-initiated requests, 3-12
    - for managing queued requests, 3-12
  - response information
    - definition of, 3-11
  - service announcement
    - definition of, 3-10
  - service class
    - service announcement, 4-3

Messages (cont'd.)
  solicit information
    definition of, 3-10
  status
    definition of, 3-11
  virtual circuit
    empty run, 4-14
    overview of, 3-18
    retransmission of, 4-14
    run, 3-18, 4-15
    start, 3-18
    stop, 3-19
Modem control
  on servers, 1-10
  use with non-LAT hosts, 2-17
Modems
  dial-in/dial-out modems
    definition of, 2-16
  dial-in modems
    definition of, 2-16
  dial-out modems
    definition of, 2-16
  overview of, 2-16
MOP
  on servers, 5-6
Multicasting
  definition of, 3-4
Multicast messages
  service announcement message
    definition of, 3-10
  service-announcement messages
    use of, 4-3
  solicit-information message
    definition of, 3-10
  solicit-information messages
    use of, 4-17
Multicast timer
  definition of, 4-3
  selecting value for, 5-26
Multiple sessions
  explanation of, 1-5
Multiplexing
  benefits of, 1-20

N

Network terminal services (NTS)
  overview of, 2-13
Node database, 5-6
  of load host, 5-5
Non-LAT hosts
  definition of, 1-11
  security, 2-17
Nonprivileged commands
  overview of, 2-10
NTS
  overview of, 2-13

O

On-line help
  overview of, 2-11
Operational database
  definition of, 2-6
Operational value
  definition of, 2-6
Out-of-band transmission
  data, 3-14
  flow control, 3-12

P

Password characteristic
  definition of, 5-21
Passwords
  overview of, 2-8
  service password
    use of, 5-22
PC LAT
  explanation of, 2-12
Peer-to-peer communication

Peer-to-peer communication (cont'd.)
  definition of, 3-2
Permanent database
  customizing with the TSC, 2-4
  definition of, 2-4
Permanent value
  definition of, 2-4
Personal computers
  file transfers, 2-12
  LAT implementations on, 2-12
  on LAT networks, 2-11
  overview of, 2-16
Physical Link layer
  definition of, 3-4
Port characteristics
  overview of, 2-3
Port configurations
  of server ports, 2-7
Port groups
  explanation of, 1-23
Printers
  overview of, 2-16
Privileged commands
  overview of, 2-10
Privileged password
  definition of, 2-9
Protocols
  CTERM, 2-13
  DNA
    layers used, 3-4
  LAT
    definition of, 3-1
  MOP
    on servers, 5-6

Q

QIO, 3-20
Queue
  see Connection queue
  disabling for a service, 5-25
Queue (cont'd.)
  queue limit, 5-25
  removing entries from, 5-25
Queue characteristic
  definition of, 5-22

R

RCF
  overview of, 5-6
  use by TSM, 5-27
Receive buffers
  flow control for, 3-14
  quantity of, 4-12
  with balanced circuit, 4-14
Reject slot
  definition of, 3-13
Remote access
  explanation of, 2-8
Remote Console Facility (RCF)
  overview of, 5-6
  use by TSM, 5-27
Remote device
  definition of, 1-16, 5-23
Remote management port
  definition of, 5-6
Request number
  definition of, 3-11
Response-information message
  definition of, 3-11
Retransmit limit
  selecting value for, 4-14, 5-26
Run message
  definition of, 3-18

S

Security
  non-LAT Hosts, 2-17
  specifying serverwide passwords, 5-8

Server
  as a local service node
    overview of, 5-12
  port configurations, 2-7
Server characteristics
  overview of, 2-3
Server distribution software
  definition of, 5-4
Server groups
  explanation of, 1-23
Server manager
  definition of, 1-27
Servers
  security
  accessed using the RCF, 5-6
  as service nodes, 1-16
  characteristics of
    local-service characteristics, 2-4
    port characteristics, 2-3
    server characteristics, 2-3
  circuit timer
    selecting value for, 5-26
  command interface of, 2-9
  configuring ports
    for applications devices, 5-11
    for interactive devices, 5-9
  databases of
    log-in database, 2-6
    operational database, 2-6
    permanent database, 2-4
  definition of, 1-8
  devices supported by all, 1-13
  devices supported by dual-purpose, 1-16
  dual purpose
    overview of, 2-15
    services offered by, 2-16
  introduction, 1-10
  keepalive timer
    selecting value for, 5-27
  modem control on, 1-10
Servers (cont'd.)
  passwords
    lock password, 2-8
    log-in password, 2-8
    maintenance password, 2-8
    privileged password, 2-9
    service password, 2-9
  products, 1-10
  remote management port, 5-6
  retransmit limit
    selecting value for, 5-26
  used with VAXclusters, 4-6
Server user
  definition of, 1-11
Service
  name translation, 4-6
Service-announcement message
  definition of, 4-3
Service-announcement messages
  overview of, 3-10
Service-announcement process
  discussion of, 4-2
Service Class 1 module
  example of operation, 4-10
  functions, 3-9
  messages, 3-9
  multicast timer, 4-3
Service classes
  introduction, 3-8
Service Class layer
  definition of, 3-3
  flow control, 3-12
Service-class messages
  definition of, 3-9
Service directory
  description of, 4-3
Service groups
  definition of, 5-13
Service names
  explanation of, 1-3
  setting up, 5-20

Service names (cont'd.)
- using with host-initiated requests, 5-24

Service-node directory
- description of, 4-3

Service-node groups
- definition of, 5-13
- explanation of, 1-23

Service-node managers
- definition of, 5-11

Service nodes
- definition of, 1-8
- dual purpose
  - definition of, 1-8
  - devices supported, 1-17
  - local-service characteristics of, 5-21
  - managing services on, 5-21
  - node name, 5-12
  - overview of, 2-15
  - products, 1-16
  - services offered by, 2-16
- general purpose
  - definition of, 1-8
  - host-initiated-request capability, 1-16
  - host-initiated requests, 5-22
  - managing services on, 5-21
  - node identification string, 5-12
  - node name, 5-12
  - products, 1-14
  - setting up, 5-12
  - VAX/VMS systems, 3-20, 5-22
- general-purpose, 2-14
- groups, 5-13
- multicast timer
  - selecting value for, 5-26
- node name, 5-12
- overview of, 1-13

Service password
- definition of, 2-9

Service rating
- definition of, 4-2
- in load balancing, 4-6

Service ratings
- effect of multicast timer on, 5-26
- explanation of, 4-2
- maintenance of, 4-2

Services
- assigning names, 5-24
- definition of, 1-3
- local-service passwords, 5-22
- management of, 5-21
- service-announcement messages, 4-3
- service names
  - assigning, 5-20
  - setting up, 5-20
- service ratings
  - assigning static values, 5-21
  - explanation of, 4-2
- services names of, 1-3

Service software
- LAT architecture V5.1, 5-22

Service-specific approach to groups, 5-13

Session control
- definition of, 3-3

Session-establishment process
- explanation of, 4-4

Session management
- discussion of, 4-12

Sessions
- data transparency of, 2-12
- definition of, 1-4
- disconnection of, 4-13
- establishment of, 4-4
- management of, 4-12
- passing user data, 4-12
- performing failover for, 4-13
- starting of, 4-9

SETHOST utility, 2-13

Slot

Slot (cont'd.)
  definition of, 3-13
Slot layer
  credit
    definition of, 3-14
  credits, 3-14
  data slot, 3-14
  data slots, 3-14
  definition of, 3-3
  flow control, 3-14
  functional description, 3-13
  introduction, 3-13
  operation of credits, 3-15
  reject slot, 3-13
  start slot, 3-13
  stop slot, 3-14
Slots
  data slots, 3-14
  reject slot, 3-13
  start slot, 3-13
  stop, 3-14
SNA networks
  accessing from LAT servers, 1-24
Solicit-information message
  definition of, 3-10
Start message
  definition of, 3-18
Start slot
  definition of, 3-13
Start slots
  responding to, 4-10
  sending of, 4-9
Status levels
  explanation of, 2-9
Status message
  and connection queue, 3-12
  and host-initiated requests, 3-12
  definition of, 3-11
Stop message
  definition of, 3-19
Stop slot
  definition of, 3-14
System manager
  definition of, 1-27
  tasks, 5-11

T

Terminal emulation
  definition of, 2-12
Terminal server
  definition of, 2-1
Terminal Server Configurator
  definition of, 2-4
Terminal Server Manager (TSM)
  see also TSM
  overview of, 5-27
Terminal servers
  see Servers
Transmit buffers
  flow control for, 3-12
  quantity of, 4-12
  use of, 4-9
Transport mechanism
  explanation of, 3-3
TSC
  definition of, 2-4
TSM
  command files, 5-30
  database manipulation, 5-29
  fault management, 5-29
  initial configuration, 5-29
  management directory, 5-28
  overview of, 5-27
  server selection, 5-29
  Wildcard operations, 5-29

U

User commands
  overview of, 2-10
User data

User data (cont'd.)
  passing for sessions, 4-12
User-initiated requests
  definition of, 1-12
  in session establishment, 4-4
User interface
  of servers, 2-9
User-oriented characteristics, 5-9

V

VAX/VMS system
  LAT implementation on, 3-20
VAX/VMS systems
  setting up host-initiated requests, 5-22
VAXclusters
  automatic failover, 4-13
  load balancing for, 1-20, 4-6
Virtual Circuit layer
  balanced circuits, 4-14
  definition of, 3-3, 3-17
  flow control, 3-19
  functions, 3-17
  introduction, 3-17
  messages, 3-18, 4-15
  run message, 3-18
  run messages
    retransmission of, 4-14
  start message, 3-18
  stop message, 3-19
    definition of, 3-19
  timers
    circuit timer, 4-7
    keepalive timer, 4-14
Virtual-circuit maintenance
  discussion of, 4-14
Virtual-circuit messages
  see Virtual Circuit layer
Virtual circuits
  definition of, 3-17

Virtual circuits (cont'd.)
  establishment of, 4-7
  Run message, 3-18
  use of, 3-17

Local Area Transport (LAT)
Network Concepts
AA–LD84A–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
Networks and
Communications Publications
550 King Street
Littleton, MA 01460–1289

DO NOT CUT – FOLD HERE