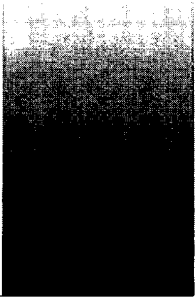


Internetworking Technology Overview





Internetworking Technology Overview

Corporate Headquarters:
1525 O'Brien Drive
Menlo Park, California 94026 USA
1 (415) 326-1941
1-800-553-NETS

Customer Order Number: DOC-ITO13
Text Part Number: 78-1070-01



The products and specifications, configurations, and other technical information regarding the products contained in this manual are subject to change without notice. All statements, technical information, and recommendations contained in this manual are believed to be accurate and reliable but are presented without warranty of any kind, express or implied, and users must take full responsibility for their application of any products specified in this manual. THIS MANUAL IS PROVIDED "AS IS" WITH ALL FAULTS. CISCO DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING THOSE OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Some states do not allow limitation or exclusion of liability for consequential or incidental damages or limitation on how long implied warranties last, so the above limitations or exclusions may not apply to you. This warranty gives Customers specific legal rights, and you may also have other rights that vary from state to state.

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. This equipment has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright (c) 1981 Regents of the University of California.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Point-to-Point Protocol. Copyright (c) 1989 Carnegie Mellon University. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by Carnegie Mellon University. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Notice of Restricted Rights:

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR § 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS § 252.227-7013.

The information in this manual is subject to change without notice.

APPI, ciscoBus, Cisco Systems, CiscoWorks, CxBus, Netscape, The Packet, and SMARTnet are trademarks, and the Cisco logo is a registered trademark of Cisco Systems, Inc. All other products or services mentioned in this document are the trademarks, service marks, registered trademarks, or registered service marks of their respective owners.

Internetworking Technology Overview

Copyright © 1993, Cisco Systems, Inc.

All rights reserved. Printed in USA.

Table of Contents

About This Manual xvii

- Document Objectives xvii
- Audience xvii
- Document Organization xviii
- Document Conventions xviii
- Related Cisco Documentation xix

Service and Support xxi

- Warranty Information xxi
- Maintenance Agreements xxi
- Customer Support xxii

Obtaining Additional Information xxiii

- Ordering Additional Cisco Publications xxiii
- Obtaining Cisco Technical Information Electronically xxiii
- Obtaining Information from Other Sources xxiv
 - Obtaining RFCs xxiv
 - Obtaining Technical Standards xxv

Part 1 Internetworking Basics

Chapter 1 Introduction to Internetworking 1-1

- Introduction 1-1
- OSI Reference Model 1-1
 - Hierarchical Communication 1-1
 - Information Formats 1-3
 - Compatibility Issues 1-4
 - OSI Layers 1-4
- Important Terms and Concepts 1-6
 - Addressing 1-6
 - Frames, Packets, and Messages 1-7
- Key Organizations 1-7

Chapter 2	<i>Routing Basics 2-1</i>
	Background 2-1
	Routing Components 2-1
	Path Determination 2-1
	Switching 2-2
	Routing Algorithms 2-4
	Design Goals 2-4
	Types 2-6
	Metrics 2-8
	Routed vs. Routing Protocols 2-9
Chapter 3	<i>Bridging Basics 3-1</i>
	Background 3-1
	Internetworking Device Comparison 3-1
	Technology Basics 3-2
	Types of Bridges 3-3
Chapter 4	<i>Network Management Basics 4-1</i>
	Background 4-1
	Network Management Architecture 4-1
	The ISO Network Management Model 4-2
	Performance Management 4-3
	Configuration Management 4-3
	Accounting Management 4-4
	Fault Management 4-4
	Security Management 4-4
Part 2	<i>Media-Access Technologies</i>
Chapter 5	<i>Ethernet/IEEE 802.3 5-1</i>
	Background 5-1
	Ethernet/IEEE 802.3 Comparison 5-1
	Physical Connections 5-2
	Frame Formats 5-3
Chapter 6	<i>Token Ring/IEEE 802.5 6-1</i>
	Background 6-1
	Token Ring/IEEE 802.5 Comparison 6-1

	Token Passing 6-2
	Physical Connections 6-3
	Priority System 6-4
	Fault Management Mechanisms 6-4
	Frame Format 6-5
Chapter 7	<i>FDDI 7-1</i>
	Background 7-1
	Technology Basics 7-1
	FDDI Specifications 7-2
	Physical Connections 7-3
	Traffic Types 7-4
	Fault-Tolerant Features 7-4
	Frame Format 7-8
Chapter 8	<i>UltraNet 8-1</i>
	Background 8-1
	Technology Basics 8-1
	UltraNet Components 8-3
	UltraNet Hub 8-3
	UltraNet Host Software 8-3
	Ultra Network Manager 8-3
	Network Processors 8-4
	Link Adapters 8-4
Chapter 9	<i>HSSI 9-1</i>
	Background 9-1
	Technology Basics 9-2
Chapter 10	<i>PPP 10-1</i>
	Background 10-1
	PPP Components 10-1
	General Operation 10-2
	Physical-Layer Requirements 10-2
	The PPP Data-Link Layer 10-2
	The PPP Link Control Protocol (LCP) 10-3

Chapter 11	ISDN 11-1
	Background 11-1
	Components 11-1
	Services 11-3
	Layer 1 11-3
	Layer 2 11-4
	Layer 3 11-5
Part 3	Packet-Switching Technologies
Chapter 12	SDLC and Derivatives 12-1
	Background 12-1
	Technology Basics 12-1
	Frame Format 12-2
	Derivative Protocols 12-4
	HDLC 12-4
	LAPB 12-5
	IEEE 802.2 12-5
Chapter 13	X.25 13-1
	Background 13-1
	Technology Basics 13-1
	Frame Format 13-4
	Layer 3 13-4
	Layer 2 13-5
	Layer 1 13-6
Chapter 14	Frame Relay 14-1
	Background 14-1
	Technology Basics 14-1
	LMI Extensions 14-2
	Frame Format 14-3
	LMI Message Format 14-5
	Global Addressing 14-5
	Multicasting 14-6
	Network Implementation 14-7

Chapter 15	SMDS 15-1
	Background 15-1
	Technology Basics 15-1
	Addressing 15-2
	Access Classes 15-3
	SMDS Interface Protocol (SIP) 15-3
	CPE Configurations 15-4
	SIP Levels 15-5
	Network Implementation 15-9
Part 4	Routed Protocols
Chapter 16	AppleTalk 16-1
	Background 16-1
	Technology Basics 16-1
	Media Access 16-2
	Network Layer 16-3
	Protocol Address Assignment 16-3
	Network Entities 16-4
	Datagram Delivery Protocol (DDP) 16-5
	Routing Table Maintenance Protocol (RTMP) 16-6
	Transport Layer 16-7
	AppleTalk Transaction Protocol (ATP) 16-8
	AppleTalk Data Stream Protocol (ADSP) 16-8
	Upper-Layer Protocols 16-8
Chapter 17	DECnet 17-1
	Background 17-1
	Digital Network Architecture (DNA) 17-1
	Media Access 17-2
	Network Layer 17-2
	DECnet Phase IV Routing Frame Format 17-3
	Addressing 17-4
	Routing Levels 17-5
	Transport Layer 17-6
	Upper-Layer Protocols 17-6

Chapter 18	<i>Internet Protocols 18-1</i>
	Background 18-1
	Network Layer 18-2
	Addressing 18-4
	Internet Routing 18-6
	ICMP 18-8
	Transport Layer 18-8
	Transmission Control Protocol (TCP) 18-8
	User Datagram Protocol (UDP) 18-10
	Upper-Layer Protocols 18-10
Chapter 19	<i>NetWare Protocols 19-1</i>
	Background 19-1
	Technology Basics 19-1
	Media Access 19-2
	Network Layer 19-3
	Transport Layer 19-5
	Upper-Layer Protocols 19-5
Chapter 20	<i>OSI Protocols 20-1</i>
	Background 20-1
	Technology Basics 20-1
	Media Access 20-2
	Network Layer 20-2
	Connectionless Service 20-3
	Connection-Oriented Service 20-3
	Addressing 20-4
	Transport Layer 20-5
	Upper-Layer Protocols 20-6
	Session Layer 20-6
	Presentation Layer 20-6
	Application Layer 20-6
Chapter 21	<i>Banyan VINES 21-1</i>
	Background 21-1
	Technology Basics 21-1
	Media Access 21-2

	Network Layer 21-2
	VINES Internetwork Protocol (VIP) 21-2
	Routing Update Protocol (RTP) 21-6
	Address Resolution Protocol (ARP) 21-7
	Internet Control Protocol (ICP) 21-7
	Transport Layer 21-8
	Upper-Layer Protocols 21-8
Chapter 22	XNS 22-1
	Background 22-1
	Technology Basics 22-1
	Media Access 22-2
	Network Layer 22-3
	Transport Layer 22-4
	Upper-Layer Protocols 22-5
Part 5	Routing Protocols
Chapter 23	RIP 23-1
	Background 23-1
	Routing Table Format 23-1
	Packet Format (IP Implementations) 23-2
	Stability Features 23-4
	Hop Count Limit 23-4
	Hold-Downs 23-4
	Split Horizons 23-5
	Poison Reverse Updates 23-5
Chapter 24	IGRP 24-1
	Background 24-1
	Technology 24-1
	Packet Format 24-2
	Stability Features 24-4
	Hold-Downs 24-4
	Split Horizons 24-4
	Poison Reverse Updates 24-5
	Timers 24-5

Chapter 25	OSPF 25-1
	Background 25-1
	Technology Basics 25-2
	Routing Hierarchy 25-2
	The SPF Algorithm 25-4
	Packet Format 25-5
	Additional OSPF Features 25-6
Chapter 26	EGP 26-1
	Background 26-1
	Technology Basics 26-1
	Packet Format 26-2
	Message Types 26-3
	Neighbor Acquisition 26-3
	Neighbor Reachability 26-3
	Poll 26-3
	Routing Update 26-4
	Error 26-4
Chapter 27	BGP 27-1
	Background 27-1
	Technology Basics 27-1
	Packet Format 27-2
	Messages 27-2
	Open 27-3
	Update 27-3
	Keepalive 27-3
	Notification 27-4
Chapter 28	OSI Routing 28-1
	Background 28-1
	Terminology 28-1
	ES-IS 28-2
	IS-IS 28-4
	Routing Hierarchy 28-4
	Inter-ES Communication 28-4
	Metrics 28-5
	Packet Format 28-5

Integrated IS-IS 28-6
Inter-Domain Routing Protocol (IDRP) 28-7

Part 6 ***Bridging Technologies***

Chapter 29 ***Transparent Bridging 29-1***

Background 29-1
Technology Basics 29-1
Bridging Loops 29-2
Spanning-Tree Algorithm (STA) 29-3
Frame Format 29-5

Chapter 30 ***Source-Route Bridging 30-1***

Background 30-1
SRB Algorithm 30-1
Frame Format 30-3

Chapter 31 ***Mixed-Media Bridging 31-1***

Background 31-1
Technology Basics 31-2
 Translation Challenges 31-2
Translational Bridging (TLB) 31-3
Source-Route Transparent (SRT) Bridging 31-4

Part 7 ***Network Management***

Chapter 32 ***SNMP 32-1***

Background 32-1
Technology Basics 32-2
 Management Model 32-2
 Command Types 32-3
 Data Representation Differences 32-3
 Management Database 32-4
 Operations 32-5
Message Format 32-5

Chapter 33 ***IBM Network Management 33-1***
 Background 33-1
 Functional Areas of Management 33-1
 Configuration Management 33-2
 Performance and Accounting Management 33-2
 Problem Management 33-3
 Operations Management 33-3
 Change Management 33-4
 Principal Management Architectures and Platforms 33-4
 The Open Network Management (ONA) Framework 33-4
 SystemView 33-5
 NetView 33-5
 LAN Network Manager 33-6
 SNMP 33-6

Appendix A ***References and Recommended Reading A-1***

Index

List of Figures

- Figure 1-1* Communication Between Two Computer Systems 1-2
- Figure 1-2* Relationship Between Adjacent Layers in a Single System 1-3
- Figure 1-3* Headers and Data 1-4
- Figure 2-1* Destination/Next Hop Routing Table 2-2
- Figure 2-2* Switching Process 2-3
- Figure 2-3* Slow Convergence and Routing Loops 2-5
- Figure 3-1* Internetworking Product Functionality 3-2
- Figure 3-2* Local and Remote Bridging 3-3
- Figure 3-3* IEEE 802.3/IEEE 802.5 Bridging 3-4
- Figure 4-1* Typical Network Management Architecture 4-2
- Figure 5-1* IEEE 802.3 Physical-Layer Name Components 5-2
- Figure 5-2* Ethernet V2.0 and IEEE 802.3 Physical Characteristics 5-2
- Figure 5-3* Ethernet and IEEE 802.3 Frame Formats 5-3
- Figure 6-1* IBM Token Ring Network/IEEE 802.5 Comparison 6-2
- Figure 6-2* IBM Token Ring Network Physical Connections 6-3
- Figure 6-3* IEEE 802.5/Token Ring Frame Formats 6-5
- Figure 7-1* FDDI Standards 7-2
- Figure 7-2* FDDI Nodes: DAS, SAS, and Concentrator 7-3
- Figure 7-3* FDDI DAS Ports 7-3
- Figure 7-4* Station Failure, Ring Recovery Configuration 7-5
- Figure 7-5* Failed Wiring, Ring Recovery Configuration 7-6
- Figure 7-6* Use of Optical Bypass Switch 7-7
- Figure 7-7* FDDI Frame Format 7-8
- Figure 8-1* UltraNet and the OSI Reference Model 8-1
- Figure 8-2* UltraNet Network System 8-2
- Figure 9-1* HSSI Technical Characteristics 9-2
- Figure 9-2* HSSI's Four Loopback Tests 9-3

Figure 10-1 PPP Frame Format 10-2

Figure 11-1 Sample ISDN Configuration 11-2

Figure 11-2 ISDN Physical-Layer Frame Formats 11-4

Figure 11-3 LAPD Frame Format 11-5

Figure 11-4 ISDN Circuit-Switched Call Stages 11-6

Figure 12-1 SDLC Frame Format 12-2

Figure 12-2 Typical SDLC-Based Network Configuration 12-4

Figure 13-1 X.25 Model 13-2

Figure 13-2 X.25 and the OSI Reference Model 13-3

Figure 13-3 X.25 Frame 13-4

Figure 13-4 X.121 Address Format 13-5

Figure 13-5 LAPB Frame 13-6

Figure 14-1 Frame Relay Frame 14-3

Figure 14-2 Frame Relay Addressing 14-4

Figure 14-3 LMI Message Format 14-5

Figure 14-4 Global Addressing Exchange 14-6

Figure 14-5 Hybrid Frame Relay Network 14-7

Figure 15-1 SMDS Internetworking Scenario 15-2

Figure 15-2 Single-CPE and Multi-CPE Configurations 15-4

Figure 15-3 Encapsulation of User Information by SIP Levels 15-5

Figure 15-4 SIP Level 3 PDU 15-6

Figure 15-5 SIP Level 2 PDU 15-7

Figure 15-6 Segment Type Values 15-7

Figure 16-1 AppleTalk and the OSI Reference Model 16-2

Figure 16-2 AppleTalk Address Selection Process 16-3

Figure 16-3 AppleTalk Entities 16-5

Figure 16-4 Sample AppleTalk Routing Table 16-6

Figure 16-5 Sample AppleTalk ZIT 16-7

Figure 17-1 DNA and the OSI Reference Model 17-2

Figure 17-2 DNA Phase IV Routing Layer Header 17-3

Figure 17-3 DECnet Phase IV Routing Protocol Cost Calculation 17-4

Figure 17-4 DECnet Addresses 17-4

Figure 17-5 DECnet Level 1 and Level 2 Routers 17-5

Figure 18-1 Internet Protocol Suite and the OSI Reference Model 18-2

Figure 18-2 IP Packet Format 18-3

Figure 18-3 Class A, B, and C Address Formats 18-5

Figure 18-4 Subnet Addresses 18-5

Figure 18-5 Sample Subnet Mask 18-6

Figure 18-6 Internet Architecture 18-7

Figure 18-7 IP Routing Table 18-7

Figure 18-8 TCP Packet Format 18-9

Figure 18-9 Internet Protocol/Application Mapping 18-10

Figure 19-1 NetWare and the OSI Reference Model 19-2

Figure 19-2 IPX Packet Format 19-3

Figure 19-3 Ethernet, IEEE 802.3, and IPX Encapsulation Formats 19-4

Figure 20-1 OSI Primitives 20-3

Figure 20-2 OSI Address Format 20-5

Figure 20-3 Principle OSI Upper-Layer Protocols 20-6

Figure 21-1 VINES Protocol Stack 21-1

Figure 21-2 VINES Address Format 21-2

Figure 21-3 VINES Address Selection Process 21-3

Figure 21-4 VINES Routing Algorithm 21-5

Figure 21-5 VIP Packet Format 21-5

Figure 22-1 XNS and the OSI Reference Model 22-2

Figure 22-2 IDP Packet Format 22-3

Figure 23-1 Typical RIP Routing Table 23-2

Figure 23-2 RIP Packet Format 23-3

Figure 23-3 Count-To-Infinity Problem 23-4

Figure 23-4 Split Horizons 23-5

Figure 24-1 IGRP Packet Format 24-2

Figure 24-2 Split Horizons 24-4

Figure 25-1 Hierarchical OSPF Internetwork 25-3

Figure 25-2 OSPF Header Format 25-5

Figure 26-1 EGP and the ARPANET 26-1

Figure 26-2 EGP Packet Format 26-2
Figure 26-3 EGP Message Types 26-2
Figure 27-1 BGP Packet Format 27-2
Figure 28-1 Hierarchies in OSI Internetworks 28-2
Figure 28-2 ESH and ISH Packet Formats 28-3
Figure 28-3 IS-IS Logical Packet Format 28-5
Figure 28-4 IS-IS Common Header Format 28-5
Figure 29-1 Transparent Bridging Table 29-1
Figure 29-2 Inaccurate Forwarding and Learning in Transparent Bridging Environments 29-2
Figure 29-3 TB Network Before Running STA 29-4
Figure 29-4 TB Network After Running STA 29-5
Figure 29-5 TB Configuration Message Format 29-6
Figure 30-1 Sample SRB Network 30-2
Figure 30-2 IEEE 802.5 RIF 30-3
Figure 31-1 Bridging Between TB and SRB Domains 31-1
Figure 32-1 SNMP Management Model 32-2
Figure 32-2 MIB Tree 32-4
Figure 32-3 SNMP Message Format 32-6
Figure 33-1 OSI and IBM Network Management Functions 33-2
Figure 33-2 ONA Framework 33-4

About This Manual

This section discusses the objectives, audience, organization, and conventions of the *Internetworking Technology Overview* publication. It also lists related Cisco publications.

Document Objectives

This publication provides technical information on Cisco-supported internetworking technologies. It is designed for use in conjunction with other Cisco manuals or as a stand-alone reference.

The *Internetworking Technology Overview* is not intended to provide all possible information on the included technologies. Because a primary goal of this publication is to help network administrators configure Cisco products, the publication emphasizes Cisco-supported technologies.

Audience

The *Internetworking Technology Overview* is written for anyone who wants to understand internetworking technologies. Cisco anticipates that many readers will use information in this publication to help configure Cisco products, but others also may find it useful.

Readers will better understand the material in this publication if they are familiar with networking terminology. Cisco's *Internetworking Terms and Acronyms* publication is a useful reference for those with minimal knowledge of networking terms.

Document Organization

This publication is divided into seven parts. Each part is concerned with introductory material or a major area of internetworking technology and comprises chapters describing related tasks or functions.

- Part 1, “Internetworking Basics,” presents concepts basic to the understanding of internetworking and network management.
- Part 2, “Media-Access Technologies,” describes standard protocols used for accessing network physical media.
- Part 3, “Packet-Switching Technologies,” describes standard protocols used to implement packet-switching.
- Part 4, “Routed Protocols,” describes several standard networking protocol stacks that can be routed through an internetwork.
- Part 5, “Routing Protocols,” describes protocols used to route information through an internetwork.
- Part 6, “Bridging Technologies,” describes protocols and technologies used to provide Layer-2 connectivity between subnetworks.
- Part 7, “Network Management,” describes network management protocols, architectures, and technologies.

Document Conventions

In this publication, the following conventions are used:

- Commands and keywords are in **boldface**.
- New, important terms are *italicized* when accompanied by a definition or discussion of the term.
- Protocol names are *italicized* at their first use in each chapter.

Note: Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Related Cisco Documentation

Following is a list of related Cisco publications:

- *Internetworking Terms and Acronyms*
- *Router Products Configuration and Reference*, Vols. 1-3

To order these publications or additional copies of the *Internetworking Technology Overview*, contact your sales representative. (The Customer Order Number for each manual is located at the bottom of the title page.) Customer Service can provide you with the name of your sales representative if necessary.

Phone: 1-800-553-NETS (6387) or (415) 326-1941

E-mail: customer-service@cisco.com

Service and Support

Cisco Systems provides a full range of support services to ensure that you get maximum network uptime with low life-cycle equipment cost. This section contains instructions for contacting Customer Service and for obtaining assistance through the Technical Assistance Center (TAC). It also contains warranty and service information.

Warranty Information

All Cisco Systems products are covered under a limited factory warranty. This warranty covers defects in the hardware, software, or firmware. Refer to the Cisco Systems *Customer Services Product Guide* for more information on Cisco's warranty policy, or contact Customer Service at 1-800-553-NETS or (415) 326-1941.

Note: Warranty and other service agreements may differ for international customers. Contact your closest Cisco regional representative for more information.

Maintenance Agreements

Cisco Systems offers a Comprehensive Hardware Maintenance Agreement throughout North America that includes on-site remedial services, software support, a 24-hour emergency hot line, overnight parts replacement, and an escalation procedure. Cisco also offers software, maintenance, and advanced replacement services under a SMARTnet agreement for customers who desire those services. Noncontract maintenance services are provided at current time-and-materials rates. For more information, contact Customer Service at 1-800-553-NETS or (415) 326-1941.

Customer Support

Cisco's maintenance strategy is based upon customer-initiated service requests to the Cisco Systems Technical Assistance Center (TAC). The TAC coordinates all customer services, including hardware and software telephone technical support, onsite service requirements, and module exchange and repair.

The TAC is available Monday through Friday from 6:00 a.m. to 6:00 p.m. Pacific Coast time (excluding company holidays) at the numbers that follow. If you must return your Cisco equipment for repair or replacement, contact the TAC or a Cisco regional representative for more information.

Hardware and software support specialists who help diagnose and solve customer problems will be able to isolate and solve your problem much faster if you are prepared with the information they need (see the TAC escalation procedures page shipped with this product). When you call the TAC, have the following information ready:

- Chassis serial number
- Maintenance contract number
- Software version and hardware configuration

You can display your software version level and your hardware configuration by using the **show version** command.

Technical Assistance (TAC):

1-800-553-2447 Fax: (415) 903-8787
(415) 688-8209 E-mail: tac@cisco.com

Sales, Orders, Questions, and Comments:

1-800-553-NETS (6387) Fax: (415) 903-8080
(415) 903-7208 E-mail: csrep@cisco.com

Obtaining Additional Information

This section describes how to obtain additional Cisco publications and includes tips for obtaining books, standards, and other information about networks and data communications that might be helpful while using Cisco products.

Ordering Additional Cisco Publications

To order these publications or additional copies of the *Internetworking Technology Overview* manual, contact your sales representative. (The Customer Order Number for each manual is located at the bottom of the title page.) Customer Service can provide you with the name of your sales representative if necessary.

1-800-553-NETS (6387)
(415) 326-1941
E-mail: customer-service@cisco.com

Obtaining Cisco Technical Information Electronically

Cisco provides a directory of documents that you can access electronically using File Transfer Protocol (FTP). The directory includes such publications as product release notes, descriptions of Management Information Bases (MIBs), commonly used Requests for Comments (RFCs), and technical notes. The directory does not include electronic versions of Cisco technical manuals.

To obtain these technical documents, proceed as follows:

Step 1: At your server prompt, use the **ftp** command to connect to address *ftp.cisco.com*.

```
% ftp ftp.cisco.com
```

When you connect to the directory, you are greeted with an informational banner:

```
Connected to dirt.cisco.com.  
220 dirt FTP server (Version 5.51.28 Mon Jan 13 17:51:58 PST 1992)  
ready.
```

This is followed by a login prompt.

Step 2: Enter the word **anonymous** as your login name:

```
Name (ftp.cisco.com:cindy): anonymous
```

The system responds with this message:

```
331 Guest login ok, send ident as password.  
Password:
```

Step 3: Enter your login name at the Password: prompt. The following message and ftp> prompt appear:

```
230 Guest login ok, access restrictions apply.  
ftp>
```

Step 4: To obtain a list of available files, enter **get README** at the ftp> prompt:

```
ftp> get README  
200 PORT command successful.  
150 Opening ASCII mode data connection for README (10093 bytes).  
226 Transfer complete.  
local: README remote: README  
10307 bytes received in 0.17 seconds (59 Kbytes/s)
```

Step 5: Enter the **get** command and the full file name for each file you require.

Step 6: To exit FTP, use the **quit** command.

```
ftp> quit  
221 Goodbye.
```

Note: In the FTP directory, the **ls** command does not accept wildcards; therefore, you cannot use this command to obtain a list of available files. To obtain a list of available files, you must use the README file.

Obtaining Information from Other Sources

This section describes how to obtain RFCs and technical standards.

For a list of relevant publications from other sources, see Appendix A, “References.”

Obtaining RFCs

Information about the Internet suite of protocols is contained in documents called *Requests for Comments*, or *RFCs*. These documents are maintained by Government Systems, Inc. (GSI). You can request copies by contacting GSI directly, or you can use the TCP/IP File Transfer Protocol (FTP) to obtain an electronic copy.

Contacting GSI

You can contact GSI through mail, by telephone, or through electronic mail.

Government Systems, Incorporated
Attn: Network Information Center
14200 Park Meadow Drive, Suite 200
Chantilly, Virginia 22021

1-800-365-3642
(703) 802-4535
(703) 802-8376 (FAX)

NIC@NIC.DDN.MIL
Network address: 192.112.36.5
Root domain server: 192.112.36.4

Obtaining an Electronic Copy

To obtain an electronic copy of an RFC via FTP, complete the following steps:

Step 1: At your server prompt, use the **ftp** command to connect to address *nic.ddn.mil*:

```
% ftp nic.ddn.mil
```

The following display appears, followed by a login prompt:

```
Connected to nic.ddn.mil.
220-*****Welcome to the Network Information Center*****
*****Login with username "anonymous" and password "guest"
*****You may change directories to the following:
    ddn-news          - DDN Management Bulletins
    domain            - Root Domain Zone Files
    ien               - Internet Engineering Notes
    iesg              - IETF Steering Group
    ietf              - Internet Engineering Task Force
    internet-drafts   - Internet Drafts
    netinfo           - NIC Information Files
    netprog           - Guest Software (ex. whois.c)
    protocols         - TCP-IP & OSI Documents
    rfc               - RFC Repository
    scc               - DDN Security Bulletins
220 And more.
```

Step 2: At the login prompt, enter the word **anonymous** as your login name:

```
Name (nic.ddn.mil:cindy): anonymous
```

The NIC responds with this message:

```
331 Guest login ok, send "guest" as password.
Password:
```

Step 3: Enter the word **guest** at the Password: prompt. The following message and ftp> prompt appear:

```
230 Guest login ok, access restrictions apply.
ftp>
```

Step 4: Use the **cd** command to change directories. The following example illustrates how to change the RFC directory and obtain RFC 1158:

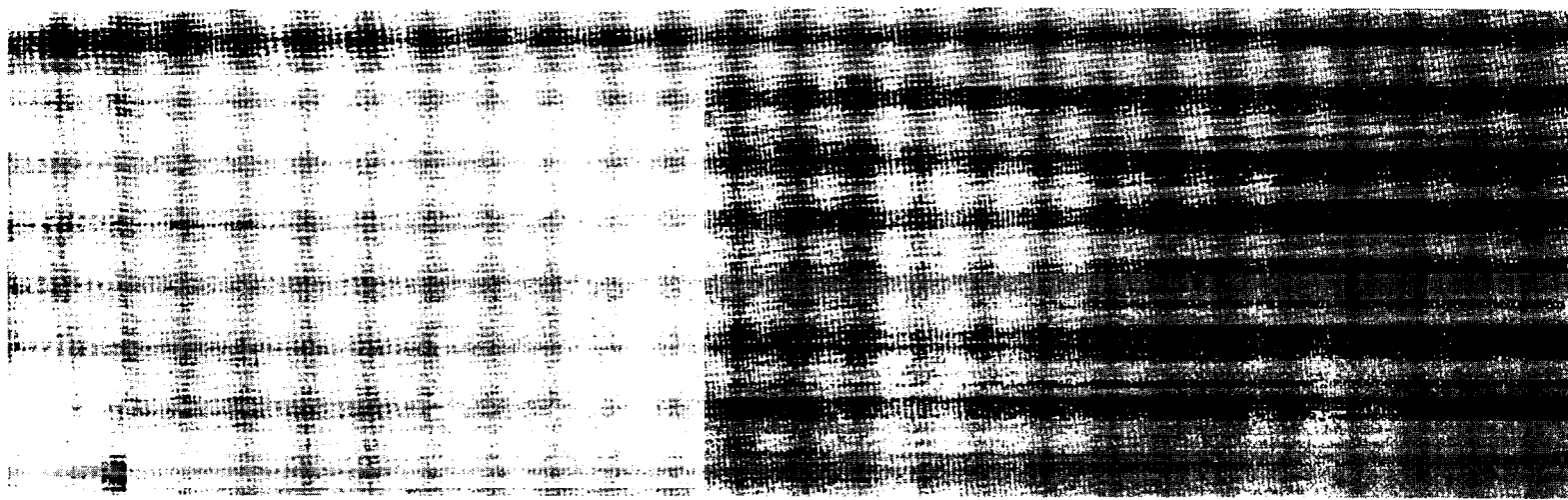
```
ftp> cd rfc  
250 CWD command successful.  
ftp> get rfc1158.txt
```

Step 5: To exit the FTP facility, enter the **quit** command at the ftp> prompt.

Obtaining Technical Standards

Following are additional sources for technical standards:

- Omnicom, 1-800-OMNICOM
- Global Engineering Documents, 2805 McGraw Ave., Irvine, CA 92714
1-800-854-7179
- American National Standards Institute, 1430 Broadway, New York, NY 10018
(212) 642-4932 or (212) 302-1286



Chapter 1

Introduction to Internetworking



Introduction

This chapter explains basic internetworking concepts. The foundational information presented here helps the reader comprehend the technical material that makes up the bulk of this publication. Sections on the OSI reference model, important terms and concepts, and key organizations are included.

OSI Reference Model

Moving information between computers of possibly diverse design is a formidable task. In the early 1980s, the International Organization for Standardization (ISO) recognized the need for a network model that would help vendors create interoperable network implementations. The *Open Systems Interconnection (OSI) reference model*, released in 1984, addresses this need.

The OSI reference model quickly became the primary architectural model for intercomputer communications. Although other architectural models (mostly proprietary) have been created, most network vendors relate their network products to the OSI reference model when they want to educate users about their products. As such, the model is the best tool available to those hoping to learn about network technology.

Hierarchical Communication

The OSI reference model divides the problem of moving information between computers over a network medium into seven smaller and more manageable problems. Each of the seven smaller problems was chosen because it was reasonably self-contained and therefore more easily solved without excessive reliance on external information.

Each of the seven problem areas is solved by a *layer* of the model. Most network devices implement all seven layers. To streamline operations, however, some network implementations skip one or more layers. The lower two OSI layers are implemented with hardware and software; the upper five layers are generally implemented in software.

The OSI reference model describes how information makes its way from application programs (such as spreadsheets) through a network medium (such as wires) to another application program in another computer. As the information to be sent descends through the layers of a given system, the information looks less and less like human language and more and more like the ones and zeros that a computer understands.

As an example of OSI-type communication, assume that System A in Figure 1-1 has information to send to System B. The application program in System A communicates with System A's Layer 7 (the top layer), which communicates with System A's Layer 6, which communicates with System A's Layer 5, and so on until System A's Layer 1 is reached. Layer 1 is concerned with putting information on (and taking information off) the physical network medium. After the information has traversed the physical network medium and been absorbed into System B, it ascends through System B's layers in reverse order (first Layer 1, then Layer 2, and so on) until it finally reaches System B's application program.

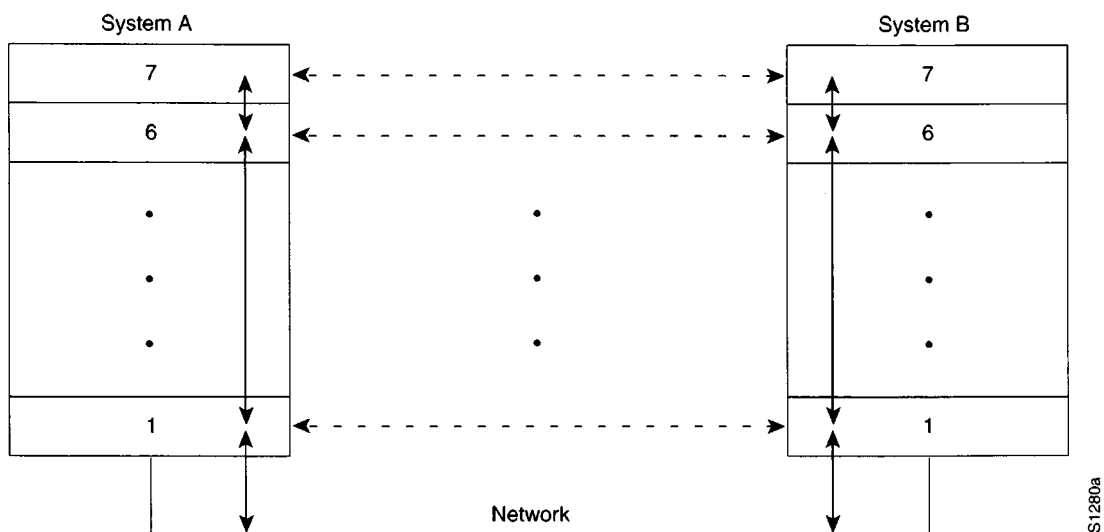


Figure 1-1 Communication Between Two Computer Systems

Although each of System A's layers communicates with adjacent System A layers, their primary objective is to communicate with their peer layers in System B. That is, the primary objective of Layer 1 in System A is to communicate with Layer 1 in System B; Layer 2 in System A communicates with Layer 2 in System B, and so on. This is necessary because each layer in a System has certain tasks it must perform. To perform these tasks, it must communicate with its peer layer in the other system.

The OSI model's layering precludes direct communication between peer layers in different systems. Each layer in System A must therefore rely on services provided by adjacent System A layers to help achieve communication with its System B peer. The relationship between adjacent layers in a single system is shown in Figure 1-2.

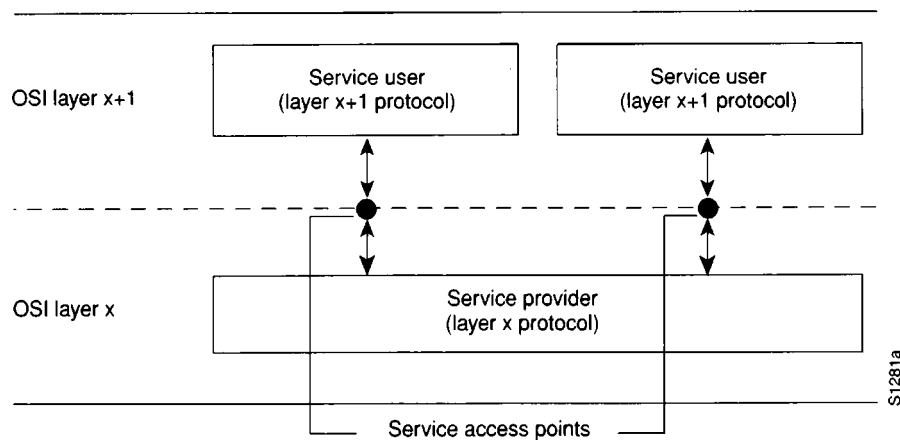


Figure 1-2 Relationship Between Adjacent Layers in a Single System

Assume Layer 4 in System A must communicate with Layer 4 in System B. To do this, Layer 4 in System A must use the services of Layer 3 in System A. Layer 4 is said to be the *service user*, while Layer 3 is the *service provider*. Layer 3 services are provided to Layer 4 at a *service access point (SAP)*, which is simply a location at which Layer 4 can request Layer 3 services. As the figure shows, Layer 3 can provide its services to multiple Layer 4 entities.

Information Formats

How does Layer 4 in System B know what Layer 4 in System A wants? Layer 4's specific requests are stored as *control information*, which is passed between peer layers in a block called a *header* that is prepended to the actual application information. For example, assume System A wishes to send the following text (called *data* or *information*) to System B:

The small grey cat ran up the wall to try to catch the red bird.

This text is passed from the application program in System A to System A's top layer. System A's application layer must communicate certain information to System B's application layer, so it prepends that control information (in the form of a coded header) to the actual text to be moved. This information unit is passed to System A's Layer 6, which may prepend its own control information. The message grows in size as it descends through the layers until it reaches the network, where the original text and all associated control information travels to System B, where it is absorbed by System B's Layer 1. System B's Layer 1 strips the Layer 1 header, reads it, and then knows how to process the information unit. The slightly smaller information unit is passed to Layer 2, which strips the Layer 2 header, analyzes the header for actions Layer 2 must take, and so forth. When the information unit finally reaches the application program in System B, it should simply contain the original text.

The concept of a header and data is relative, depending on the perspective of the layer currently analyzing the information unit. For example, to Layer 3, an information unit consists of a Layer 3 header and the data that follows. Layer 3's data, however, can potentially contain headers from Layers 4, 5, 6, and 7. Further, Layer 3's header is simply data to Layer 2.

This concept is illustrated in Figure 1-3. Finally, not all layers need to append headers. Some layers simply perform a transformation on the actual data they receive to make the data more or less readable to their adjacent layers.

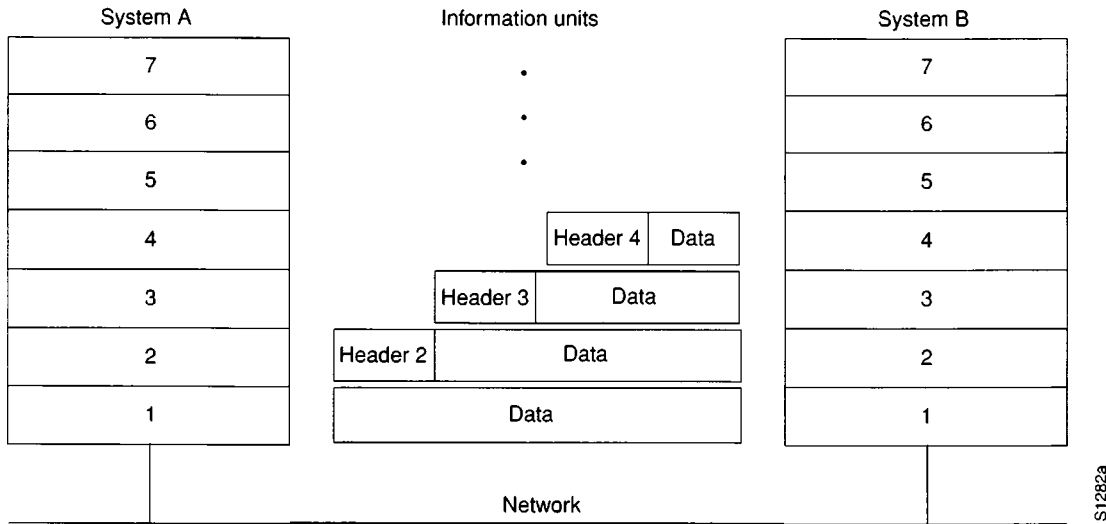


Figure 1-3 Headers and Data

Compatibility Issues

The OSI reference model is not a network implementation. Instead, it specifies the functions of each layer. In this way, it is like a blueprint for the building of a ship. After a ship blueprint is complete, the ship must still be built. Any number of shipbuilding companies can be contracted to do the actual work, just as any number of network vendors can build a protocol implementation from a protocol specification. And, unless the blueprint is extremely (impossibly) comprehensive, ships built by different shipbuilding companies using the same blueprint will differ from each other in at least minor ways. At the very least, for example, it is likely that the nails will be in different places.

What accounts for the differences between implementations of the same ship blueprint (or protocol specification)? In part, the differences are due to the inability of any specification to consider every possible implementation detail. Also, different implementors will no doubt interpret the blueprint in slightly different ways. And, finally, the inevitable implementation errors will cause different implementations to differ in execution. This explains why one company's implementation of protocol X does not always interoperate with another company's implementation of that protocol.

OSI Layers

Now that the basic features of the OSI layered approach are understood, each individual OSI layer and its functions can be discussed. Each layer has a predetermined set of functions it must perform for communication to occur.

Application Layer

The application layer is the OSI layer closest to the user. It differs from the other layers in that it does not provide services to any other OSI layer, but rather to application processes lying outside the scope of the OSI model. Examples of such application processes include spreadsheet programs, word processing programs, banking terminal programs, and so on.

The application layer identifies and establishes the availability of intended communication partners, synchronizes cooperating applications, and establishes agreement on procedures for error recovery and control of data integrity. Also, the application layer determines whether sufficient resources for the intended communication exist.

Presentation Layer

The presentation layer ensures that information sent by the application layer of one system will be readable by the application layer of another system. If necessary, the presentation layer translates between multiple data representation formats by using a common data representation format.

The presentation layer concerns itself not only with the format and representation of actual user data, but also with data structures used by programs. Therefore, in addition to actual data format transformation (if necessary), the presentation layer negotiates data transfer syntax for the application layer.

Session Layer

As its name implies, the session layer establishes, manages, and terminates sessions between applications. Sessions consist of dialogue between two or more presentation entities (recall that the session layer provides its services to the presentation layer). The session layer synchronizes dialogue between presentation layer entities and manages their data exchange. In addition to basic regulation of conversations (sessions), the session layer offers provisions for data expedition, class of service, and exception reporting of session-layer, presentation-layer, and application-layer problems.

Transport Layer

The boundary between the session layer and the transport layer can be thought of as the boundary between application-layer protocols and lower-layer protocols. Whereas the application, presentation, and session layers are concerned with application issues, the lower four layers are concerned with data transport issues.

The transport layer attempts to provide a data transport service that shields the upper layers from transport implementation details. Specifically, issues such as how reliable transport over an internetwork is accomplished are the concern of the transport layer. In providing reliable service, the transport layer provides mechanisms for the establishment, maintenance, and orderly termination of virtual circuits, transport fault detection and recovery, and information flow control (to prevent one system from overrunning another with data).

Network Layer

The network layer is a complex layer that provides connectivity and path selection between two end systems that may be located on geographically diverse *subnetworks*. A subnetwork, in this instance, is essentially a single network cable (sometimes called a *segment*).

Because a substantial geographic distance and many subnetworks can separate two end systems desiring communication, the network layer is the domain of routing. Routing protocols select optimal paths through the series of interconnected subnetworks. Traditional network-layer protocols then move information along these paths.

Link Layer

The link layer (formally referred to as the data-link layer) provides reliable transit of data across a physical link. In so doing, the link layer is concerned with *physical* (as opposed to *network*, or *logical*) addressing, network topology, line discipline (how end systems will use the network link), error notification, ordered delivery of frames, and flow control.

Physical Layer

The physical layer defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between end systems. Such characteristics as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, physical connectors, and other, similar, attributes are defined by physical layer specifications.

Important Terms and Concepts

Internetworking, like other sciences, has a terminology and knowledge base all its own. Unfortunately, because the science of internetworking is so young, universal agreement on the meaning of networking concepts and terms has not yet occurred. Definitions of internetworking terms will become more rigidly defined and employed as the internetworking industry matures.

Addressing

Locating computer systems on an internetwork is an essential component of any network system. There are various addressing schemes used for this purpose, depending on the protocol family being used. In other words, AppleTalk addressing is different from TCP/IP addressing, which in turn is different from OSI addressing, and so on.

Two important types of addresses are *link-layer* addresses and *network-layer* addresses. Link-layer addresses (also called *physical* or *hardware* addresses) are typically unique for each network connection. In fact, for most local area networks (LANs), link-layer addresses are resident in the interface circuitry and are assigned by the organization that defined the protocol standard represented by the interface. Because most computer systems have one

physical network connection, they have only a single link-layer address. Routers and other systems connected to multiple physical networks can have multiple link-layer addresses. As their name implies, link-layer addresses exist at Layer 2 of the OSI reference model.

Network-layer addresses (also called virtual or logical addresses) exist at Layer 3 of the OSI reference model. Unlike link-layer addresses, which usually exist within a flat address space, network-layer addresses are usually hierarchical. In other words, they are like mail addresses, which describe a person's location by providing a country, a state, a zip code, a city, a street, an address on the street, and finally, a name. One good example of a flat address space is the U.S. social security numbering system, wherein each person has a single, unique social security number.

Hierarchical addresses make address sorting and recall easier by eliminating large blocks of logically-similar addresses through a series of comparison operations. For example, one can eliminate all other countries if an address specifies the country *Ireland*. Easy sorting and recall is one reason that routers use network-layer addresses as the basis for routing.

Network-layer addresses differ depending on the protocol family being used, but they typically use similar logical divisions to find computer systems on an internetwork. Some of these logical divisions are based on physical network characteristics (such as the network segment a system is located on); others are based on groupings that have no physical basis (for example, the AppleTalk *zone*).

Frames, Packets, and Messages

Once addresses have located computer systems, information can be exchanged between two or more of these systems. Networking literature is inconsistent in naming the logically grouped units of information that move between computer systems. The terms *frame*, *packet*, *protocol data unit*, *PDU*, *segment*, *message*, and others have all been used, based on the whim of those who write protocol specifications.

In this publication, the term *frame* denotes an information unit whose source and destination is a link-layer entity. The term *packet* denotes an information unit whose source and destination is a network-layer entity. Finally, the term *message* denotes an information unit whose source and destination entity exists above the network layer. *Message* is also used to refer to particular lower-layer information units with a specific, well-defined purpose.

Key Organizations

Without the services of several key standards organizations, the world of networking would be substantially more chaotic than it is currently. Standards organizations provide forums for discussion, help turn discussion into formal specifications, and proliferate the specifications once they complete the standardization process.

Most standards organizations have specific processes for turning ideas into formal standards. Although these processes differ slightly between standards organizations, they are similar in that they all iterate through several rounds of organizing ideas, discussing the ideas, developing draft standards, voting on all or certain aspects of the standards, and finally formally releasing the completed standard to the public.

Some of the better-known standards organizations are:

- **International Organization for Standardization (ISO)**—An international standards organization responsible for a wide range of standards, including those relevant to networking. This organization is responsible for the OSI reference model and the OSI protocol suite.
- **American National Standards Institute (ANSI)**—The coordinating body for voluntary standards groups within the United States. ANSI is a member of ISO. ANSI's best-known communications standard is FDDI.
- **Electronic Industries Association (EIA)**—A group that specifies electrical transmission standards. EIA's best-known standard is RS-232.
- **Institute of Electrical and Electronic Engineers (IEEE)**—Professional organization that defines network standards. IEEE LAN standards (including IEEE 802.3 and IEEE 802.5) are the best-known IEEE communications standards and are the predominant LAN standards in the world today.
- **Consultative Committee for International Telegraph and Telephone (CCITT)**—An international organization that develops communication standards. CCITT's best-known standard is X.25.
- **Internet Activities Board (IAB)**—A group of internetwork researchers who meet regularly to discuss issues pertinent to the Internet. This board sets much of the policy for the Internet through decisions and assignment of task forces to various issues. Some *Request for Comments (RFC)* documents are designated by the IAB as Internet standards, including *Transmission Control Protocol/Internet Protocol (TCP/IP)* and the *Simple Network Management Protocol (SNMP)*.

Chapter 2

Routing Basics

2

Background

Routing, in the popular sense, is moving information across an internetwork from source to destination. Along the way, at least one intermediate node is typically encountered. Routing is often contrasted with bridging which, in the popular sense, accomplishes precisely the same function! The primary difference between the two is that bridging occurs at Layer 2 of the OSI reference model, while routing occurs at Layer 3. This distinction provides routing and bridging with different information to use in the process of moving information from source to destination. As a result, routing and bridging accomplish their tasks in different ways and, in fact, there are several different kinds of routing and bridging. For more information on bridging, see Chapter 3, “Bridging Basics.”

The topic of routing has been covered in computer science literature for over two decades, but routing only achieved commercial popularity in the mid-1980s. The primary reason for this time lag is the nature of networks in the 1970s. During this time, networks were fairly simple, homogeneous environments. Only recently has large-scale internetworking become popular.

Routing Components

Routing involves two basic activities: determination of optimal routing paths and the transport of information groups (typically called *packets*) through an internetwork. In this publication, the latter of these is referred to as *switching*. Switching is relatively straightforward. Path determination, on the other hand, can be very complex.

Path Determination

Path determination can be based on a variety of metrics (values resulting from algorithmic computations on a particular variable—for example, path length) or metric combinations. Software implementations of routing algorithms calculate route metrics to determine optimal routes to a destination.

To aid the process of path determination, routing algorithms initialize and maintain *routing tables*, which contain route information. Route information varies depending on the routing algorithm used.

Routing algorithms fill routing tables with a variety of information. Destination/next hop associations tell a router that a particular destination can be gained optimally by sending the packet to a particular router representing the “next hop” on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop. Figure 2-1 shows an example of a destination/next hop routing table.

To reach network:	Send to:
27	Node A
57	Node B
17	Node C
24	Node A
52	Node A
16	Node B
26	Node A
.	.
.	.
.	.

S1283a

Figure 2-1 Destination/Next Hop Routing Table

Routing tables can contain other information as well. *Metrics* provide information about the desirability of a link or path. Routers compare metrics to determine optimal routes. Metrics differ depending on the design of the routing algorithm being used. A variety of common metrics will be introduced and described later in this chapter.

Routers communicate with one another (and maintain their routing tables) through the transmission of a variety of messages. The *routing update* message is one such message. Routing updates generally consist of all or a portion of a routing table. By analyzing routing updates from all routers, a router can build a detailed picture of network topology. A *link-state advertisement* is another example of a message sent between routers. Link-state advertisements inform other routers of the state of the sender’s links. Link information can also be used to build a complete picture of network topology. Once the network topology is understood, routers can determine optimal routes to network destinations.

Switching

Switching algorithms are relatively simple and are basically the same for most routing protocols. In most cases, a host determines that it must send a packet to another host. Having acquired a router’s address by some means, the source host sends a packet addressed specifically to a router’s physical (MAC-layer) address, but with the protocol (network-layer) address of the destination host.

Upon examining the packet's destination protocol address, the router determines that it either knows or does not know how to forward the packet to the next hop. In the latter case (where the router does not know how to forward the packet), the packet is typically dropped. In the former case, the router sends the packet to the next hop by changing the destination physical address to that of the next hop and transmitting the packet.

The next hop may or may not be the ultimate destination host. If not, the next hop is usually another router, which executes the same switching decision process. As the packet moves through the internetwork, its physical address changes but its protocol address remains constant. This process is illustrated in Figure 2-2.

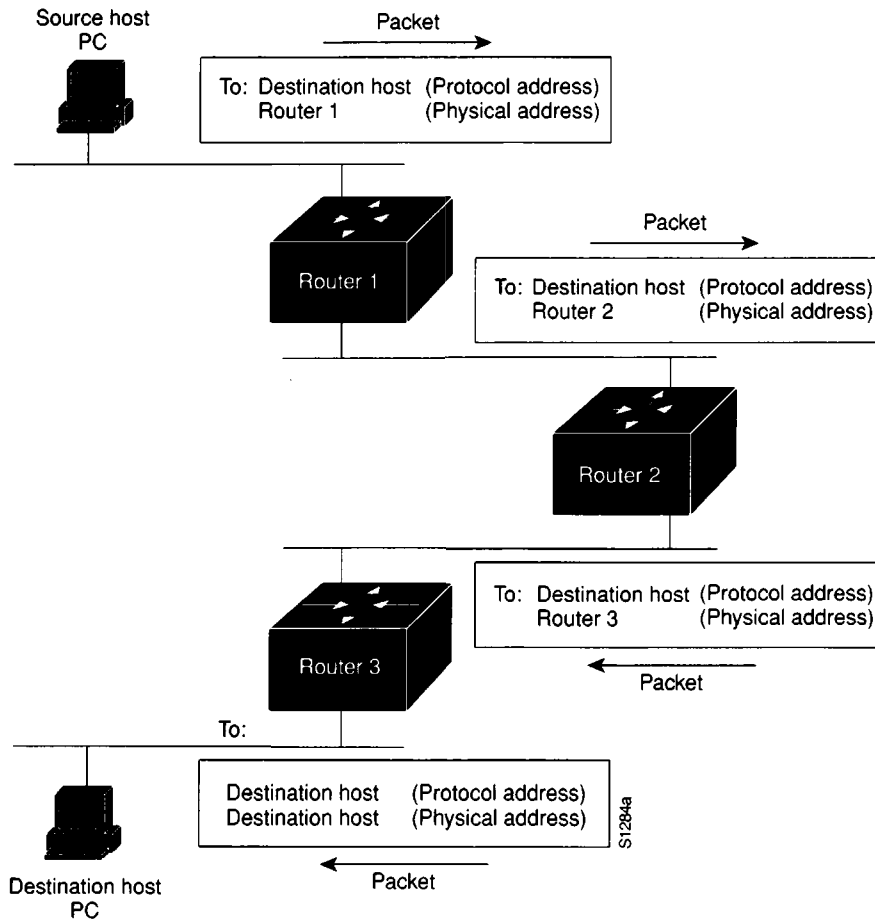


Figure 2-2 Switching Process

The preceding discussion describes switching between a source and a destination end system. The International Organization for Standardization (ISO) has developed a hierarchical terminology that is useful in describing this process. Using this terminology, network devices without the ability to forward packets between subnetworks are called *end systems* (ESs), while network devices with these capabilities are referred to as *intermediate systems* (ISs). ISs are further divided into those that can communicate within routing domains (*intradomain ISs*) and those that communicate both within and between routing domains (*interdomain ISs*). A *routing domain* is generally considered to be a portion of an internetwork under common administrative authority, regulated by a particular set of administrative guidelines. Routing

domains are also called *autonomous systems* (ASs). With certain protocols, routing domains may also be divided into *routing areas*, but intradomain routing protocols are still used for switching both within and between areas.

Routing Algorithms

Routing algorithms can be differentiated based on several key characteristics. First, the particular goals of the algorithm designer affect the operation of the resulting routing protocol. Second, there are various types of routing algorithms. Each algorithm has a different impact on network and router resources. Finally, routing algorithms use a variety of metrics that affect calculation of optimal routes. The following sections analyze these routing algorithm attributes.

Design Goals

Routing algorithms often have one or more of the following design goals:

- Optimality
- Simplicity/Low Overhead
- Robustness/Stability
- Rapid Convergence
- Flexibility

Optimality

Optimality is perhaps the most general design goal. It refers to the ability of the routing algorithm to select the “best” route. The best route depends upon the metrics and metric weightings used to make the calculation. For example, one routing algorithm might use number of hops and delay, but might weight delay more heavily in the calculation. Naturally, routing protocols must strictly define their metric calculation algorithms.

Simplicity

Routing algorithms are also designed to be as simple as possible. In other words, the routing algorithm must offer its functionality efficiently, with a minimum of software and utilization overhead. Efficiency is particularly important when the software implementing the routing algorithm must run on a computer with limited physical resources.

Robustness

Routing algorithms must be robust. In other words, they should perform correctly in the face of unusual or unforeseen circumstances such as hardware failures, high load conditions, and incorrect implementations. Because routers are located at network junction points, they can cause considerable problems when they fail. The best routing algorithms are often those that have withstood the test of time and proven stable under a variety of network conditions.

Rapid Convergence

Routing algorithms must converge rapidly. Convergence is the process of agreement, by all routers, on optimal routes. When a network event causes routes to either go down or become available, routers distribute routing update messages. Routing update messages permeate networks, stimulating recalculation of optimal routes and eventually causing all routers to agree on these routes. Routing algorithms that converge slowly can cause routing loops or network outages.

Figure 2-3 depicts a routing loop. In this case, a packet arrives at router 1 at time t_1 . Router 1 has already been updated and so knows that the optimal route to the destination calls for router 2 to be the next stop. Router 1 therefore forwards the packet to router 2. Router 2 has not yet been updated and so believes that the optimal next hop is router 1. Router 2 therefore forwards the packet back to router 1. The packet will continue to bounce back and forth between the two routers until router 2 receives its routing update or until the packet has been switched more than the maximum number of times allowed.

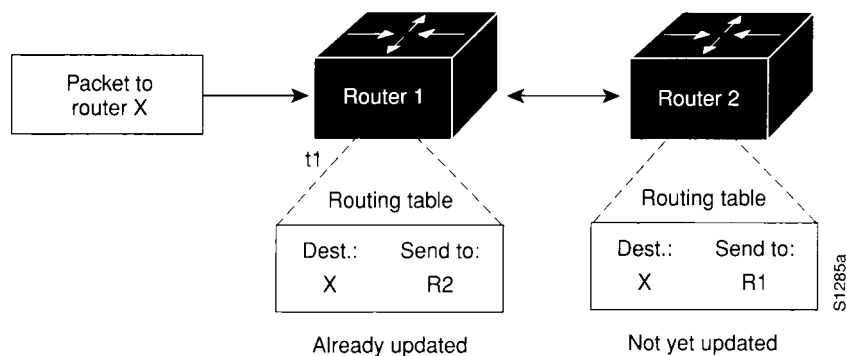


Figure 2-3 Slow Convergence and Routing Loops

Flexibility

Routing algorithms should also be flexible. In other words, routing algorithms should quickly and accurately adapt to a variety of network circumstances. For example, assume that a network segment has gone down. Many routing algorithms, upon becoming aware of this problem, will quickly select the next-best path for all routes normally using that segment. Routing algorithms can be programmed to adapt to changes in network bandwidth, router queue size, network delay, and other variables.

Types

Routing algorithms can be classified by type. For example, algorithms can be:

- Static or dynamic
- Single-path or multipath
- Flat or hierarchical
- Host-intelligent or router-intelligent
- Intradomain or interdomain
- Link state or distance vector

Static or Dynamic

Static routing algorithms are hardly algorithms at all. Static routing table mappings are established by the network administrator prior to the beginning of routing. They do not change unless the network administrator changes them. Algorithms that use static routes are simple to design and work well in environments where network traffic is relatively predictable and network design is relatively simple.

Because static routing systems cannot react to network changes, they are generally considered unsuitable for today's large, constantly changing networks. Most of the dominant routing algorithms in the 1990s are dynamic.

Dynamic routing algorithms adjust, in real time, to changing network circumstances. They do this by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, stimulating routers to rerun their algorithms and change their routing tables accordingly.

Dynamic routing algorithms may be supplemented with static routes where appropriate. For example, a *router of last resort* (a router to which all unroutable packets are sent) may be designated. This router acts as a repository for all unroutable packets, ensuring that all messages are at least handled in some way.

Single-Path or Multipath

Some sophisticated routing protocols support multiple paths to the same destination. These multipath algorithms permit traffic multiplexing over multiple lines; single-path algorithms do not. The advantages of multipath algorithms are obvious; they can provide substantially better throughput and reliability.

Flat or Hierarchical

Some routing algorithms operate in a flat space, while others use routing hierarchies. In a flat routing system, all routers are peers of all others. In a hierarchical routing system, some routers form what amounts to a routing backbone. Packets from nonbackbone routers travel

to the backbone routers, where they are sent through the backbone until they reach the general area of the destination. At this point, they travel from the last backbone router through one or more nonbackbone routers to the final destination.

Routing systems often designate logical groups of nodes called domains, ASs, or areas. In hierarchical systems, some routers in a domain can communicate with routers in other domains, while others can only communicate with routers within their domain. In very large networks, additional hierarchical levels may exist. Routers at the highest hierarchical level form the routing backbone.

The primary advantage of hierarchical routing is that it mimics the organization of most companies and therefore supports their traffic patterns very well. Most network communication occurs within small company groups (domains). Intradomain routers only need to know about other routers within their domain, so their routing algorithms can be simplified. Depending on the routing algorithm being used, routing update traffic can be reduced accordingly.

Host-Intelligent or Router-Intelligent

Some routing algorithms assume that the source end-node will determine the entire route. This is usually referred to as source routing. In source-routing systems, routers merely act as store-and-forward devices, mindlessly sending the packet to the next stop.

Other algorithms assume that hosts know nothing about routes. In these algorithms, routers determine the path through the internetwork based on their own calculations. In the first system, the hosts have the routing intelligence. In the latter system, routers have the routing intelligence.

The trade-off between host-intelligent and router-intelligent routing is one of path optimality versus traffic overhead. Host-intelligent systems choose the better routes more often, because they typically discover all possible routes to the destination before the packet is actually sent. They then choose the best path based on that particular system's definition of optimal. The act of determining all routes, however, often requires substantial discovery traffic and a significant amount of time.

Intradomain or Interdomain

Some routing algorithms work only within domains; others work within and between domains. The nature of these two algorithm types is different. It stands to reason, therefore, that an optimal intradomain routing algorithm would not necessarily be an optimal interdomain routing algorithm.

Link-State or Distance Vector

Link-state algorithms (also known as *shortest path first* algorithms) flood routing information to all nodes in the internetwork. However, each router sends only that portion of the routing table that describes the state of its own links. Distance-vector algorithms (also known as

Bellman-Ford algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link-state algorithms send small updates everywhere, while distance-vector algorithms send larger updates only to neighboring routers.

Because they converge more quickly, link-state algorithms are somewhat less prone to routing loops than are distance-vector algorithms. On the other hand, link-state algorithms are computationally difficult compared to distance-vector algorithms, requiring more CPU power and memory than distance-vector algorithms. Link-state algorithms can therefore be more expensive to implement and support. Despite their differences, both algorithm types perform well in most circumstances.

Metrics

Routing tables contain information used by switching software to select the best route. But how, specifically, are routing tables built? What is the specific nature of the information they contain? This section on algorithm metrics attempts to answer the question “how do routing algorithms determine that one route is preferable to others?”

Many different metrics have been used in routing algorithms. Sophisticated routing algorithms can base route selection on multiple metrics, combining them in a manner resulting in a single (hybrid) metric. All of the following metrics have been used:

- Path length
- Reliability
- Delay
- Bandwidth
- Load
- Communication cost

Path Length

Path length is the most common routing metric. Some routing protocols allow network administrators to assign arbitrary costs to each network link. In this case, path length is the sum of the costs associated with each link traversed. Other routing protocols define *hop count*, a metric that specifies the number of passes through internetworking products (such as routers) that a packet must take en route from a source to a destination.

Reliability

Reliability, in the context of routing algorithms, refers to the reliability (usually described in terms of the bit-error rate) of each network link. Some network links may go down more often than others. Once down, some network links may be repaired more easily or more quickly than other links. Any reliability factors can be taken into account in the assignment of reliability ratings. Reliability ratings are usually assigned to network links by network administrators. They are typically arbitrary numeric values.

Delay

Routing delay refers to the length of time required to move a packet from source to destination through the internetwork. Delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, network congestion on all intermediate network links, and the physical distance to be travelled. Because it is a conglomeration of several important variables, delay is a common and useful metric.

Bandwidth

Bandwidth refers to the available traffic capacity of a link. All other things being equal, a 10 Mbps Ethernet link would be preferable to a 64 Kbps leased line. Although bandwidth is a rating of the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. If, for example, a faster link is much busier, the actual time required to send a packet to the destination may be greater through the fast link.

Load

Load refers to the degree to which a network resource (such as a router) is busy. Load may be calculated in a variety of ways, including CPU utilization and packets processed per second. Monitoring these parameters on a continual basis can itself be resource intensive.

Communications Cost

Communication cost is another important metric. Some companies may not care about performance as much as they care about operating expenditures. Even though line delay might be longer, they will send packets over their own lines rather than through public lines that will cost money for usage time.

Routed vs. Routing Protocols

Confusion about the terms *routed* protocol and *routing* protocol is common. Routed protocols are protocols that are routed over an internetwork. Examples of such protocols are the *Internet Protocol (IP)*, *DECnet*, and *AppleTalk*. Routing protocols are protocols that implement routing algorithms. Put simply, they route routed protocols through an internetwork. Examples of these protocols include *Interior Gateway Routing Protocol (IGRP)*, *Open Shortest Path First (OSPF)*, *Intermediate System to Intermediate System (IS-IS)*, and *Routing Information Protocol (RIP)*.

Chapter 3

Bridging Basics

3

Background

Bridges became commercially available in the early 1980s. At the time of their introduction, bridges connected and enabled packet forwarding between homogeneous networks. More recently, bridging between different networks has also been defined and standardized.

Several kinds of bridging have emerged as important. *Transparent bridging* is found primarily in Ethernet environments. *Source-route bridging* is found primarily in Token Ring environments. *Translational bridging* provides translation between the formats and transit principles of different media types (usually Ethernet and Token Ring). *Source-route transparent bridging* combines the algorithms of transparent bridging and source-route bridging to allow communication in mixed Ethernet/Token Ring environments.

The diminishing price and the recent inclusion of bridging capability in many routers has taken substantial market share away from pure bridges. Those bridges that have survived include features such as sophisticated filtering, pseudo-intelligent path selection, and high throughput rates. Whereas an intense debate about the benefits of bridging versus routing raged in the late 1980s, most now agree that each has its place and that both are often necessary in any comprehensive internetworking scheme.

Internetworking Device Comparison

Internetworking devices offer communication between local area network (LAN) segments. There are four primary types of internetworking devices: *repeaters*, *bridges*, *routers*, and *gateways*. These devices can be differentiated very generally by the *Open System Interconnection (OSI)* layer at which they establish the LAN-to-LAN connection. Repeaters connect LANs at OSI Layer 1; bridges connect LANs at Layer 2; routers connect LANs at Layer 3; and gateways connect LANs at Layers 4 through 7. Each device offers the functionality found at its layers of connection and uses the functionality of all lower layers. This idea is portrayed graphically in Figure 3-1.

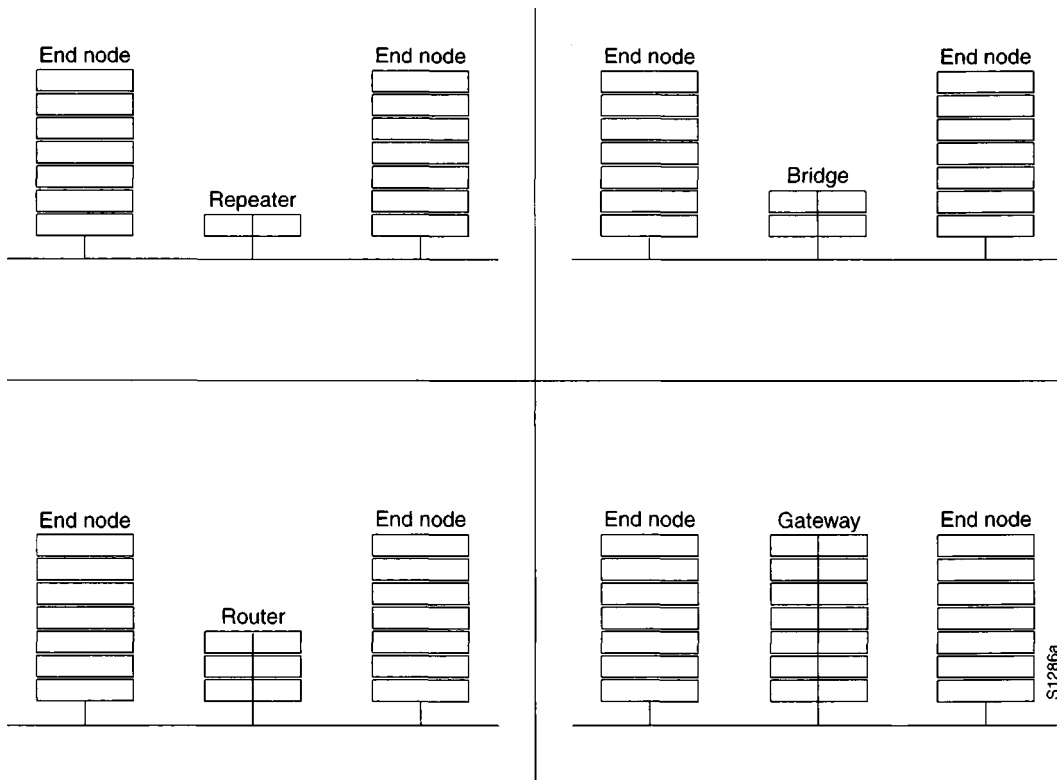


Figure 3-1 Internetworking Product Functionality

Technology Basics

The layer at which bridging occurs (also called the *data-link layer*, or simply the *link layer*) controls data flow, handles transmission errors, provides physical (as opposed to logical) addressing, and manages access to the physical medium. Bridges provide these functions by supporting various link-layer protocols that dictate specific flow control, error handling, addressing, and media-access algorithms. Examples of popular data-link layer protocols include Ethernet, Token Ring, and FDDI.

Bridges are not complicated devices. They analyze incoming frames, make forwarding decisions based on information contained in the frames, and forward the frames toward the destination. In some cases (for example, *source-route bridging*), the entire path to the destination is contained in each frame. In other cases (for example, *transparent bridging*), frames are forwarded one hop at a time toward the destination. See Chapter 30, “Source-Route Bridging,” and Chapter 29, “Transparent Bridging,” respectively, for more information on source-route bridging and transparent bridging.

Upper-layer protocol transparency is a primary advantage of bridging. Because bridges operate at the data-link layer, they are not required to examine upper-layer information. This means that they can rapidly forward traffic representing any network-layer protocol. It is not uncommon for a bridge to move AppleTalk, DECnet, TCP/IP, XNS, and other traffic between two or more networks.

Bridges are capable of filtering frames based on any Layer 2 fields. For example, a bridge can be programmed to reject (not forward) all frames sourced from a particular network. Since data-link layer information often includes a reference to an upper-layer protocol, bridges can usually filter on this parameter. Further, filters can be helpful in dealing with unnecessary broadcast and multicast packets.

By dividing large networks into self-contained units, bridges provide several advantages. First, because only some percentage of traffic is forwarded, the bridge diminishes the traffic experienced by devices on all connected segments. Second, the bridge acts as a firewall for some potentially damaging network errors. Third, bridges allow for communication between a larger number of devices than would be supported on any single LAN connected to the bridge. Fourth, bridges extend the effective length of a LAN, permitting attachment of distant stations not previously connected.

Types of Bridges

Bridges can be grouped into categories based on various product characteristics. Using one popular classification scheme, bridges are either *local* or *remote*. Local bridges provide a direct connection between multiple LAN segments in the same area. Remote bridges connect multiple LAN segments in different areas, usually over telecommunications lines. These two configurations are shown in Figure 3-2.

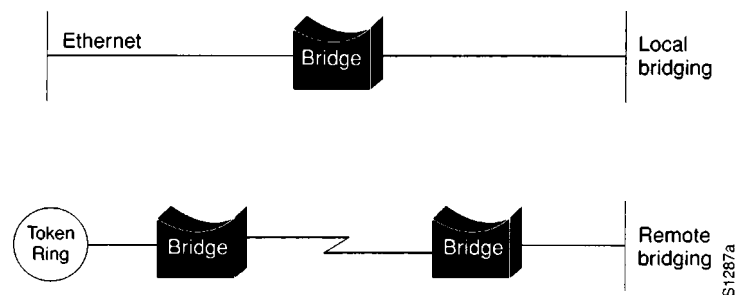


Figure 3-2 Local and Remote Bridging

Remote bridging presents several unique internetworking challenges. One of these is the difference between LAN and wide area network (WAN) speeds. Although several fast WAN technologies are now establishing a presence in geographically dispersed internetworks, LAN speeds are often an order of magnitude faster than WAN speeds. Vastly different LAN and WAN speeds sometimes prevent users from running delay-sensitive LAN applications over the WAN.

Remote bridges cannot improve WAN speeds, but can compensate for speed discrepancies through sufficient buffering capability. If a LAN device capable of a 3-Mbps transmission rate wishes to communicate with a device on a remote LAN, the local bridge must regulate the 3-Mbps data stream so that it does not overwhelm the 64-Kbps serial link. This is done by storing the incoming data in on-board buffers and sending it over the serial link at a rate the serial link can accommodate. This can be achieved only for short bursts of data that do not overwhelm the bridge's buffering capability.

The Institute of Electrical and Electronic Engineers (IEEE) organization has divided the OSI data-link layer into two separate sublayers: the *media access control (MAC)* sublayer and the *logical link control (LLC)* sublayer. The MAC sublayer permits and orchestrates media access (For example, contention, token passing, or others), while the LLC sublayer is concerned with framing, flow control, error control, and MAC-sublayer addressing.

Some bridges are *MAC-layer bridges*. These devices bridge between homogeneous networks (for example, IEEE 802.3 and IEEE 802.3). Other bridges can translate between different link-layer protocols (for example, IEEE 802.3 and IEEE 802.5). The basic mechanics of such a translation are shown in Figure 3-3.

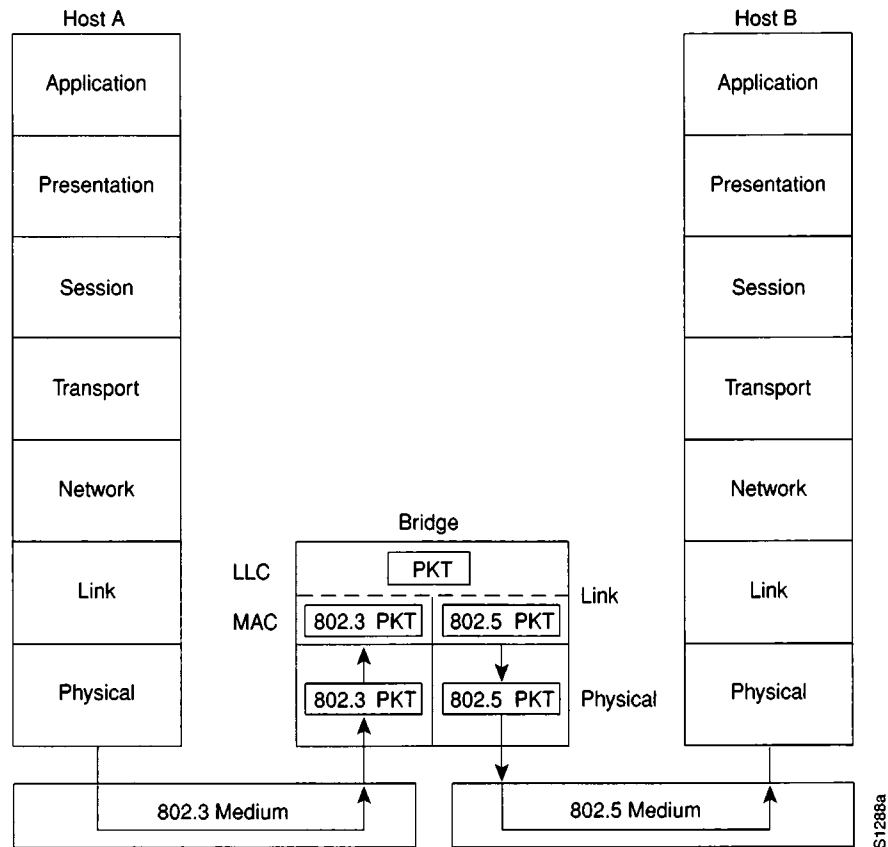


Figure 3-3 IEEE 802.3/IEEE 802.5 Bridging

In the figure, the IEEE 802.3 host (Host A) formulates a packet containing application information and encapsulates the packet in an IEEE 802.3-compatible frame for transit over the IEEE 802.3 medium to the bridge. At the bridge, the frame is stripped of its IEEE 802.3 header at the MAC sublayer of the link layer and is subsequently passed up to the LLC sublayer for further processing. After this processing, the packet is passed back down to an IEEE 802.5 implementation, which encapsulates the packet in an IEEE 802.5 header for transmission on the IEEE 802.5 network to the IEEE 802.5 host (Host B).

A bridge's translation between networks of different types is never perfect because it is likely that one network will support certain frame fields and protocol functions not supported by the other network. This is roughly analogous to the problem experienced by an Eskimo who tries to translate several of her fifty words for "snow" into English. Many bridging translation issues are discussed in more detail in Chapter 31, "Mixed-Media Bridging."

Chapter 4

Network Management Basics

4

Background

The early 1980s saw tremendous expansion in the area of network deployment. As companies realized the cost benefits and productivity gains created by network technology, they began adding networks and expanding existing networks almost as rapidly as new network technologies and products were introduced. By the mid-1980s, growing pains from this expansion were being felt, especially by those companies that had deployed many different (and incompatible) network technologies.

The primary problems associated with network expansion are day-to-day network operation management and strategic network growth planning. Specifically, each new network technology requires its own set of experts to operate and maintain. In the early 1980s, strategic planning for the growth of these networks became a nightmare. The staffing requirements alone for managing large, heterogeneous networks created a crisis for many organizations. Automated network management (including what is typically called *network capacity planning*), integrated across diverse environments, became an urgent need.

This chapter describes technical features common to most network management architectures and protocols. It also presents the five functional areas of management as defined by the International Organization for Standardization (ISO).

Network Management Architecture

Most network management architectures use the same basic structure and set of relationships. End stations (*managed devices*) such as computer systems and other network devices run software allowing them to send alerts when they recognize problems. Problems are recognized, when one or more user-determined thresholds are exceeded. Upon receiving these alerts, *management entities* are programmed to react by executing one, several, or all of a group of actions, including:

- Operator notification
- Event logging
- System shutdown
- Automatic attempts at system repair

Management entities can also poll end stations to check the values of certain variables. Polling can be automatic or user initiated. *Agents* in the managed devices respond to these polls. Agents are software modules that compile information about the managed devices in which they reside, store this information in a *management database*, and provide it (proactively or reactively) to management entities within *network management systems (NMSs)* via a *network management protocol*. Well-known network management protocols include the Simple Network Management Protocol (SNMP) and Common Management Information Protocol (CMIP). *Management proxies* are entities that provide management information on behalf of other entities. A typical network management architecture is shown in Figure 4-1.

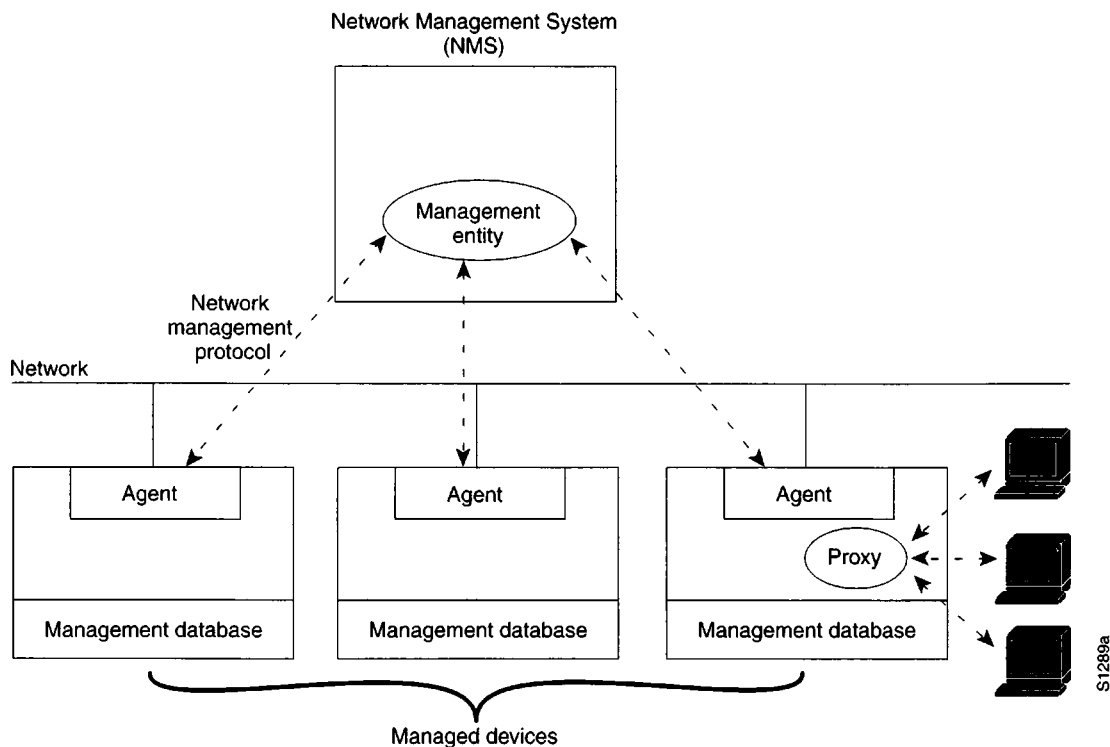


Figure 4-1 Typical Network Management Architecture

The ISO Network Management Model

ISO has contributed a great deal to network standardization. Their network management model is the primary means for understanding the major functions of network management systems. This model consists of five conceptual areas:

- Performance management
- Configuration management
- Accounting management
- Fault management
- Security management

Performance Management

The goal of performance management is to measure and make available various aspects of network performance so that internetwork performance can be maintained at an acceptable level. Examples of performance variables that might be provided include network throughput, user response times, and line utilization.

Performance management involves several steps:

1. Gather performance data on those variables of interest to network administrators.
2. Analyze the data to determine normal (baseline) levels.
3. Determine appropriate performance thresholds for each important variable such that exceeding of these thresholds indicates a network problem worthy of attention.

Managed entities continually monitor performance variables. When a performance threshold is exceeded, an alert is generated and sent to the NMS.

Each of the steps just described is part of the process to set up a reactive system. When performance becomes unacceptable by virtue of an exceeded user-defined threshold, the system reacts by sending a message. Performance management also permits proactive methods. For example, network simulation can be used to project how network growth will affect performance metrics. Such simulation can effectively alert administrators to impending problems, so that counteractive measures can be taken.

Configuration Management

The goal of configuration management is to monitor network and system configuration information so that the affects on network operation of various versions of hardware and software elements can be tracked and managed. Because all hardware and software elements have operational quirks, flaws, or both that might affect network operation, such information is important to maintaining a smooth-running network.

Each network device has a variety of version information associated with it. For example, an engineering workstation might be configured as follows:

- Operating system, Version 3.2
- Ethernet interface, Version 5.4
- TCP/IP software, Version 2.0
- NetWare software, Version 4.1
- NFS software, Version 5.1
- Serial communications controller, Version 1.1
- X.25 software, Version 1.0
- SNMP software, Version 3.1

Configuration management subsystems store this information in a database for easy access. When a problem occurs, this database can be searched for clues that might help solve the problem.

Accounting Management

The goal of accounting management is to measure network utilization parameters so that individual or group uses of the network can be regulated appropriately. Such regulation minimizes network problems (because network resources can be apportioned out based on resource capacities) and maximizes the fairness of network access across all users.

As with performance management, the first step toward appropriate accounting management is to measure utilization of all important network resources. Analysis of the results provides insight into current usage patterns. Usage quotas can be set at this point. Some correction will be required to reach optimal access practices. From that point on, ongoing measurement of resource use can yield billing information as well as information used to assess continued fair and optimal resource utilization.

Fault Management

The goal of fault management is to detect, log, notify users of, and (to the extent possible) automatically fix network problems so as to keep the network running effectively. Because faults can cause downtime or unacceptable network degradation, fault management is perhaps the most widely implemented of the ISO network management elements.

Fault management involves several steps:

1. Determine problem symptoms.
2. Isolate the problem.
3. Fix the problem.
4. Test the fix on all important subsystems.
5. Record the problem's detection and resolution.

Security Management

The goal of security management is to control access to network resources according to local guidelines so that the network cannot be sabotaged (intentionally or unintentionally) and sensitive information cannot be accessed by those without appropriate authorization. For example, a security management subsystem can monitor users logging onto a network resource, refusing access to those who enter inappropriate access codes.

Security management subsystems work by partitioning network resources into authorized and unauthorized areas. For some users, access to any network resources is inappropriate. Such users are usually company outsiders. For other (internal) network users, access to information originating from a particular department is inappropriate. For example, access to human resource files is inappropriate for any users outside the human resource department (excepting, potentially, executive staff).

Security management subsystems perform several functions:

- Identify sensitive network resources (including systems, files, and other entities).
- Determine mappings between sensitive network resources and user sets.
- Monitor access points to sensitive network resources.
- Log inappropriate access to sensitive network resources.

Part 2

Media-Access
Technologies

Chapter 5

Ethernet/IEEE 802.3

5

Background

Ethernet was developed by Xerox Corporation's *Palo Alto Research Center (PARC)* in the early- to mid-1970s. Ethernet was the technological basis for the IEEE 802.3 specification, which was initially released in 1980. Shortly thereafter, Digital Equipment Corporation, Intel Corporation, and Xerox Corporation jointly developed and released an Ethernet specification (Version 2.0) that is substantially compatible with IEEE 802.3. Together, Ethernet and IEEE 802.3 currently maintain the greatest market share of any local area network (LAN) protocol. Today, the term *Ethernet* is often used to refer to all *carrier sense multiple access/collision detection (CSMA/CD)* LANs that generally conform to Ethernet specifications, including IEEE 802.3.

At the time of its creation, Ethernet was designed to fill the middle ground between long-distance, low-speed networks and specialized, computer-room networks carrying data at high speeds for very limited distances. Ethernet is well-suited to applications where a local communication medium must carry sporadic, occasionally heavy traffic at high peak data rates.

Ethernet/IEEE 802.3 Comparison

Ethernet and IEEE 802.3 specify similar technologies. Both are CSMA/CD LANs. Stations on a CSMA/CD LAN can access the network at any time. Before sending data, CSMA/CD stations "listen" to the network to see if it is already in use. If so, the station wishing to transmit waits. If the network is not in use, the station transmits. A collision occurs when two stations listen for network traffic, "hear" none, and transmit simultaneously. In this case, both transmissions are damaged and the stations must retransmit at some later time. *Backoff* algorithms determine when the colliding stations retransmit. CSMA/CD stations can detect collisions so they know when they must retransmit.

Both Ethernet and IEEE 802.3 LANs are broadcast networks. In other words, all stations see all frames, regardless of whether they represent an intended destination. Each station must examine received frames to determine if the station is a destination. If so, the frame is passed to a higher protocol layer for appropriate processing.

Differences between Ethernet and IEEE 802.3 LANs are subtle. Ethernet provides services corresponding to Layers 1 and 2 of the OSI reference model, while IEEE 802.3 specifies the physical layer (Layer 1) and the channel-access portion of the link layer (Layer 2), but does not define a logical link control protocol. Both Ethernet and IEEE 802.3 are implemented in hardware. Typically, the physical manifestation of these protocols is either an interface card that inserts inside a host computer or circuitry on a primary circuit board within a host computer.

Physical Connections

IEEE 802.3 specifies several different physical layers, whereas Ethernet defines only one. Each IEEE 802.3 physical layer protocol has a name that summarizes its characteristics. The coded components of an IEEE 802.3 physical-layer name are shown in Figure 5-1.

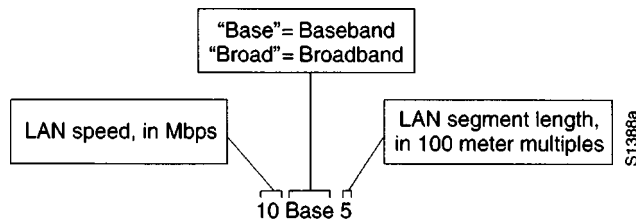


Figure 5-1 IEEE 802.3 Physical-Layer Name Components

A summary of Ethernet Version 2 and IEEE 802.3 characteristics appears in Figure 5-2.

	Ethernet	IEEE 802.3				
		10Base5	10Base2	1Base5	10BaseT	10Broad36
Data rate (Mbps)	10	10	10	1	10	10
Signaling method	Baseband	Baseband	Baseband	Baseband	Baseband	Broadband
Max. segment length (m)	500	500	185	250	100(UTP)	1800
Media	50-ohm coax (thick)	50-ohm coax (thick)	50-ohm coax (thin)	Unshielded twisted pair (UTP)	Unshielded twisted pair (UTP)	75-ohm coax
Topology	Bus	Bus	Bus	Star	Star	Bus

Figure 5-2 Ethernet V2.0 and IEEE 802.3 Physical Characteristics

Ethernet is most similar to IEEE 802.3 10Base5. Both of these protocols specify a bus topology network with a connecting cable between the end stations and the actual network medium. In the case of Ethernet, that cable is called a *transceiver cable*. The transceiver cable connects to a transceiver device attached to the physical network medium. The IEEE 802.3 configuration is much the same, except that the connecting cable is referred to as an *attachment unit interface (AUI)*, and the transceiver is called a *medium attachment unit (MAU)*. In both cases, the connecting cable attaches to an interface board (or interface circuitry) within the end station.

Frame Formats

Ethernet and IEEE 802.3 frame formats are shown in Figure 5-3.

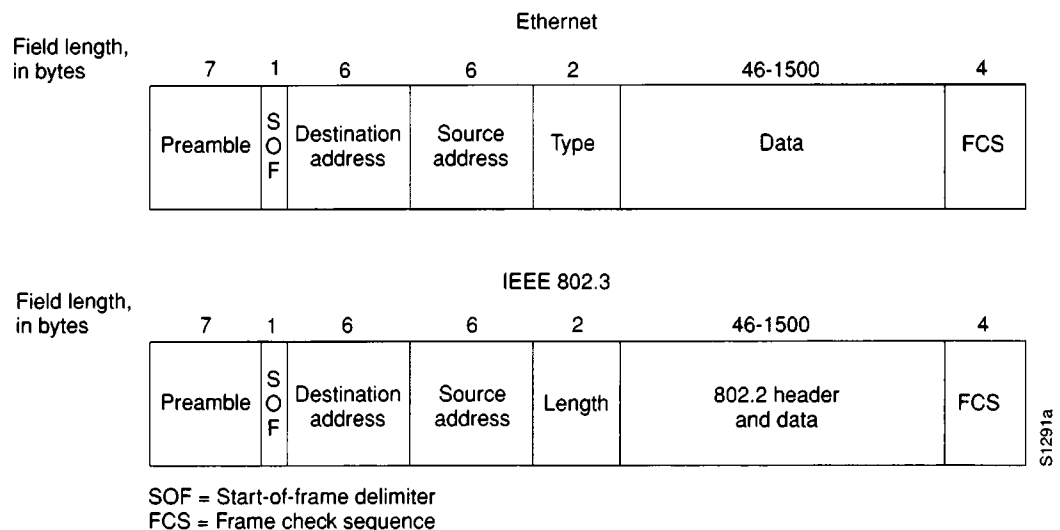


Figure 5-3 Ethernet and IEEE 802.3 Frame Formats

Both Ethernet and IEEE 802.3 frames begin with an alternating pattern of ones and zeros called a *preamble*. The preamble tells receiving stations that a frame is coming.

The last byte before the destination address in an IEEE 802.3 frame is a *start-of-frame delimiter*. This byte ends with two consecutive one bits, which serve to synchronize the frame reception portions of all stations on the LAN.

Immediately following the preamble in both Ethernet and IEEE 802.3 LANs are the *destination* and *source address* fields. Both Ethernet and IEEE 802.3 addresses are six bytes long. Addresses are contained in hardware on the Ethernet and IEEE 802.3 interface cards. The first three bytes of the addresses are specified by the IEEE on a vendor-dependent basis, while the last three bytes are specified by the Ethernet or IEEE 802.3 vendor. The source address is always a unicast (single node) address, while the destination address may be unicast, multicast (group), or broadcast (all nodes).

In Ethernet frames, the 2-byte field following the source address is a *type* field. This field specifies the upper-layer protocol to receive the data after Ethernet processing is complete.

In IEEE 802.3 frames, the 2-byte field following the source address is a *length* field, which indicates the number of bytes of data that follow this field and precede the *frame check sequence* (FCS) field.

Following the type/length field is the actual *data* contained in the frame. After physical-layer and link-layer processing is complete, this data will eventually be sent to an upper-layer protocol. In the case of Ethernet, the upper-layer protocol is identified in the type field. In the case of IEEE 802.3, the upper-layer protocol must be defined within the data portion of the frame, if at all. If data in the frame is insufficient to fill the frame to its minimum 64-byte size, padding bytes are inserted to ensure at least a 64-byte frame.

After the data field is a four-byte FCS field containing a *cyclic redundancy check* (CRC) value. The CRC is created by the sending device and recalculated by the receiving device to check for damage that might have occurred to the frame in transit.

Chapter 6

Token Ring/IEEE 802.5



Background

The Token Ring Network was originally developed by IBM in the 1970s. It is still IBM's primary local area network (LAN) technology, and is second only to Ethernet/IEEE 802.3 in general LAN popularity. The IEEE 802.5 specification is almost identical to, and completely compatible with, IBM's Token Ring Network. In fact, the IEEE 802.5 specification was modeled after IBM Token Ring, and continues to shadow IBM's Token Ring development. The term *Token Ring* is generally used to refer to both IBM's Token Ring Network and IEEE 802.5 networks.

Token Ring/IEEE 802.5 Comparison

Token Ring and IEEE 802.5 networks are basically quite compatible, although the specifications differ in relatively minor ways. IBM's Token Ring Network specifies a star, with all end stations attached to a device called a *multistation access unit (MSAU)*, whereas IEEE 802.5 does not specify a topology (although virtually all IEEE 802.5 implementations also are based on a star). Other differences exist, including media type (IEEE 802.5 does not specify a media type, while IBM Token Ring Networks use twisted pair) and routing information field size (see the discussion on RIFs later in this chapter). Figure 6-1 summarizes IBM Token Ring Network and IEEE 802.5 specifications.

	IBM Token Ring Network	IEEE 802.5
Data rates	4.16 Mbps	4.16 Mbps
Stations/segment	260 (Shielded T.P.) 72 (Unshielded T.P.)	250
Topology	Star	Not specified
Media	Twisted pair	Not specified
Signaling	Baseband	Baseband
Access method	Token passing	Token passing
Encoding	Differential Manchester	Differential Manchester

S1292a

Figure 6-1 IBM Token Ring Network/IEEE 802.5 Comparison

Token Passing

Token Ring and IEEE 802.5 are the primary examples of token-passing networks. Token-passing networks move a small frame, called a token, around the network. Possession of the token grants the right to transmit. If a node receiving the token has no information to send, it simply passes the token to the next end station. Each station can hold the token for a maximum period of time.

If a station possessing the token does have information to transmit, it seizes the token, alters one bit of the token (which turns the token into a start-of-frame sequence), appends the information it wishes to transmit, and finally sends this information to the next station on the ring. While the information frame is circling the ring, there is no token on the network (unless the ring supports *early token release*), so other stations wishing to transmit must wait. Therefore, collisions cannot occur in Token Ring networks. If early token release is supported, a new token can be released when frame transmission is completed.

The information frame circulates the ring until it reaches the intended destination station, which copies the information for further processing. The information frame continues to circle the ring and is finally removed when it reaches the sending station. The sending station can check the returning frame to see whether the frame was seen and subsequently copied by the destination.

Unlike CSMA/CD networks (such as Ethernet), token-passing networks are deterministic. In other words, it is possible to calculate the maximum time that will pass before any end station will be able to transmit. This feature and several reliability features to be discussed later make Token Ring networks ideal for applications where delay must be predictable and robust network operation is important. Factory automation environments are examples of such applications.

Physical Connections

IBM Token Ring Network stations are directly connected to MSAUs, which can be wired together to form one large ring (as shown in Figure 6-2). Patch cables connect MSAUs to adjacent MSAUs. Lobe cables connect MSAUs to stations. MSAUs include bypass relays for removing stations from the ring.

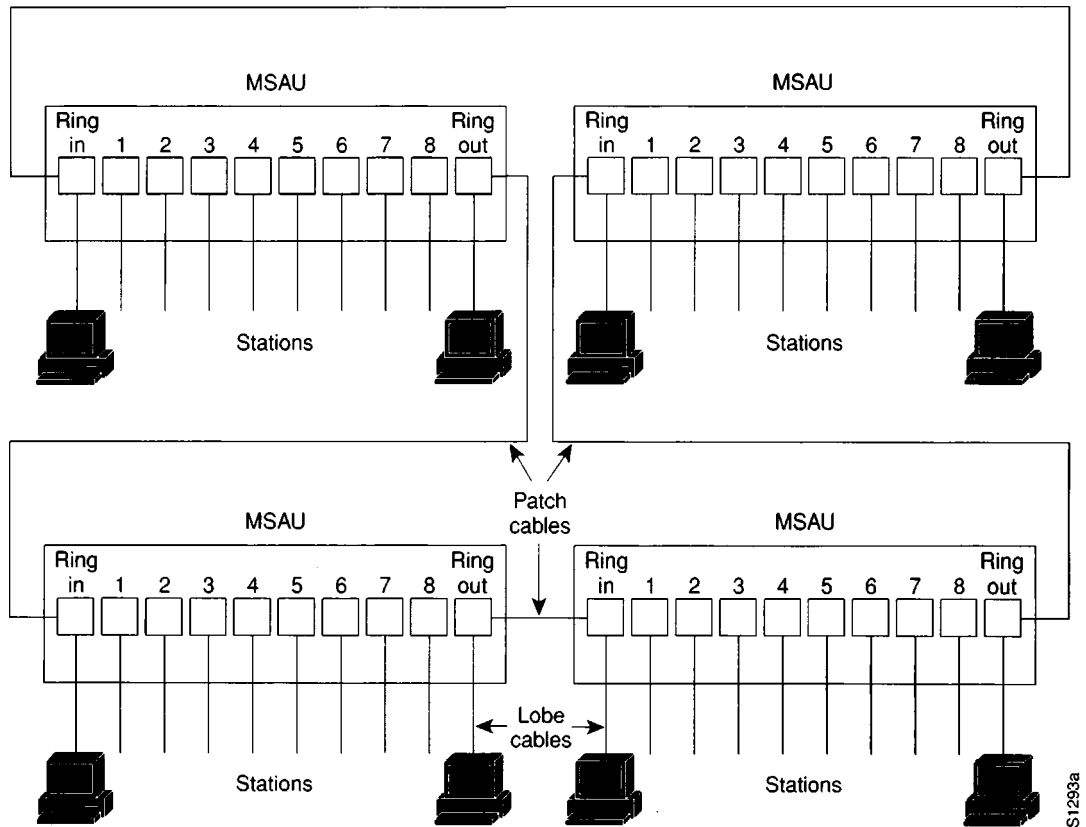


Figure 6-2 IBM Token Ring Network Physical Connections

Priority System

Token Ring networks use a sophisticated priority system that permits certain user-designated, high-priority stations to use the network more frequently. Token Ring frames have two fields that control priority: the *priority* field and the *reservation* field.

Only stations with a priority equal to or higher than the priority value contained in a token can seize that token. Once the token is seized and changed to an information frame, only stations with a priority value higher than that of the transmitting station can reserve the token for the next pass around the network. When the next token is generated, it includes the higher priority of the reserving station. Stations that raise a token's priority level must reinstate the previous priority after their transmission is complete.

Fault Management Mechanisms

Token Ring networks employ several mechanisms for detecting and compensating for network faults. For example, one station in the Token Ring network is selected to be the *active monitor*. This station, which can potentially be any station on the network, acts as a centralized source of timing information for other ring stations and performs a variety of ring maintenance functions. One of these functions is the removal of continuously circulating frames from the ring. When a sending device fails, its frame may continue to circle the ring. This can prevent other stations from transmitting their own frames and essentially lock up the network. The active monitor can detect such frames, remove them from the ring, and generate a new token.

The IBM Token Ring Network's star topology also contributes to overall network reliability. Since all information in a Token Ring network is seen by active MSAUs, these devices can be programmed to check for problems and selectively remove stations from the ring if necessary.

A Token Ring algorithm called *beaconing* detects and tries to repair certain network faults. Whenever a station detects a serious problem with the network (such as a cable break), it sends a beacon frame. The beacon frame defines a failure domain, which includes the station reporting the failure, its *nearest active upstream neighbor (NAUN)*, and everything in between. Beaconing initiates a process called *autoreconfiguration*, where nodes within the failure domain automatically perform diagnostics in an attempt to reconfigure the network around the failed areas. Physically, the MSAU can accomplish this through electrical reconfiguration.

Frame Format

Token Ring networks define two frame types: tokens and data/command frames. Both formats are shown in Figure 6-3.

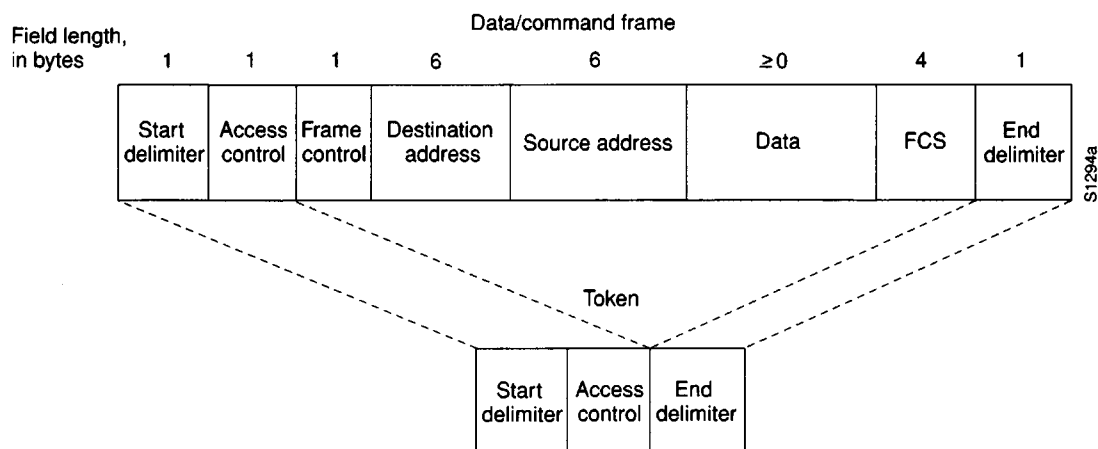


Figure 6-3 IEEE 802.5/Token Ring Frame Formats

Tokens

Tokens are three bytes in length and consist of a start delimiter, an access control byte, and an end delimiter.

The *start delimiter* serves to alert each station to the arrival of a token (or data/command frame). This field includes signals that distinguish the byte from the rest of the frame by violating the encoding scheme used elsewhere in the frame.

The *access control byte* contains the priority and reservation fields, as well as a token bit (used to differentiate a token from a data/command frame) and a monitor bit (used by the active monitor to determine whether a frame is circling the ring endlessly).

Finally, the *end delimiter* signals the end of the token or data/command frame. It also contains bits to indicate a damaged frame and a frame that is the last in a logical sequence.

Data/Command Frames

Data/command frames vary in size, depending on the size of the information field. Data frames carry information for upper-layer protocols; command frames contain control information and have no data for upper-layer protocols.

In data/command frames, a *frame control* byte follows the access control byte. The frame control byte indicates whether the frame contains data or control information. In control frames, this byte specifies the type of control information.

Following the frame control byte are the two *address* fields, which identify the destination and source stations. As with IEEE 802.3, addresses are 6 bytes in length.

The *data* field follows the address fields. The length of this field is limited by the ring token holding time, which defines the maximum time a station may hold the token.

Following the data field is the *frame check sequence (FCS)* field. This field is filled by the source station with a calculated value dependent on the frame contents. The destination station recalculates the value to determine whether the frame may have been damaged in transit. If so, the frame is discarded.

As with the token, the *end delimiter* completes the data/command frame.

Chapter 7

FDDI



Background

The Fiber Distributed Data Interface (FDDI) standard was produced by the ANSI X3T9.5 standards committee in the mid-1980s. During this period, high-speed engineering workstations were beginning to tax the capabilities of existing local area networks (LANs) (primarily Ethernet and Token Ring). A new LAN was needed that could easily support these workstations and their new distributed applications. At the same time, network reliability was becoming an increasingly important issue as system managers began to migrate mission-critical applications from large computers to networks. FDDI was created to fill these needs.

After completing the FDDI specification, ANSI submitted FDDI to ISO. ISO has created an international version of FDDI that is completely compatible with the ANSI standard version.

Today, although FDDI implementations are not as common as Ethernet or Token Ring, FDDI has gained a substantial following that continues to increase as the cost of FDDI interfaces diminishes. FDDI is frequently used as a backbone technology as well as a means to connect high-speed computers in a local area.

Technology Basics

FDDI specifies a 100-Mbps, token-passing dual-ring LAN using a fiber-optic transmission medium. It defines the physical layer and media-access portion of the link layer, and so is roughly analogous to IEEE 802.3 and IEEE 802.5 in its relationship to the OSI reference model.

Although it operates at faster speeds, FDDI is similar in many ways to Token Ring. The two networks share many features, including topology (ring), media-access technique (token passing), reliability features (beaconing, for example), and others. Refer to Chapter 6, “Token Ring/IEEE 802.5,” for more information on Token Ring and related technologies.

One of FDDI’s most important characteristics is its use of optical fiber as a transmission medium. Optical fiber offers several advantages over traditional copper wiring, including security (fiber does not emit electrical signals that can be tapped), reliability (fiber is immune to electrical interference), and speed (optical fiber has much higher throughput potential than copper cable).

FDDI defines use of two types of fiber: *single mode* (sometimes called *monomode*) and *multimode*. Modes can be thought of as bundles of light rays entering the fiber at a particular angle. Single-mode fiber allows only one mode of light to propagate through the fiber, while multimode fiber allows multiple modes of light to propagate through the fiber. Because multiple modes of light propagating through the fiber may travel different distances (depending on the entry angles), causing them to arrive at the destination at different times (a phenomenon called *modal dispersion*), single-mode fiber is capable of higher bandwidth and greater cable run distances than multimode fiber. Due to these characteristics, single-mode fiber is often used for campus backbones, while multimode fiber is often used for workgroup connectivity. Multimode fiber uses light-emitting diodes (LEDs) as the light-generating devices, while single-mode fiber generally uses lasers.

FDDI Specifications

FDDI is defined by four separate specifications (see Figure 7-1):

- *Media Access Control (MAC)*—Defines how the medium is accessed, including packet format, token handling, addressing, CRC algorithm, and error recovery mechanisms.
- *Physical Layer Protocol (PHY)*—Defines data encoding/decoding procedures, clocking requirements, framing, and other functions.
- *Physical Layer Medium Dependent (PMD)*—Defines the characteristics of the transmission medium, including the fiber optic link, power levels, bit error rates, optical components, and connectors.
- *Station Management (SMT)*—Defines the FDDI station configuration, ring configuration, and ring control features, including station insertion and removal, initialization, fault isolation and recovery, scheduling, and collection of statistics.

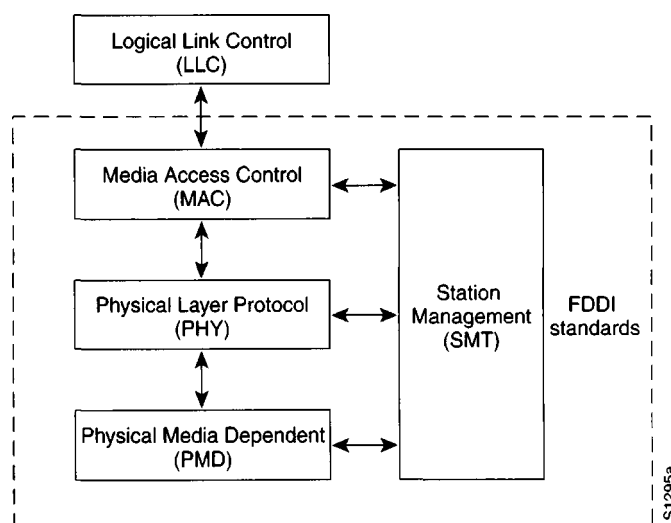


Figure 7-1 FDDI Standards

Physical Connections

FDDI specifies the use of dual rings. Traffic on these rings travels in opposite directions. Physically, the rings consist of two or more point-to-point connections between adjacent stations. One of the two FDDI rings is called the *primary* ring; the other is called the *secondary* ring. The primary ring is used for data transmission, while the secondary ring is generally used as a backup.

Class B or *single attached stations (SAS)* attach to one ring; *Class A* or *dual attached stations (DAS)* attach to both rings. SASs are attached to the primary ring through a *concentrator*, which provides connections for multiple SASs. The concentrator ensures that failure or power down of any given SAS does not interrupt the ring. This is particularly useful when PCs or similar devices that power on and off frequently connect to the ring.

A typical FDDI configuration with both DASs and SASs is shown in Figure 7-2.

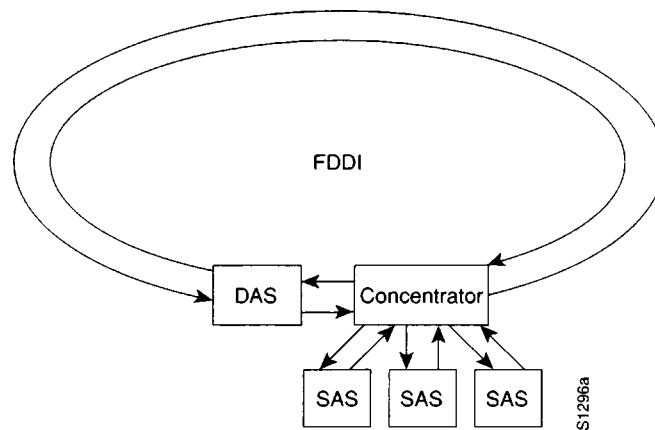


Figure 7-2 FDDI Nodes: DAS, SAS, and Concentrator

Each FDDI DAS has two ports, designated A and B. These ports connect the station to the dual FDDI ring. Therefore, each port provides a connection for both the primary and the secondary ring, as shown in Figure 7-3.

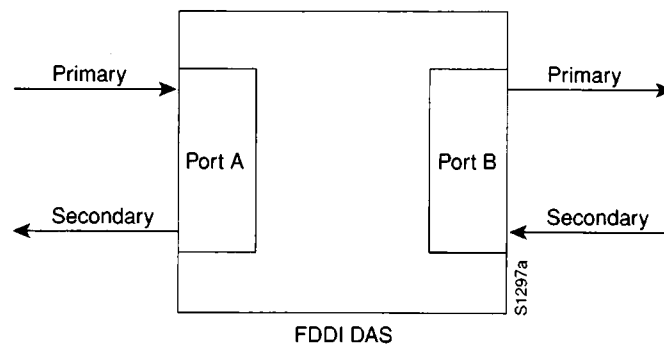


Figure 7-3 FDDI DAS Ports

Traffic Types

FDDI supports real-time allocation of network bandwidth, making it ideal for a variety of different application types. FDDI provides this support by defining two types of traffic: *synchronous* and *asynchronous*. Synchronous traffic can consume a portion of the 100 Mbps total bandwidth of an FDDI network, while asynchronous traffic can consume the rest. Synchronous bandwidth is allocated to those stations requiring continuous transmission capability. Such capability is useful for transmitting voice and video information, for example. Other stations use the remaining bandwidth asynchronously. FDDI's SMT specification defines a distributed bidding scheme to allocate FDDI bandwidth.

Asynchronous bandwidth is allocated using an eight-level priority scheme. Each station is assigned an asynchronous priority level. FDDI also permits extended dialogues, where stations may temporarily use all asynchronous bandwidth. FDDI's priority mechanism can essentially lock out stations that cannot use synchronous bandwidth and have too low an asynchronous priority.

Fault-Tolerant Features

FDDI provides a number of fault-tolerant features. The primary fault-tolerant feature is the dual ring. If a station on the dual ring fails or is powered down or if the cable is damaged, the dual ring is automatically "wrapped" (doubled back onto itself) into a single ring, as shown in Figure 7-4. In this figure, when Station 3 fails, the dual ring is automatically wrapped in Stations 2 and 4, forming a single ring. Although Station 3 is no longer on the ring, network operation continues for the remaining stations.

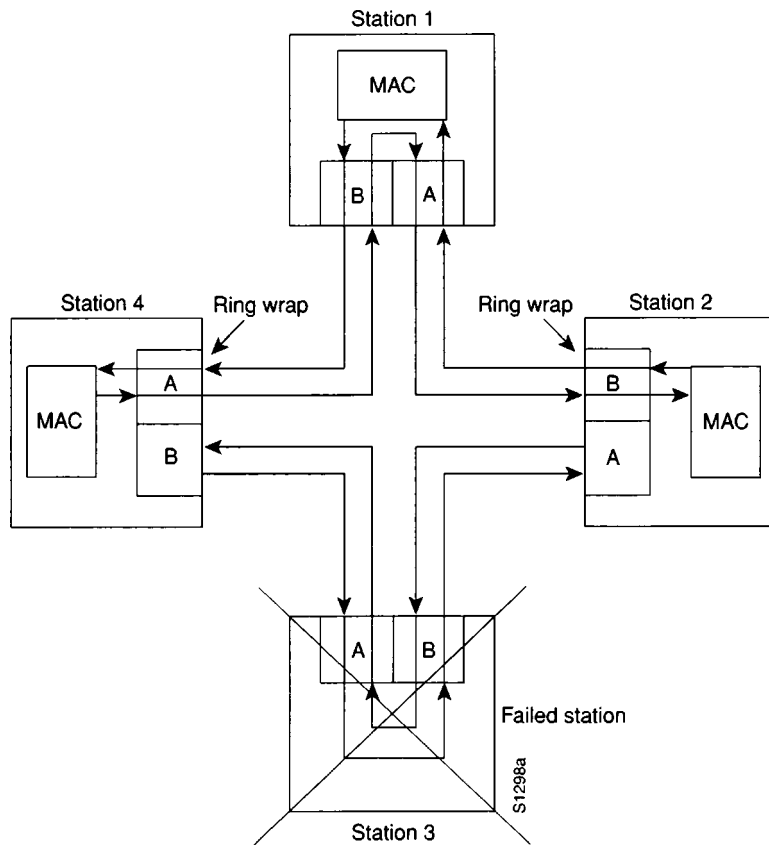


Figure 7-4 Station Failure, Ring Recovery Configuration

Figure 7-5 shows how FDDI compensates for a wiring failure. Stations 3 and 4 wrap the ring within themselves when wiring between these stations fails.

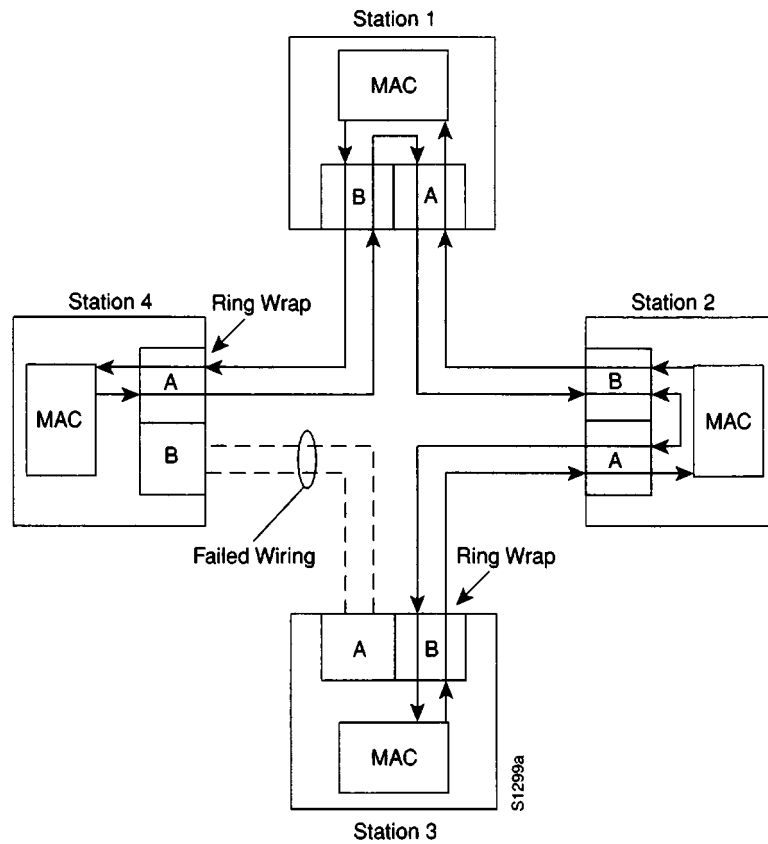


Figure 7-5 Failed Wiring, Ring Recovery Configuration

As FDDI networks grow, the possibility of multiple ring failures grows. When two ring failures occur, the ring will be wrapped in both cases, effectively segmenting the ring into two separate rings that cannot communicate with each other. Subsequent failures cause additional ring segmentation.

Optical bypass switches can be used to prevent ring segmentation by eliminating failed stations from the ring. This is shown in Figure 7-6.

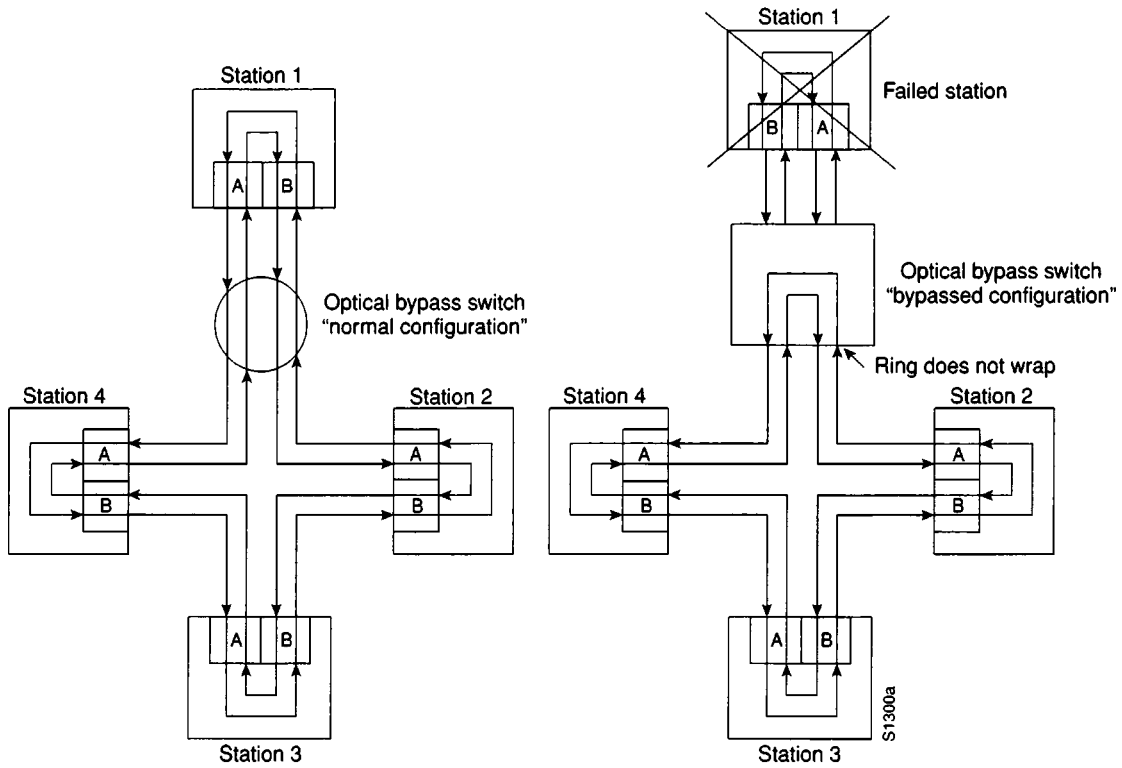


Figure 7-6 Use of Optical Bypass Switch

Critical devices such as routers or mainframe hosts can use another fault-tolerant technique called *dual homing* to provide additional redundancy and help guarantee operation. In dual homing situations, the critical device is attached to two concentrators. One pair of concentrator links is declared the active link; the other pair is declared passive. The passive link stays in backup mode until the primary link (or the concentrator to which it is attached) is determined to have failed. When this occurs, the passive link is automatically activated.

Frame Format

FDDI frame formats (shown in Figure 7-7) are similar to those of Token Ring.

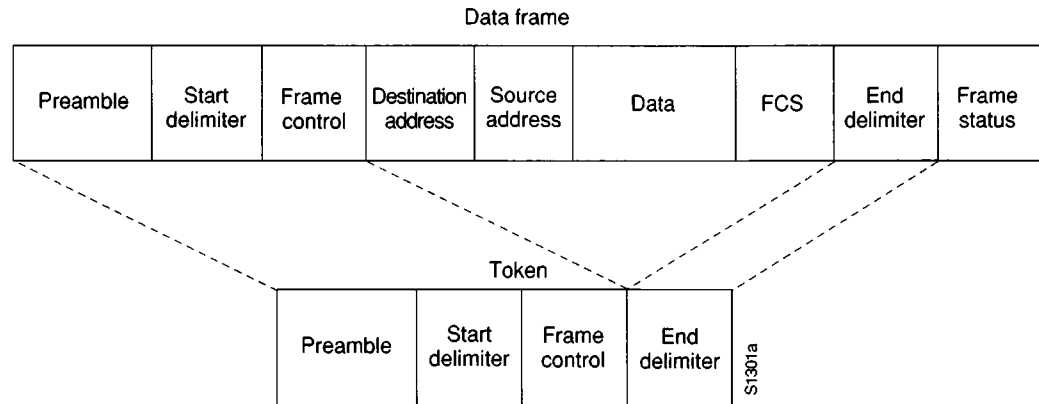


Figure 7-7 FDDI Frame Format

The *preamble* prepares each station for the upcoming frame.

The *start delimiter* indicates the beginning of the frame. It consists of signaling patterns that differentiate it from the rest of the frame.

The *frame control* field indicates the size of the address fields, whether the frame contains asynchronous or synchronous data, and other control information.

As with Ethernet and Token Ring, FDDI addresses are six bytes. The *destination address* field can contain a unicast (singular), multicast (group), or broadcast (every station) address, while the *source address* identifies the single station that sent the frame.

The *data* field contains either information destined for an upper-layer protocol or control information.

As with Token Ring and Ethernet, the *frame check sequence (FCS)* field is filled by the source station with a calculated *cyclic redundancy check (CRC)* value dependent on the frame contents. The destination station recalculates the value to determine whether the frame may have been damaged in transit. If so, the frame is discarded.

The *end delimiter* contains nondata symbols that indicate the end of the frame.

The *frame status* field allows the source station to determine if an error occurred and if the frame was recognized and copied by a receiving station.

Chapter 8

UltraNet

8

Background

The UltraNet network system, or simply UltraNet, consists of a family of high-speed networking software and hardware products capable of offering an aggregate throughput of one gigabit per second (Gbps). UltraNet is manufactured and marketed by Ultra Network Technologies. UltraNet is typically used to link very fast computer systems such as supercomputers, minisupercomputers, mainframes, servers, and workstations. UltraNet can itself be connected to other networks (for example, Ethernet and Token Ring) through routers that provide gateway functions.

Technology Basics

UltraNet provides services corresponding to the lower four layers of the OSI reference model. Figure 8-1 shows the relationship between these layers and the UltraNet implementation. In addition to the protocols listed, UltraNet also supports the *Simple Network Management Protocol (SNMP)* and the *Routing Information Protocol (RIP)*. For more information on these protocols, see Chapter 23, "RIP," and Chapter 32, "SNMP," respectively.

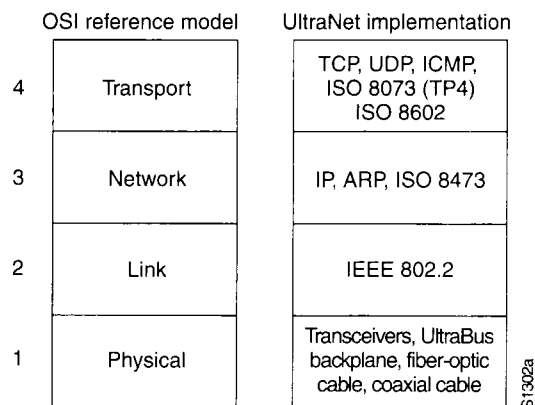


Figure 8-1 UltraNet and the OSI Reference Model

UltraNet uses a star topology with a network hub at the star's focal point. Other components of the UltraNet system include host software, network processors, link adapters, network management tools, and internetworking products such as routers and bridges. Network processors connect hosts to the UltraNet system and provide virtual circuit and datagram services. Hosts directly connected to the UltraNet system can be up to 30 kilometers from each other. This range can be extended through connection to a wide area network (WAN) using, for example, T3 links. Figure 8-2 provides a diagram of the UltraNet system.

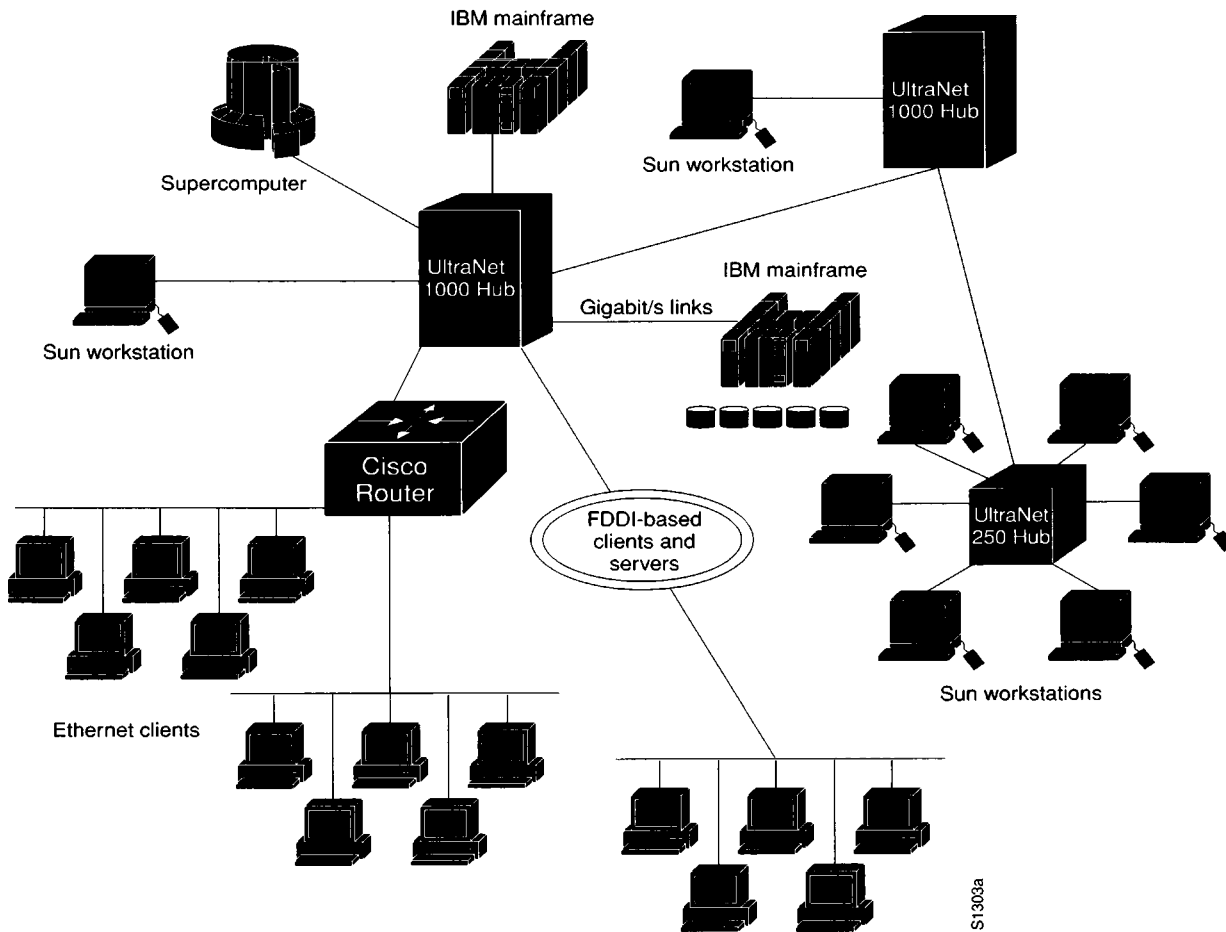


Figure 8-2 UltraNet Network System

UltraNet Components

UltraNet networks consist of various components, including hubs, host software, network managers, network processors, and link adapters. These system elements are described in the following sections.

UltraNet Hub

The UltraNet hub is the central connection point for hosts on an UltraNet network. It contains a high-speed parallel internal bus (the UltraBus) connecting all processors within the hub. The UltraBus is responsible for switching data on the UltraNet network. UltraNet hubs provide speed matching, congestion management, and direct channel attachment.

UltraNet Host Software

UltraNet host software includes:

- Programming libraries allowing standard *Transmission Control Protocol/Internet Protocol (TCP/IP)* client programs and graphics application programs to run across UltraNet.
- Network processor device drivers that provide the interface between user processes and the UltraNet network processor through the processor adapter.
- Support for UNIX *Berkeley Standard Distribution (BSD)* socket library-based applications—This support is delivered in the form of a collection of C language library functions that replace standard socket system calls to provide compatibility with existing socket-based applications.
- Configuration utilities that allow users to define the network processors present in the UltraNet system, to define routes between UltraNet hubs and network processors, and to define UltraNet addresses.
- Diagnostic utilities that allow users to check the UltraNet system for possible problems. These utilities can be run from the Ultra Network Manager computer as well as from the host.

Ultra Network Manager

The Ultra Network Manager provides tools that help initialize and monitor the UltraNet. Physically, the manager is an Intel 80386-based PC running DOS and Windows that attaches to the UltraNet hub through a *network management bus (NMB)*. The NMB is an independent 1-Mbps LAN based on the *StarLAN (1Base5)* specification. The Ultra Network Manager exchanges management information using SNMP.

Network Processors

UltraNet network processors provide connections between UltraNet hubs and hosts. Network processors are available that support the *High-Performance Parallel Interface (HIPPI)*, *HSX* (supported by Cray), *BMC* (supported by IBM), and *LSC* (supported by Cray) channels, as well as the *VMEbus*, *SBus*, *HP/EISA* bus, and the *IBM Micro Channel* bus. Network processors can reside either in the host computer or in an UltraNet hub.

Each hub-resident network processor consists of a *protocol processor* board, a *personality module* board, and a *paddlecard*. The protocol processor board executes network protocols and contains FIFO buffers to perform packet buffering and speed matching. The personality module board manages information exchange between the protocol processor and different network media, host channels, or specialized hardware. The paddlecard manages input/output (I/O) between the network processor and the host computer, graphics display monitor, or another hub.

UltraNet also offers a high-resolution graphics display system that accepts pixel data from a host on the UltraNet and displays it on a monitor connected to the adapter. This device is called a *frame buffer* network processor.

Most network protocol processing tasks are handled by the UltraNet network processors. Network processors can host implementations of TCP/IP and related protocols as well as a modified OSI protocol stack to effect communication between hosts.

Link Adapters

Link adapters connect and transfer data between two UltraNet hubs or between an UltraNet hub and a Cisco Systems AGS+ router. Consisting of link controllers, one to four link multiplexers, and one paddlecard for each link multiplexer, link adapters have a full-duplex private bus with a 1-gigabit per second bandwidth capacity.

On a regular basis, link adapters determine the adapters and hubs to which they are directly connected. Link adapters forward this and other routing information to other link adapters to dynamically build and maintain a routing database containing best-path information to all hosts within the network.

Chapter 9

HSSI



9

Background

Increasing communication speeds is an undeniable networking trend. Local area networks (LANs) have recently moved into the 100-Mbps range with *Fiber Distributed Data Interface (FDDI)*. Local applications driving these speed increases include imaging, video, and today's distributed (client-server) data transmission applications. Faster computer platforms will continue to drive rates up in the local environment as they make new, high-speed applications possible.

Higher-throughput wide area network (WAN) pipes have been developed to match the ever-increasing LAN speeds and to allow mainframe channel extension over WANs. WAN technologies such as *Frame Relay*, *Switched Multimegabit Data Service (SMDS)*, *Synchronous Optical Network (SONET)*, and *Broadband Integrated Services Digital Network (Broadband ISDN, or simply BISDN)* take advantage of new digital and fiber-optic technologies to ensure that WANs are not a significant bottleneck in end-to-end communication over large geographic areas. See Chapter 14, "Frame Relay," and Chapter 15, "SMDS," for more information on Frame Relay and SMDS, respectively.

With higher speeds being achieved in both the local and the wide-area environments, a *data terminal equipment (DTE)/data circuit-terminating equipment (DCE)* interface that could bridge these two worlds without becoming a bottleneck became a critical need. Classical DTE/DCE interface standards such as RS-232 and V.35 are not capable of supporting T3 or similar rates. By the late 1980s, it was clear that a new DTE/DCE protocol was needed.

The *High-Speed Serial Interface (HSSI)* is a DTE/DCE interface developed by Cisco Systems and T3Plus Networking to address the previously stated needs. The HSSI specification is available to any organization wishing to implement HSSI. So far, over 150 copies of the specification have been distributed, and dozens of companies either have or are currently implementing an HSSI solution. In less than three years, HSSI has become a de facto industry standard.

HSSI is now in the American National Standards Institute (ANSI) Electronic Industries Association (EIA)/TIA TR.30.2 committee for formal standardization. It has recently moved into the Consultative Committee for International Telegraph and Telephone (CCITT) and International Organization for Standardization (ISO) organizations as well, and is expected to be standardized by these bodies.

Technology Basics

HSSI defines both the electrical and the physical DTE/DCE interface. It therefore corresponds to the physical layer of the OSI reference model. HSSI technical characteristics are summarized in Figure 9-1.

HSSI technical characteristics

Max. signaling rate	52 Mbps
Max. cable length	50 feet
Connector pins	50
Interface	DTE-DCE
Electrical technology	Differential ECL
Typical power consumption	610 mW
Topology	Point-to-point
Cable type	Shielded twisted pair

S1304a

Figure 9-1 HSSI Technical Characteristics

The maximum signaling rate of HSSI is 52 Mbps. At this rate, HSSI can handle the T3 speeds (45 Mbps) of many of today's fast WAN technologies, the *Office Channel (OC)-1* speeds (52 Mbps) of the *synchronous digital hierarchy (SDH)*, and can easily provide high-speed connectivity between LANs such as Token Ring and Ethernet.

The use of differential *emitter-coupled logic (ECL)* helps HSSI achieve high data rates and low noise levels. ECL has been used in Cray interfaces for years, and is also specified by the ANSI *High-Performance Parallel Interface (HIPPI)* communications standard for supercomputer LAN communications. ECL is off-the-shelf technology that permits excellent retiming on the receiver, resulting in reliable timing margins.

The flexibility of HSSI's clock and data signaling protocol makes user (or vendor) bandwidth allocation possible. The DCE controls the clock by changing its speed or by deleting clock pulses. In this way, the DCE can allocate bandwidth between applications. For example, a PBX may require a particular amount of bandwidth, a router another amount, and a channel extender a third amount. Bandwidth allocation is key to making T3 and other broadband services affordable and popular.

HSSI uses a subminiature, FCC approved 50-pin connector that is smaller than its V.35 counterpart. To reduce the need for male-male and female-female adapters, HSSI cable connectors are specified as male. The HSSI cable uses the same number of pins and wires as the *Small Computer Systems Interface 2 (SCSI-2)* cable, but the HSSI electrical specification is tighter.

For a high level of diagnostic input, HSSI provides four loopback tests. These tests are shown in Figure 9-2. The first provides a local cable test, as the signal loops back once it reaches the DTE port. The second test reaches the line port of the local DCE. The third test reaches the line port of the remote DCE. Finally, the fourth test is a DCE-initiated test of the DTE's DCE port.

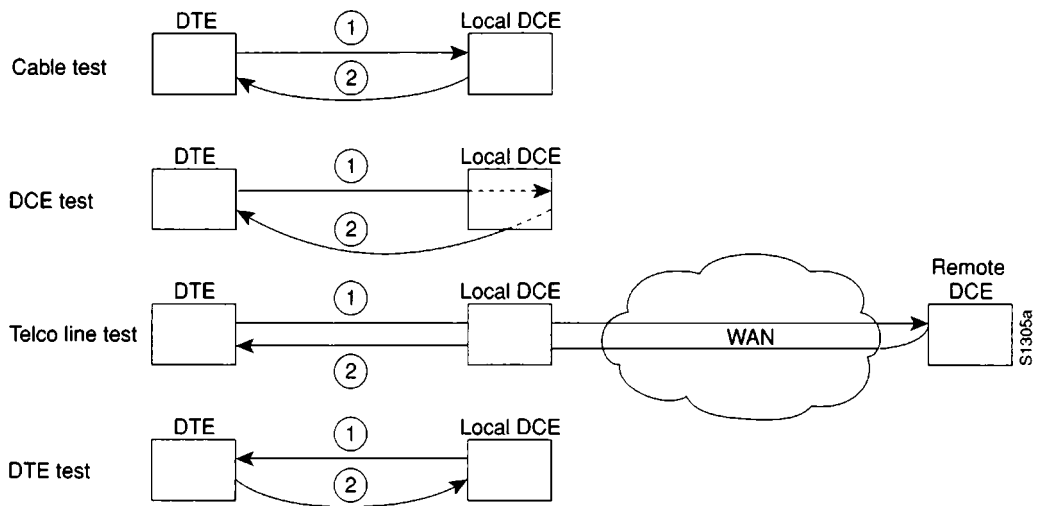


Figure 9-2 HSSI's Four Loopback Tests

HSSI assumes a peer-to-peer intelligence in the DCE and DTE. The control protocol is simplified, with just two control signals required ("DTE available" and "DCE available"). Both signals must be asserted before the data circuit is valid. The DCE and DTE are expected to be able to manage the networks behind their interfaces. Reducing the number of control signals improves circuit reliability by reducing the number of circuits that can fail.

Chapter 10

PPP

10

Background

In the late 1980s, the Internet (a large international network connecting many research organizations, universities, and commercial concerns) began to experience explosive growth in the number of hosts supporting TCP/IP. The vast majority of these hosts were connected to local area networks (LANs) of various types, Ethernet being the most common. Most of the other hosts were connected through wide area networks (WANs) such as X.25-style *public data networks* (PDNs). Relatively few of these hosts were connected with simple point-to-point (that is, serial) links. Yet point-to-point links are among the oldest methods of data communications and almost every host supports point-to-point connections. For example, asynchronous RS-232-C interfaces are essentially ubiquitous.

One reason for the small number of point-to-point IP links was the lack of a standard Internet encapsulation protocol. The Point-to-Point Protocol (PPP) was designed to solve this problem. In addition to solving the problem of standardized Internet encapsulation of IP over point-to-point links, PPP was also designed to address other issues, including assignment and management of IP addresses, asynchronous (start/stop) and bit-oriented synchronous encapsulation, network protocol multiplexing, link configuration, link quality testing, error detection, and option negotiation for such capabilities as network-layer address negotiation and data compression negotiation. PPP addresses these issues by providing an extensible *Link Control Protocol* (LCP) and a family of *Network Control Protocols* (NCPs) to negotiate optional configuration parameters and facilities. Today, PPP supports other protocols besides IP, including IPX and DECnet.

PPP Components

PPP provides a method for transmitting datagrams over serial point-to-point links. It has three main components:

- A method for encapsulating datagrams over serial links—PPP uses the *High-level Data Link Control* (HDLC) protocol as a basis for encapsulating datagrams over point-to-point links. See Chapter 12, “SDLC and Derivatives,” for more information on HDLC.
- An extensible LCP to establish, configure, and test the data-link connection.
- A family of NCPs for establishing and configuring different network-layer protocols. PPP is designed to allow the simultaneous use of multiple network-layer protocols.

General Operation

In order to establish communications over a point-to-point link, the originating PPP first sends LCP packets to configure and (optionally) test the data link. After the link has been established and optional facilities have been negotiated as needed by the LCP, the originating PPP sends NCP packets to choose and configure one or more network-layer protocols. Once each of the chosen network-layer protocols has been configured, datagrams from each network-layer protocol can be sent over the link. The link will remain configured for communications until explicit LCP or NCP packets close the link, or until some external event occurs (for example, an inactivity timer expires or a user intervenes).

Physical-Layer Requirements

PPP is capable of operating across any DTE/DCE interface (for example, EIA RS-232-C, EIA RS-422, EIA RS-423 and CCITT V.35). The only absolute requirement imposed by PPP is the provision of a duplex circuit, either dedicated or switched, that can operate in either an asynchronous or synchronous bit-serial mode, transparent to PPP data-link layer frames. PPP does not impose any restrictions regarding transmission rate, other than those imposed by the particular DTE/DCE interface in use.

The PPP Data-Link Layer

PPP uses the principles, terminology, and frame structure of the International Organization for Standardization's (ISO's) HDLC procedures (ISO 3309-1979), as modified by ISO 3309:1984/PDAD1 "Addendum 1: Start/stop transmission." ISO 3309-1979 specifies the HDLC frame structure for use in synchronous environments. ISO 3309:1984/PDAD1 specifies proposed modifications to ISO 3309-1979 to allow its use in asynchronous environments. The PPP control procedures use the definitions and control field encodings standardized in ISO 4335-1979 and ISO 4335-1979/Addendum 1-1979.

The PPP frame format appears in Figure 10-1.

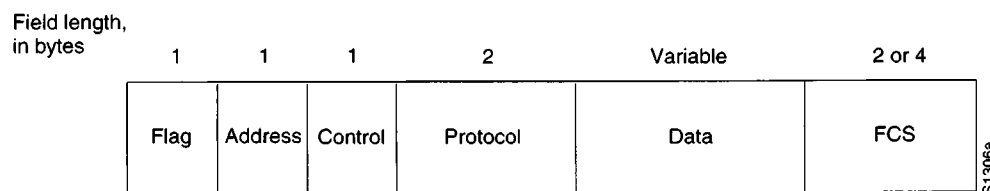


Figure 10-1 PPP Frame Format

The *flag* sequence is a single byte long and indicates the beginning or end of a frame. The flag sequence consists of the binary sequence 01111110.

The *address* field is a single byte long and contains the binary sequence 11111111, the standard broadcast address. PPP does not assign individual station addresses.

The *control* field is a single byte and contains the binary sequence 00000011, which calls for transmission of user data in an unsequenced frame. A connectionless link service similar to that of LLC Type 1 is provided. Refer to Chapter 12, “SDLC and Derivatives,” for more information on LLC types and frame types.

The *protocol* field is two bytes long and its value identifies the protocol encapsulated in the information field of the frame. The most up-to-date values of the protocol field are specified in the most recent *Assigned Numbers Request for Comments (RFC)*.

The *data* field is zero or more bytes in length, and contains the datagram for the protocol specified in the protocol field. The end of the information field is found by locating the closing flag sequence and allowing two bytes for the FCS field. The default maximum length of the information field is 1500 bytes. By prior agreement, consenting PPP implementations can use other values for the maximum information field length.

The *frame check sequence (FCS)* field is normally 16 bits (two bytes). By prior agreement, consenting PPP implementations can use a 32-bit (four-byte) FCS for improved error detection.

The LCP can negotiate modifications to the standard PPP frame structure. However, modified frames will always be clearly distinguishable from standard frames.

The PPP Link Control Protocol (LCP)

The LCP provides a method of establishing, configuring, maintaining and terminating the point-to-point connection. LCP goes through four distinct phases:

- Link establishment and configuration negotiation—Before any network-layer datagrams (for example, IP) can be exchanged, LCP must first open the connection and negotiate configuration parameters. This phase is complete when a configuration acknowledgment packet has been both sent and received.
- Link quality determination—LCP allows an optional link quality determination phase following the link establishment and configuration negotiation phase. In this phase, the link is tested to determine if the link quality is sufficient to bring up network-layer protocols. This phase is completely optional. LCP can delay transmission of network-layer protocol information until this phase is completed.
- Network-layer protocol configuration negotiation—Once LCP has finished the link quality determination phase, network-layer protocols can be separately configured by the appropriate NCP and can be brought up and taken down at any time. If LCP closes the link, it informs the network-layer protocols so that they may take appropriate action.

- Link termination—LCP can terminate the link at any time. This will usually be done at the request of a human user, but can happen because of a physical event such as the loss of carrier, or the expiration of an idle-period timer.

There are three classes of LCP packets:

- Link establishment packets—Used to establish and configure a link.
- Link termination packets—Used to terminate a link.
- Link maintenance packets—Used to manage and debug a link.

These packets are used to accomplish the work of each of the LCP phases.

Chapter 11

ISDN

11

Background

Integrated Services Digital Network (ISDN) refers to a set of digital services becoming available to end users. ISDN involves the digitization of the telephone network so that voice, data, text, graphics, music, video, and other source material can be provided to end users from a single end-user terminal over existing telephone wiring. Proponents of ISDN imagine a worldwide network much like the present telephone network, except that digital transmission is used and a variety of new services are available.

ISDN is an effort to standardize subscriber services, user/network interfaces, and network and internetwork capabilities. Standardizing subscriber services attempts to ensure a level of international compatibility. Standardizing the user/network interface stimulates development and marketing of these interfaces by third-party manufacturers. Standardizing network and internetwork capabilities helps achieve the goal of worldwide connectivity by ensuring that ISDN networks easily communicate with one another.

ISDN applications include high-speed image applications (such as Group IV facsimile), additional telephone lines in homes to serve the telecommuting industry, high-speed file transfer, and video conferencing. Voice, of course, will also be a popular application for ISDN.

Many carriers are beginning to offer ISDN under tariff. In North America, large *local-exchange carriers (LECs)* are beginning to provide ISDN service as an alternative to the T1 connections that currently carry bulk *wide-area telephone service (WATS)* services.

Components

ISDN components include terminals, terminal adapters (TAs), network-termination devices, line-termination equipment, and exchange-termination equipment. ISDN terminals come in two types. Specialized ISDN terminals are referred to as *terminal equipment type 1 (TE1)*. Non-ISDN terminals such as DTE that predate the ISDN standards are referred to as *terminal equipment type 2 (TE2)*. TE1s connect to the ISDN network through a four-wire twisted-pair digital link. TE2s connect to the ISDN network through a terminal adapter. The ISDN TA can either be a standalone device or a board inside the TE2. If implemented as a standalone device, the TE2 connects to the TA via a standard physical-layer interface (for example, EIA232, V.24, or V.35).

Beyond the TE1 and TE2 devices, the next connection point in the ISDN network is the NT1 or NT2. These are network-termination devices that connect the four-wire subscriber wiring to the conventional two-wire local loop. In North America, the NT1 is a *customer premises equipment (CPE)* device. In most other parts of the world, the NT1 is part of the network provided by the carrier. The NT2 is a more complicated device, typically found in digital *private branch exchanges (PBXs)*, that performs Layer 2 and 3 protocol functions and concentration services. An NT1/2 device also exists; it is a single device that combines the functions of an NT1 and an NT2.

A number of reference points are specified in ISDN. These reference points define logical interfaces between functional groupings such as TAs and NT1s. ISDN reference points include *R* (the reference point between non-ISDN equipment and a TA), *S* (the reference point between user terminals and the NT2), *T* (the reference point between NT1 and NT2 devices), and *U* (the reference point between NT1 devices and line-termination equipment in the carrier network). The *U* reference point is relevant only in North America, where the NT1 function is not provided by the carrier network.

A sample ISDN configuration is shown in Figure 11-1. This figure shows three devices attached to an ISDN switch at the central office. Two of these devices are ISDN-compatible, so they can be attached through an *S* reference point to NT2 devices. The third device (a standard, non-ISDN telephone) attaches through the *R* reference point to a TA. Any of these devices could also attach to a NT1/2 device, which would replace both the NT1 and the NT2. And, although they are not shown, similar user stations are attached to the right-most ISDN switch.

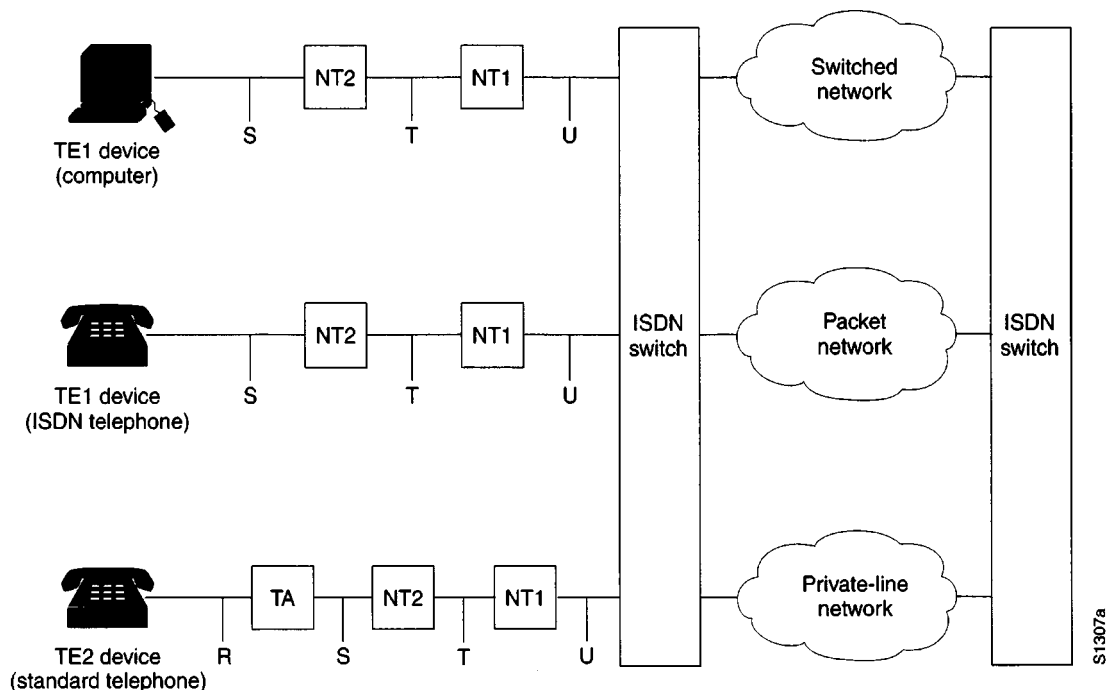


Figure 11-1 Sample ISDN Configuration

Services

ISDN's *Basic Rate Interface (BRI)* service offers two *B channels* and one *D channel (2B+D)*. BRI B-channel service operates at 64 Kbps and is meant to carry user data; BRI D-channel service operates at 16 Kbps and is meant to carry control and signaling information, although it can support user data transmission under certain circumstances. The D channel signaling protocol comprises Layers 1 through 3 of the OSI reference model. BRI also provides for framing control and other overhead, bringing its total bit rate to 192 Kbps. The BRI physical layer specification is CCITT I.430.

ISDN *Primary Rate Interface (PRI)* service offers 23 B channels and one D channel in North America and Japan, yielding a total bit rate of 1.544 Mbps (the PRI D channel runs at 64 Kbps). ISDN PRI in Europe, Australia, and other parts of the world provides 30 B plus one 64-Kbps D channel and a total interface rate of 2.048 Mbps. The PRI physical-layer specification is CCITT I.431.

Layer 1

ISDN physical-layer (Layer 1) frame formats differ depending on whether the frame is outbound (from terminal to network) or inbound (from network to terminal). Both physical-layer interfaces are shown in Figure 11-2. The frames are 48 bits long, of which 36 bits represent data. The F bits provide synchronization. The L bits adjust the average bit value. The E bits are used for contention resolution when several terminals on a passive bus contend for a channel. The A bit activates devices. The S bits have not yet been assigned. The B1, B2, and D bits are for user data.

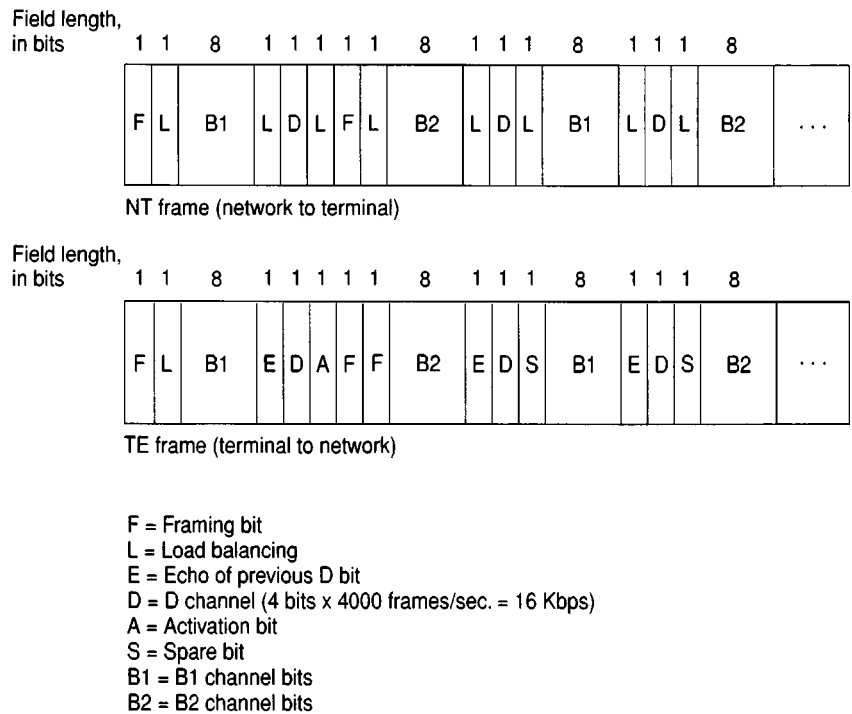


Figure 11-2 ISDN Physical-Layer Frame Formats

Multiple ISDN user devices can be physically attached to one circuit. In this configuration, collisions can result if two terminals transmit simultaneously. ISDN therefore provides features to determine link contention. When an NT receives a D bit from the TE, it echoes back the bit in the next E-bit position. The TE expects the next E bit to be the same as its last transmitted D bit.

Terminals cannot transmit into the D channel unless they first detect a specific number of ones (indicating “no signal”) corresponding to a pre-established priority. If the TE detects a bit in the echo (E) channel that is different from its D bits, it must stop transmitting immediately. This simple technique ensures that only one terminal can transmit its D message at one time. After successful D message transmission, the terminal has its priority reduced by requiring it to detect more continuous ones before transmitting. Terminals may not raise their priority until all other devices on the same line have had an opportunity to send a D message. Telephone connections have higher priority than all other services, and signaling information has a higher priority than nonsignaling information.

Layer 2

Layer 2 of the ISDN signaling protocol is *Link Access Procedure, D channel*, also known as *LAPD*. LAPD is similar to *High-level Data Link Control (HDLC)* and *Link Access Procedure, Balanced (LAPB)* (see Chapter 12, “SDLC and Derivatives,” and Chapter 13, “X.25,” for more information on these protocols). As LAPD’s acronym expansion indicates, it is used across the D channel to ensure that control and signaling information flows and is received

properly. LAPD's frame format (see Figure 11-3) is very similar to that of HDLC and, like HDLC, LAPD uses supervisory, information, and unnumbered frames. The LAPD protocol is formally specified in CCITT Q.920 and CCITT Q.921.

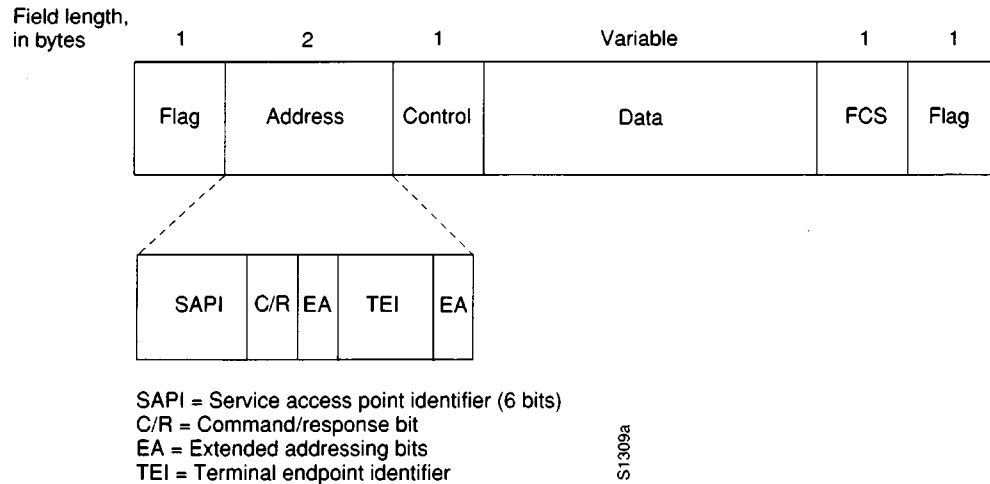


Figure 11-3 LAPD Frame Format

The LAPD *flag* and *control* fields are identical to those of HDLC. The LAPD *address* field can be either one or two bytes long. If the extended address bit of the first byte is set, the address is one byte; if it is not set, the address is two bytes. The first address field byte contains the *service access point identifier* (SAPI), which identifies the portal at which LAPD services are provided to Layer 3. The C/R bit indicates whether the frame contains a command or a response. The *terminal end-point identifier* (TEI) field identifies either a single terminal or multiple terminals. A TEI of all ones indicates a broadcast.

Layer 3

Two Layer 3 specifications are used for ISDN signaling: CCITT I.450 (also known as CCITT Q.930) and CCITT I.451 (also known as CCITT Q.931). Together, these protocols support user-to-user, circuit-switched, and packet-switched connections. A variety of call establishment, call termination, information, and miscellaneous messages are specified, including SETUP, CONNECT, RELEASE, USER INFORMATION, CANCEL, STATUS, and DISCONNECT. These messages are functionally similar to those provided by the X.25 protocol (see Chapter 13, "X.25," for more information). Figure 11-4, from CCITT I.451, shows the typical stages of an ISDN circuit-switched call.

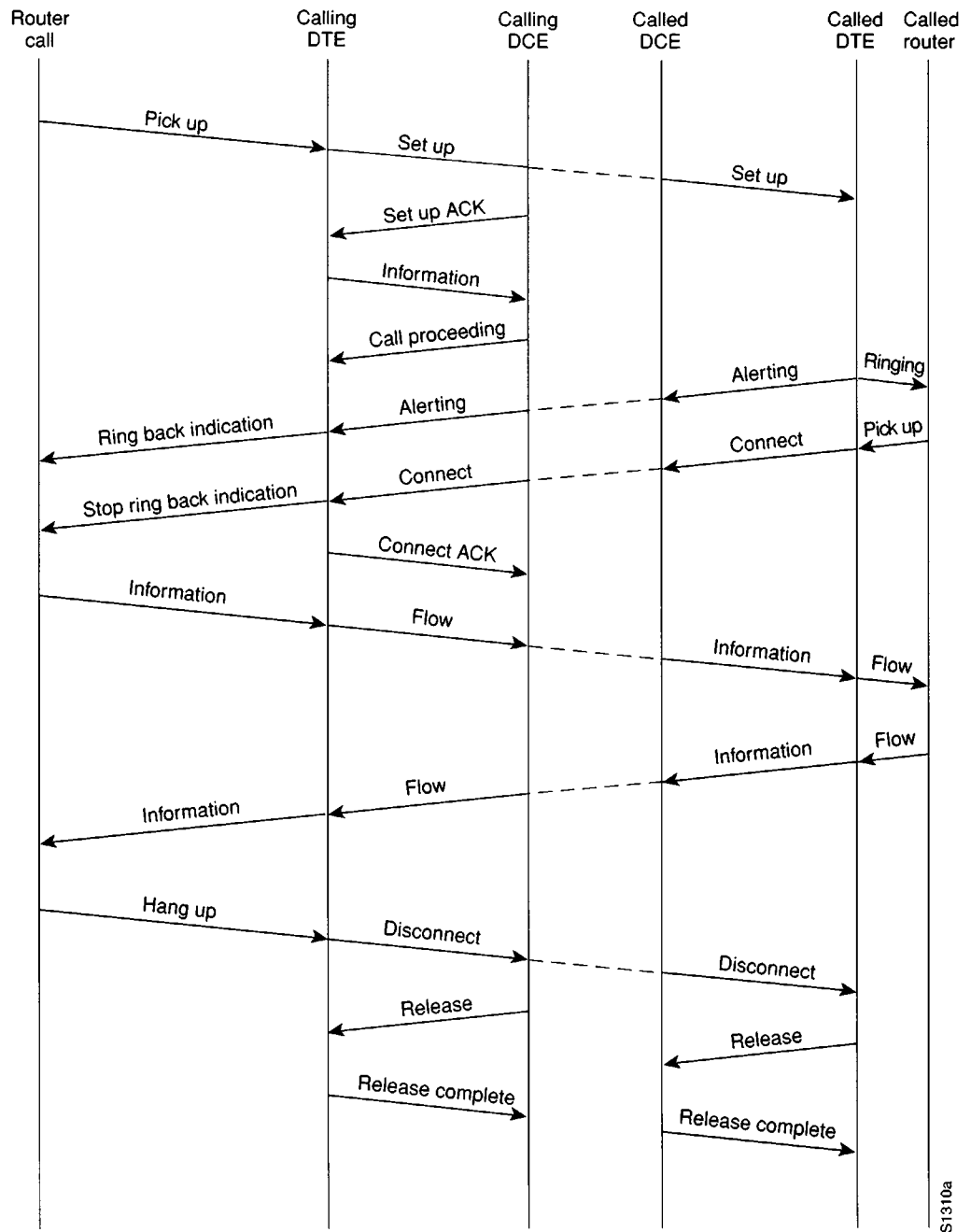
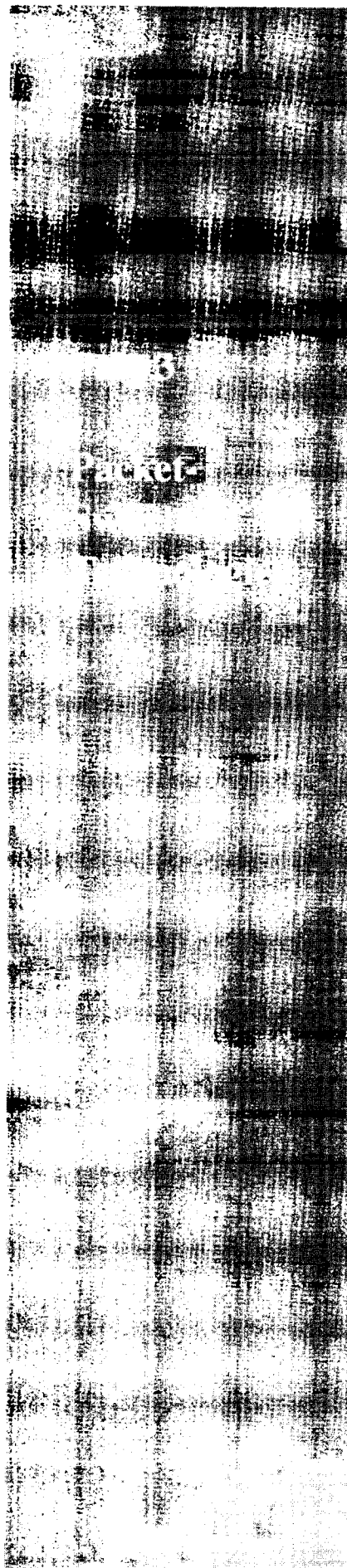


Figure 11-4 ISDN Circuit-Switched Call Stages



Chapter 12

SDLC and Derivatives

12

Background

IBM created the Synchronous Data-Link Control (SDLC) protocol in the mid-1970s for use in *Systems Network Architecture (SNA)* environments. SDLC was the first of an important new breed of data-link layer protocols based on synchronous, bit-oriented operation. Compared to synchronous character-oriented (for example, *Bisync* from IBM) and synchronous byte-count-oriented protocols (for example, *Digital Data Communications Message Protocol* from Digital Equipment Corporation), bit-oriented synchronous protocols are more efficient, more flexible, and often faster.

After developing SDLC, IBM submitted it to various standards committees. The International Organization for Standardization (ISO) modified SDLC to create the *High-level Data Link Control (HDLC)* protocol. The Consultative Committee for International Telegraph and Telephone (CCITT) subsequently modified HDLC to create *Link Access Procedure (LAP)*, and then *Link Access Procedure, Balanced (LAPB)*. The Institute of Electrical and Electronic Engineers (IEEE) modified HDLC to create *IEEE 802.2*. Each of these protocols has become important in its own domain. SDLC remains SNA's primary link-layer protocol for wide area network (WAN) links.

Technology Basics

SDLC supports a variety of link types and topologies. It can be used with point-to-point and multipoint links, bounded and unbounded media, half-duplex and full-duplex transmission facilities, and circuit-switched and packet-switched networks.

SDLC identifies two types of network nodes:

- *Primary*—Controls the operation of other stations (called secondaries). The primary polls the secondaries in a predetermined order. Secondaries can then transmit if they have outgoing data. The primary also sets up and tears down links and manages the link while it is operational.
- *Secondary*—Are controlled by a primary. Secondaries can only send information to the primary, but cannot do this unless the primary gives permission.

SDLC primaries and secondaries can be connected in four basic configurations:

- *Point-to-point*—Involves only two nodes: one primary and one secondary.
- *Multipoint*—Involves one primary and multiple secondaries.
- *Loop*—Involves a loop topology, with the primary connected to the first and last secondaries. Intermediate secondaries pass messages through one another as they respond to the primary's requests.
- *Hub go-ahead*—Involves an inbound and an outbound channel. The primary uses the outbound channel to communicate with the secondaries. The secondaries use the inbound channel to communicate with the primary. The inbound channel is daisy-chained back to the primary through each secondary.

Frame Format

The SDLC frame appears in Figure 12-1.

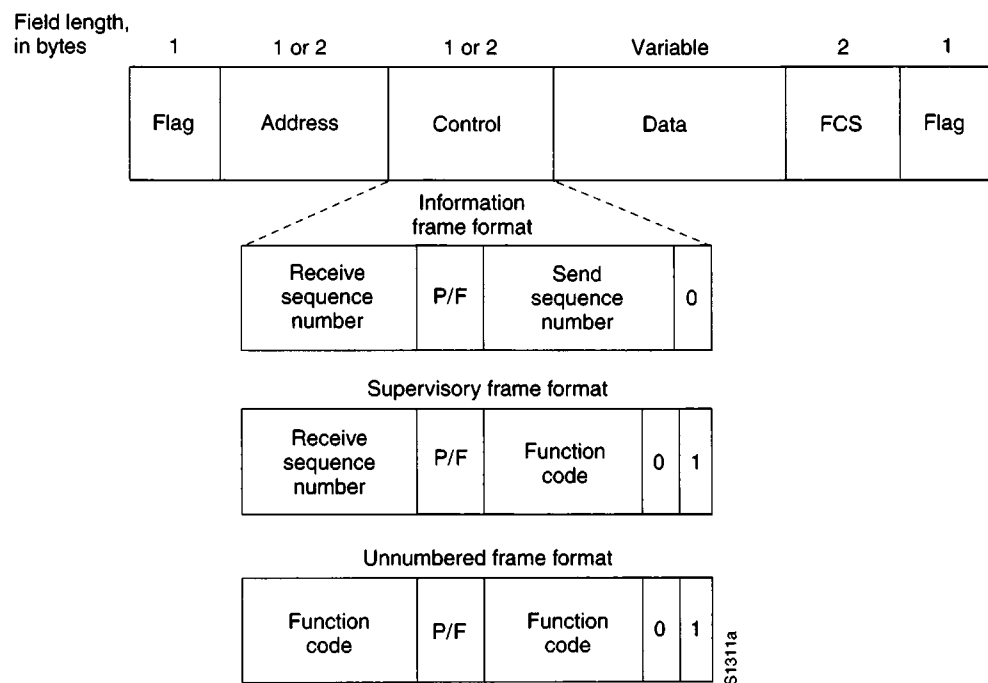


Figure 12-1 SDLC Frame Format

As the figure shows, SDLC frames are bounded by a unique *flag* pattern. The *address* field always contains the address of the secondary involved in the current communication. Since the primary is either the communication source or destination, there is no need to include the primary's address—it is known a priori by all secondaries.

The *control* field uses three different formats, depending on the type of SDLC frame used. The three SDLC frames are described in the following list:

- *Information (I)* frames—These frames carry upper-layer information and some control information (necessary for full-duplex operation). Send and receive sequence numbers and the poll final (P/F) bit perform flow and error control. The *send sequence number* refers to the number of the frame to be sent next. The *receive sequence number* provides the number of the frame to be received next. In full duplex conversation, both the sender and the receiver keep send and receive sequence numbers. The primary uses the P/F bit to tell the secondary whether it requires an immediate response. The secondary uses this bit to tell the primary whether the current frame is the last in its current response.
- *Supervisory (S)* frames—These frames provide control information. They do not have an information field. Supervisory frames request and suspend transmission, report on status, and acknowledge the receipt of I frames.
- *Unnumbered (U)* frames—These frames, as the name suggests, are not sequenced. They may have an information field. U frames are used for control purposes. For example, they may specify either a one- or a two-byte control field, initialize secondaries, and do other, similar, functions.

The *frame check sequence (FCS)* precedes the ending flag delimiter. The FCS is usually a *cyclic redundancy check (CRC)* calculation remainder. The CRC calculation is redone in the receiver. If the result differs from the value in the sender's frame, an error is assumed.

A typical SDLC-based network configuration appears in Figure 12-2. As illustrated, an IBM establishment controller (formerly called a cluster controller) in a remote site connects to dumb terminals and to a Token Ring network. In a local site, an IBM host connects (via channel attach techniques) to an IBM front-end processor (FEP), which can also have links to local Token Ring local area networks (LANs) and an SNA backbone. The two sites are connected through a SDLC-based 56-Kbps leased line.

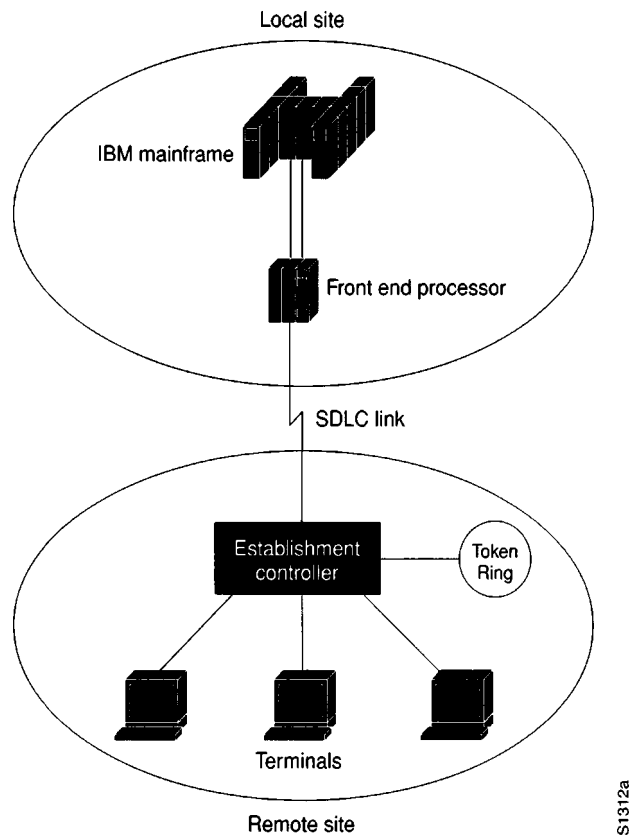


Figure 12-2 Typical SDLC-Based Network Configuration

Derivative Protocols

Despite the fact that it omits several features used in SDLC, HDLC is generally considered to be a compatible superset of SDLC. LAP is a subset of HDLC. LAPB was created to ensure ongoing compatibility with HDLC, which had been modified in the early 1980s. IEEE 802.2 is a modification of HDLC for LAN environments.

HDLC

HDLC shares SDLC's frame format and HDLC fields provide the same functionality as those in SDLC. Also, like SDLC, HDLC supports synchronous, full-duplex operation.

HDLC differs from SDLC in several minor ways. First, HDLC has an option for a 32-bit checksum. And, unlike SDLC, HDLC does not support the loop or hub go-ahead configurations.

The major difference between HDLC and SDLC is that SDLC supports only one transfer mode, while HDLC supports three. The three HDLC transfer modes are as follows:

- *Normal response mode (NRM)*—This transfer mode is used by SDLC. In this mode, secondaries cannot communicate with a primary until the primary has given permission.
- *Asynchronous response mode (ARM)*—This transfer mode allows secondaries to initiate communication with a primary without receiving permission.
- *Asynchronous balanced mode (ABM)*—ABM introduces the *combined* node. A combined node can act as a primary or a secondary, depending on the situation. All ABM communication is between multiple combined nodes. In ABM environments, any combined station may initiate data transmission without permission from any other.

LAPB

LAPB is best known for its presence in the X.25 protocol stack. LAPB shares the same frame format, frame types, and field functions as SDLC and HDLC. Unlike either of these, however, LAPB is restricted to the ABM transfer mode, and so is appropriate only for combined stations. Also, LAPB circuits can be established by either the data terminal equipment (DTE) or the data circuit-terminating equipment (DCE). The station initiating the call is determined to be the primary, while the responding station is the secondary. Finally, LAPB use of the P/F bit is somewhat different than that of the other protocols. See Chapter 13, “X.25,” for details on LAPB.

IEEE 802.2

IEEE 802.2 is often referred to as *Logical Link Control (LLC)*. It is extremely popular in LAN environments, where it interoperates with protocols such as *IEEE 802.3*, *IEEE 802.4*, and *IEEE 802.5*.

IEEE 802.2 offers three types of service. *Type 1* provides unacknowledged connectionless service. *Type 2* provides connection-oriented service. *Type 3* provides acknowledged connectionless service.

As an unacknowledged connectionless service, LLC Type 1 does not confirm data transfers. Because many upper-layer protocols such as *Transmission Control Protocol/Internet Protocol (TCP/IP)* offer reliable data transfer that can compensate for unreliable lower-layer protocols, Type 1 is a commonly used service.

LLC Type 2 (often called *LLC2*) service establishes virtual circuits between sender and receiver, and is therefore connection-oriented. LLC2 acknowledges data upon receipt and is found in IBM communication systems.

Although it supports acknowledged data transfer, LLC Type 3 service does not establish virtual circuits. As a compromise between the other two LLC services, LLC Type 3 is useful in factory automation environments where error detection is important, but context storage space (for virtual circuits) is extremely limited.

End stations can support multiple LLC service types. A Class I device supports only Type 1 service. A Class II device supports both Type 1 and Type 2 service. Class III devices support both Type 1 and Type 3 service, while Class IV devices support all three types of service.

Upper-layer processes use IEEE 802.2 services through *service access points (SAPs)*. The IEEE 802.2 header begins with a *destination service access point (DSAP)* field, which identifies the receiving upper-layer process. In other words, after the receiving node's IEEE 802.2 implementation completes its processing, the upper-layer process identified in the DSAP field receives the remaining data. Following the DSAP address is the *source service access point (SSAP)* address, which identifies the sending upper-layer process.

Chapter 13

X.25

13

Background

In the mid-to-late 1970s, a set of protocols was needed to provide users with wide area network (WAN) connectivity across *public data networks (PDNs)*. PDNs such as TELENET and TYMNET had achieved remarkable success, but it was felt that protocol standardization would further subscription to PDNs by providing increased equipment compatibility and lower cost. The result of the ensuing development effort was a group of protocols, the most popular of which is X.25.

X.25 (formally referred to as *CCITT Recommendation X.25*) was developed by the common carriers (phone companies, essentially) rather than any single commercial enterprise. The specification is therefore designed to work well regardless of a user's system type or manufacturer. Users contract with the common carriers to use their *packet-switched networks (PSNs)* and are charged based on PSN use. Services offered (and charges levied) are regulated by the Federal Communications Commission (FCC).

One of X.25's unique attributes is its international nature. X.25 and related protocols are administered by an agency of the United Nations called the International Telecommunications Union (ITU). The ITU committee responsible for voice and data communications is the Consultative Committee for International Telegraph and Telephone (CCITT). CCITT members include the FCC, the European PTTs, the common carriers, and many computer and data communications companies. As a direct result of its heritage, X.25 is truly a global standard.

Technology Basics

X.25 defines a telephone network for data communications. To begin communication, one computer calls another to request a communication session. The called computer can accept or refuse the connection. If the call is accepted, the two systems can begin full-duplex information transfer. Either side can terminate the connection at any time.

The X.25 specification defines a point-to-point interaction between *data terminal equipment (DTE)* and *data circuit-terminating equipment (DCE)*. DTEs (terminals and hosts in the user's facilities) connect to DCEs (modems, packet switches and other ports into the PDN, generally located in the carrier's facilities), which connect to *packet switching exchanges (PSEs)*, or simply *switches* and other DCEs inside a PSN and, ultimately, to another DTE. The relationship between the entities in an X.25 network is shown in Figure 13-1.

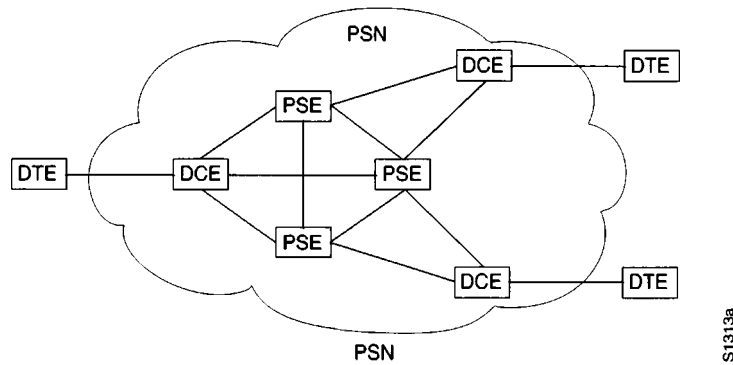


Figure 13-1 X.25 Model

A DTE can be a terminal that does not implement the complete X.25 functionality. These DTE are connected to the DCE through a translation device called a *packet assembler/disassembler (PAD)*. The operation of the terminal-to-PAD interface, the services offered by the PAD, and the interaction between the PAD and the host are defined by CCITT *Recommendations X.28, X.3, and X.29*, respectively.

The X.25 specification maps to Layers 1 through 3 of the OSI reference model. Layer 3 X.25 describes packet formats and packet exchange procedures between peer Layer 3 entities. Layer 2 X.25 is implemented by *Link Access Procedure, Balanced (LAPB)*. LAPB defines packet framing for the DTE/DCE link. Layer 1 X.25 defines the electrical and mechanical procedures for activating and deactivating the physical medium connecting the DTE and the DCE. This relationship is shown in Figure 13-2. Note that Layers 2 and 3 are also referred to as the ISO standards ISO 7776 (LAPB) and ISO 8208 (X.25 packet layer).

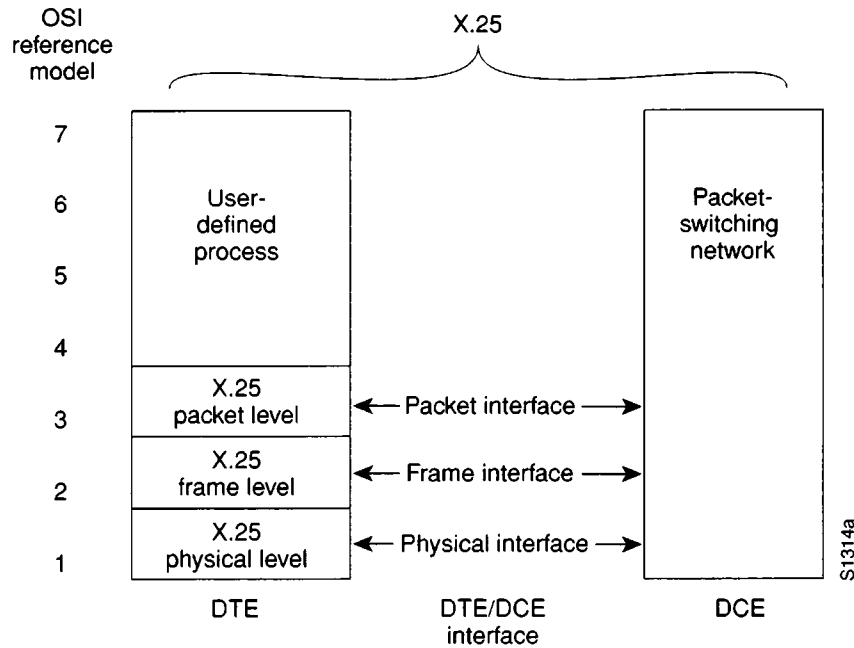


Figure 13-2 X.25 and the OSI Reference Model

End-to-end communication between DTEs is accomplished through a bidirectional association called a *virtual circuit*. Virtual circuits permit communication between distinct network elements through any number of intermediate nodes without the dedication of portions of the physical medium that characterizes physical circuits. Virtual circuits can either be permanent or switched (temporary). *Permanent virtual circuits* are commonly called *PVCs*; *switched virtual circuits* are commonly called *SVCs*. PVCs are typically used for the most-often-used data transfers, while SVCs are used for sporadic data transfers. Layer 3 X.25 is concerned with end-to-end communication involving both PVCs and SVCs.

Once a virtual circuit is established, the DTE sends a packet to the other end of the connection by sending it to the DCE using the proper virtual circuit. The DCE looks at the virtual circuit number to determine how to route the packet through the X.25 network. The Layer 3 X.25 protocol multiplexes between all the DTE served by the DCE on the destination side of the network and the packet is delivered to the destination DTE.

Frame Format

An X.25 frame is composed of a series of fields, as shown in Figure 13-3. Layer 3 X.25 fields make up an X.25 packet and include a header and user data. Layer 2 X.25 (LAPB) fields include frame-level control and addressing fields, the embedded Layer 3 packet, and a frame check sequence (FCS).

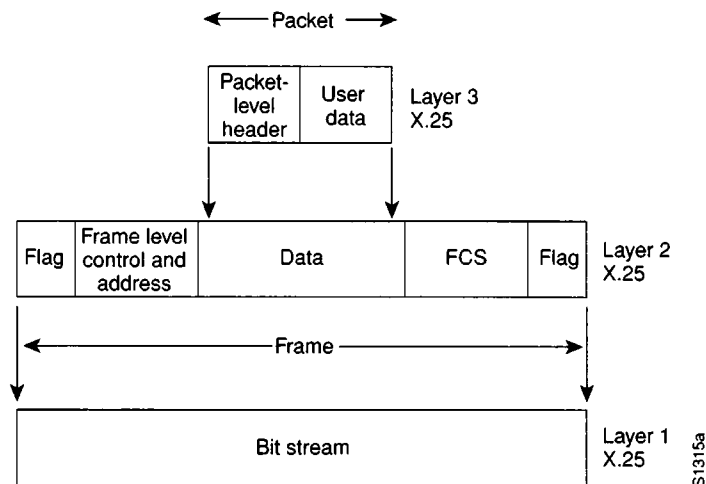


Figure 13-3 X.25 Frame

Layer 3

The Layer 3 X.25 header is made up of a *general format identifier (GFI)*, a *logical channel identifier (LCI)*, and a *packet type identifier (PTI)*. The GFI is a 4-bit field that indicates the general format of the packet header. The LCI is a 12-bit field that identifies the virtual circuit. The LCI is locally significant at the DTE/DCE interface. In other words, the PDN connects two logical channels, each with an independent LCI, on two DTE/DCE interfaces to establish a virtual circuit. The PTI field identifies one of X.25's 17 packet types.

Addressing fields in call setup packets provide source and destination DTE addresses. These are used to establish the virtual circuits that comprise X.25 communication. *CCITT Recommendation X.121* specifies the source and destination address formats. X.121 addresses (also referred to as *International Data Numbers*, or *IDNs*) vary in length and can be up to 14 decimal digits long. Byte four in the call setup packet specifies the source DTE and destination DTE address lengths. The first four digits of an IDN are called the *data network identification code (DNIC)*. The DNIC is divided into two parts, the first (three digits) specifying the country and the last specifying the PSN itself. The remaining digits are called the *national terminal number (NTN)*, and are used to identify the specific DTE on the PSN. The X.121 address format is shown in Figure 13-4.

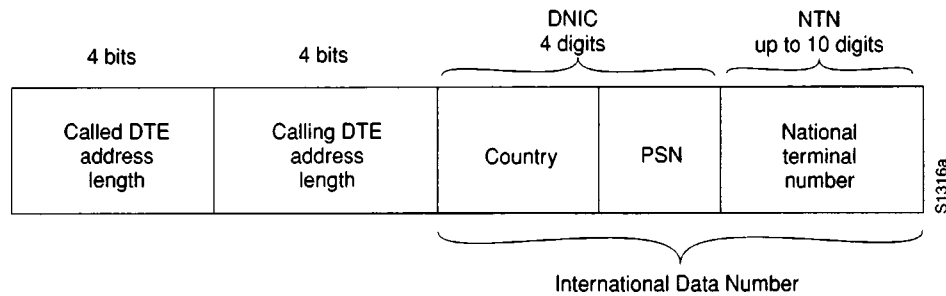


Figure 13-4 X.121 Address Format

The addressing fields that make up the X.121 address are only necessary when an SVC is used, and then only during call setup. Once the call is established, the PSN uses the LCI field of the data packet header to specify the particular virtual circuit to the remote DTE.

Layer 3 X.25 uses three virtual circuit operational procedures:

- Call setup
- Data transfer
- Call clearing

Execution of these procedures depends upon the virtual circuit type being used. For a PVC, Layer 3 X.25 is always in data transfer mode because the circuit has been permanently established. If an SVC is used, all three procedures are used.

Data transfer is effected by DATA packets. Layer 3 X.25 segments and reassembles user messages if they are too long for the circuit's maximum packet size. Each DATA packet is given a sequence number so error and flow control can occur across the DTE/DCE interface.

Layer 2

Layer 2 X.25 is implemented by LAPB. LAPB allows both sides (the DTE and the DCE) to initiate communication with the other. During information transfer, LAPB checks that the frames arrive at the receiver in the correct sequence and error-free.

As with similar link-layer protocols, LAPB uses three frame format types:

- *Information (I) frame*—These frames carry upper-layer information and some control information (necessary for full-duplex operation). Send and receive sequence numbers and the poll final (P/F) bit perform flow control and error recovery. The *send sequence number* refers to the number of the current frame. The *receive sequence number* records the number of the frame to be received next. In full duplex conversation, both the sender and the receiver keep send and receive sequence numbers. The poll bit is used to force a final bit message in response; this is used for error detection and recovery.

- *Supervisory (S) frames*—These frames provide control information. They do not have an information field. S frames request and suspend transmission, report on status, and acknowledge the receipt of I frames.
- *Unnumbered (U) frames*—These frames, as the name suggests, are not sequenced. U frames are used for control purposes. For example, they may initiate a connection using standard or extended windowing (modulo 8 versus 128), disconnect the link, report a protocol error, or similar functions.

The LAPB frame is shown in Figure 13-5.

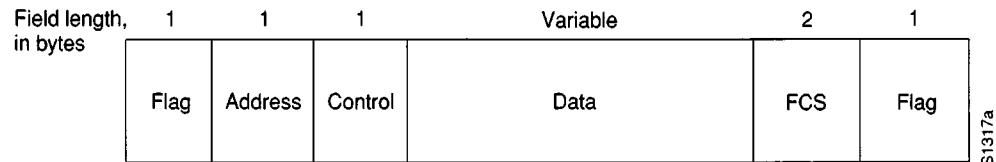


Figure 13-5 LAPB Frame

The *flag* fields delimit the LAPB frame. Bit stuffing is used to ensure that the flag pattern does not occur within the body of the frame.

The *address* field indicates whether the frame carries a command or a response.

The *control* field provides further qualifications of command and response frames, and also indicates the frame format (U, I, or S), frame function (for example, *receiver ready* or *disconnect*), and the send/receive sequence number.

The *data* field carries upper-layer data. Its size and format vary depending upon the Layer 3 packet type. The maximum length of this field is set by agreement between a PSN administrator and the subscriber at subscription time.

The *FCS* field ensures the integrity of the transmitted data.

Layer 1

Layer 1 X.25 uses the X.21 bis physical-layer protocol, which is roughly equivalent to RS-232-C. X.21 bis was derived from CCITT Recommendations V.24 and V.28, which identify the interchange circuits and electrical characteristics (respectively) of a DTE to DCE interface. X.21 bis supports point-to-point connections, speeds up to 19.2 Kbps, and synchronous, full-duplex transmission over four-wire media. The maximum distance between DTE and DCE is 15 meters.

Chapter 14

Frame Relay

14

Background

Frame Relay was originally conceived as a protocol for use over ISDN interfaces, and initial proposals to this effect were submitted to the Consultative Committee for International Telegraph and Telephone (CCITT) in 1984. Work on Frame Relay was also undertaken in the American National Standards Institute (ANSI)-accredited T1S1 standards committee in the United States.

A major development in Frame Relay's history occurred in 1990 when Cisco Systems, StrataCom, Northern Telecom, and Digital Equipment Corporation formed a consortium to focus Frame Relay technology development and accelerate the introduction of interoperable Frame Relay products. This consortium developed a specification conforming to the basic Frame Relay protocol being discussed in T1S1 and CCITT, but extended it with features that provide additional capabilities for complex internetworking environments. These Frame Relay extensions are referred to collectively as the *local management interface (LMI)*.

Technology Basics

Frame Relay provides a packet-switching data communications capability that is used across the interface between user devices (for example, routers, bridges, host machines) and network equipment (for example, switching nodes). User devices are often referred to as *data terminal equipment (DTE)*, while network equipment that interfaces to DTE is often referred to as *data circuit-terminating equipment (DCE)*. The network providing the Frame Relay interface can be either a carrier-provided public network or a network of privately owned equipment serving a single enterprise.

As an interface to a network, Frame Relay is the same type of protocol as X.25 (see Chapter 13, "X.25"). However, Frame Relay differs significantly from X.25 in its functionality and format. In particular, Frame Relay is a more streamlined protocol, facilitating higher performance and greater efficiency.

As an interface between user and network equipment, Frame Relay provides a means for statistically multiplexing many logical data conversations (referred to as *virtual circuits*) over a single physical transmission link. This contrasts with systems that use only *time-division-multiplexing (TDM)* techniques for supporting multiple data streams. Frame Relay's statistical multiplexing provides more flexible and efficient use of available bandwidth. It can be used without TDM techniques or on top of channels provided by TDM systems.

Another important characteristic of Frame Relay is that it exploits the recent advances in wide area network (WAN) transmission technology. Earlier WAN protocols such as X.25 were developed when analog transmission systems and copper media were predominant. These links are much less reliable than the fiber media/digital transmission links available today. Over links such as these, data-link level protocols can forego time-consuming error correction algorithms, leaving these to be performed at higher protocol layers. Greater performance and efficiency is therefore possible without sacrificing data integrity. Frame Relay is designed with this approach in mind. It includes a *cyclic redundancy check (CRC)* algorithm for detecting corrupted bits (so the data can be discarded), but it does not include any protocol mechanisms for correcting bad data (for example, by retransmitting it at this level of protocol).

Another difference between Frame Relay and X.25 is the absence of explicit, per-virtual-circuit flow control in Frame Relay. Now that many upper-layer protocols are effectively executing their own flow control algorithms, the need for this functionality at the link layer has diminished. Frame Relay, therefore, does not include explicit flow control procedures that are redundant with those in higher layers. Instead, very simple congestion notification mechanisms are provided to allow a network to inform a user device that the network resources are close to a congested state. This notification can alert higher-layer protocols that flow control may be needed.

Current Frame Relay standards address *permanent virtual circuits (PVCs)* that are administratively configured and managed in a Frame Relay network. Another type, *switched virtual circuits (SVCs)*, has also been proposed. The *Integrated Services Digital Network (ISDN)* signalling protocol is proposed as the means by which DTE and DCE will communicate to dynamically establish, terminate, and manage SVCs. See Chapter 11, “ISDN,” for more information on ISDN. Both T1S1 and CCITT have work in progress to include SVCs in Frame Relay standards.

LMI Extensions

In addition to the basic Frame Relay protocol functions for transferring data, the consortium Frame Relay specification includes LMI extensions that make supporting large, complex internetworks easier. Some LMI extensions are referred to as “common” and are expected to be implemented by all adopters of the specification. Other LMI functions are referred to as “optional.” A summary of the LMI extensions follows:

- *Virtual circuit status messages (common)*—Provide communication and synchronization between the network and the user device, periodically reporting the existence of new PVCs and the deletion of already-existing PVCs, and generally providing information about PVC integrity. Virtual circuit status messages prevent the sending of data into *black holes*, that is, over PVCs that no longer exist.
- *Multicasting (optional)*—Allows a sender to transmit a single frame but have it delivered by the network to multiple recipients. Thus, multicasting supports the efficient conveyance of routing protocol messages and address resolution procedures that typically must be sent to many destinations simultaneously.

- *Global addressing* (optional)—Gives connection identifiers global rather than local significance, allowing them to be used to identify a specific interface to the Frame Relay network. Global addressing makes the Frame Relay network resemble a LAN in terms of addressing; address resolution protocols therefore perform over Frame Relay exactly as they do over a LAN.
- *Simple flow control* (optional)—Provides for an XON/XOFF flow control mechanism that applies to the entire Frame Relay interface. It is intended for those devices whose higher layers cannot use the congestion notification bits and that need some level of flow control.

Frame Format

The Frame Relay frame is shown in Figure 14-1. *Flags* delimit the frame's beginning and end. Following the leading flags are two bytes of *address* information. Ten bits of these two bytes comprise the actual circuit ID (called the *DLCI*, for *data link connection identifier*).

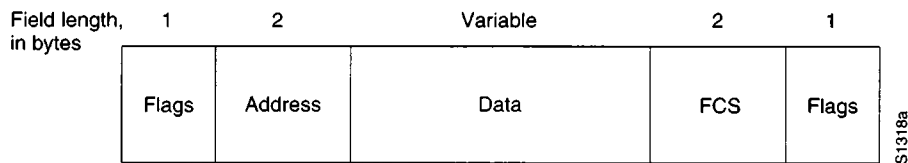


Figure 14-1 Frame Relay Frame

The 10-bit DLCI value is the heart of the Frame Relay header. It identifies the logical connection that is multiplexed into the physical channel. In the basic (that is, not extended by the LMI) mode of addressing, DLCIs have local significance; that is, the end devices at two different ends of a connection may use a different DLCI to refer to that same connection. Figure 14-2 provides an example of the use of DLCIs in nonextended Frame Relay addressing.

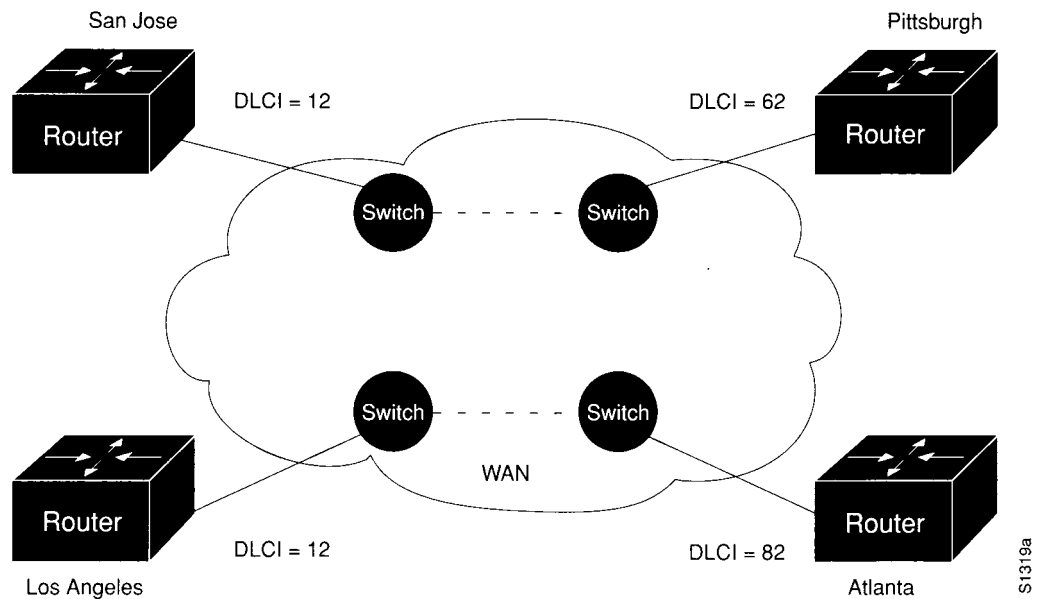


Figure 14-2 Frame Relay Addressing

In Figure 14-2, assume two PVCs, one between Atlanta and Los Angeles, and one between San Jose and Pittsburgh. Los Angeles may refer to its PVC with Atlanta using DLCI = 12, while Atlanta refers to the same PVC with DLCI = 82. Similarly, San Jose may refer to its PVC with Pittsburgh using DLCI = 62. The network uses internal proprietary mechanisms to keep the two locally significant PVC identifiers distinct.

At the end of each DLCI byte is an extended address (EA) bit. If this bit is one, the current byte is the last DLCI byte. All implementations currently use a two-byte DLCI, but the presence of the EA bits means that longer DLCIs may be agreed upon and used in the future.

The bit marked “C/R” following the most significant DLCI byte is currently not used.

Finally, three bits in the 2-byte DLCI are fields related to congestion control. The Forward Explicit Congestion Notification (FECN) bit is set by the Frame Relay network in a frame to tell the DTE receiving that frame that congestion was experienced in the path from source to destination. The Backward Explicit Congestion Notification (BECN) bit is set by the Frame Relay network in frames travelling in the opposite direction from frames encountering a congested path. The notion behind both of these bits is that the FECN or BECN indication can be promoted to a higher-level protocol that can take flow control action as appropriate. (FECN bits are useful to higher-layer protocols that use receiver-controlled flow control, while BECN bits are significant to those that depend on “emitter-controlled” flow control.)

The discard eligibility (DE) bit is set by the DTE to tell the Frame Relay network that a frame has lower importance than other frames and should be discarded before other frames if the network becomes short of resources. Thus, it represents a very simple priority mechanism. This bit is usually set only when the network is congested.

SI1319a

LMI Message Format

The previous section described the basic Frame Relay protocol format for carrying user data frames. The consortium Frame Relay specification also includes the LMI procedures. LMI messages are sent in frames distinguished by an LMI-specific DLCI (defined in the consortium specification as DLCI=1023). The LMI message format is shown in Figure 14-3.

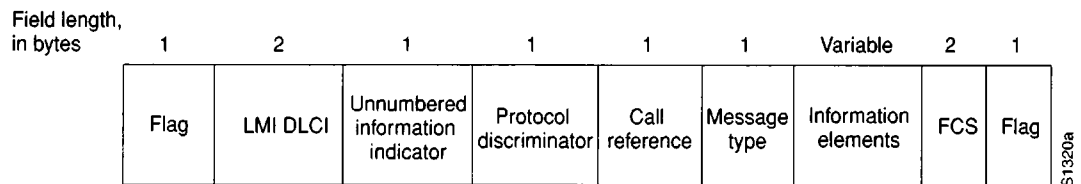


Figure 14-3 LMI Message Format

In LMI messages, the basic protocol header is the same as in normal data frames. The actual LMI message begins with four mandatory bytes, followed by a variable number of *information elements (IEs)*. The format and encoding of LMI messages is based on the ANSI T1S1 standard.

The first of the mandatory bytes (*unnumbered information indicator*) has the same format as the LAPB *unnumbered information (UI)* frame indicator with the poll/final bit set to zero. See the section “Layer 2” in Chapter 13, “X.25” for more information on LAPB. The next byte is referred to as the *protocol discriminator*, which is set to a value that indicates “LMI.” The third mandatory byte (*call reference*) is always filled with zeros.

The final mandatory byte is the *message type* field. Two message types have been defined. *Status-enquiry* messages allow the user device to inquire about network status. *Status* messages respond to status-enquiry messages. *Keepalives* (messages sent through a connection to ensure that both sides will continue to regard the connection as active) and PVC status messages are examples of these messages and are the common LMI features that are expected to be a part of every implementation that conforms to the consortium specification.

Together, status and status-enquiry messages help verify the integrity of logical and physical links. This information is critical in a routing environment, because routing algorithms make decisions based on link integrity.

Following the message type field is some number of IEs. Each IE consists of a single-byte *IE identifier*, an *IE length* field, and one or more bytes containing actual data.

Global Addressing

In addition to the common LMI features, there are several optional LMI extensions that are extremely useful in an internetworking environment. The first important optional LMI extension is *global addressing*. As noted earlier, the basic (nonextended) Frame Relay specification only supports values of the DLCI field that identify PVCs with local significance. In this case, there are no addresses that identify network interfaces, or nodes

attached to these interfaces. As these addresses do not exist, they cannot be discovered by traditional address resolution and discovery techniques. This means that with normal Frame Relay addressing, static maps must be created to tell routers which DLCIs to use to find a remote device and its associated internetwork address.

The global addressing extension permits node identifiers. With this extension, the values inserted in the DLCI field of a frame are globally significant addresses of individual end user devices (for example, routers). This is implemented as shown in Figure 14-4.

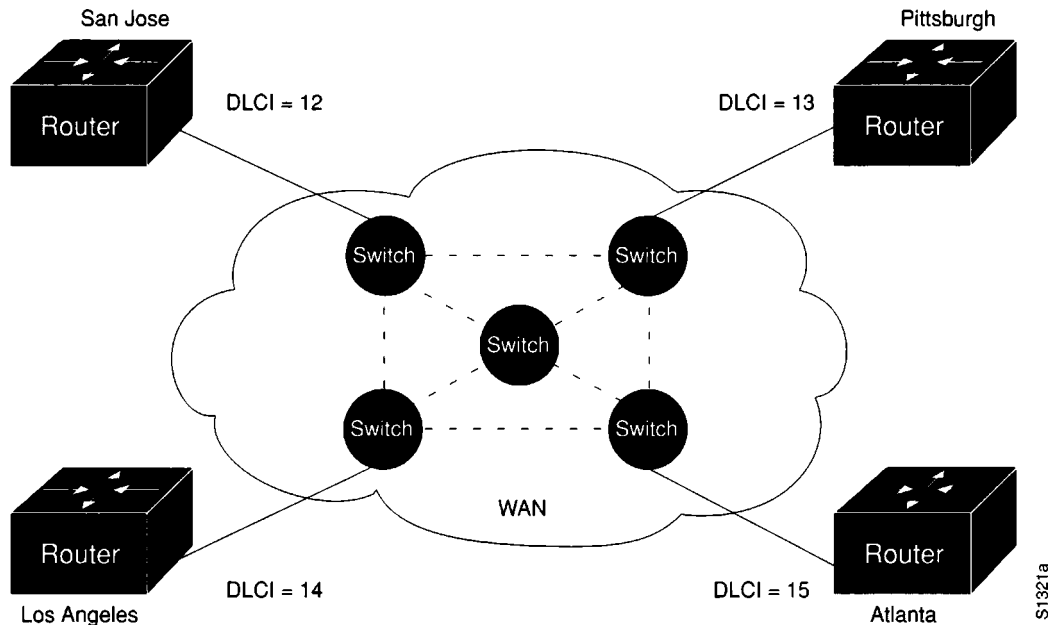


Figure 14-4 Global Addressing Exchange

In Figure 14-4, note that each interface has its own identifier. Suppose Pittsburgh must send a frame to San Jose. San Jose's identifier is 12, so Pittsburgh places the value 12 in the DLCI field and sends the frame into the Frame Relay network. At the exit point, the DLCI field contents are changed by the network to 13 to reflect the frame's source node. As each router's interface has a distinct value as its node identifier, individual devices can be distinguished. This permits adaptive routing in complex environments.

Global addressing provides significant benefits in a large complex internetwork. The Frame Relay network now appears to the routers on its periphery like any LAN. No changes to higher-layer protocols are needed to take full advantage of their capabilities.

Multicasting

Multicasting is another valuable optional LMI feature. Multicast groups are designated by a series of four reserved DLCI values (1019 through 1022). Frames sent by a device using one of these reserved DLCIs are replicated by the network and sent to all exit points in the designated set. The multicasting extension also defines LMI messages that notify user devices of the addition, deletion, and presence of multicast groups.

In networks that take advantage of dynamic routing, routing information must be exchanged among many routers. Routing messages can be sent efficiently by using frames with a multicast DLCI. This allows messages to be sent to specific groups of routers.

Network Implementation

Frame Relay can be used as an interface to either a publicly available carrier-provided service or to a network of privately owned equipment. A typical means of private network implementation is to equip traditional T1 multiplexors with Frame Relay interfaces for data devices, as well as non-Frame Relay interfaces for other applications such as voice and video-conferencing. Figure 14-5 shows this configuration.

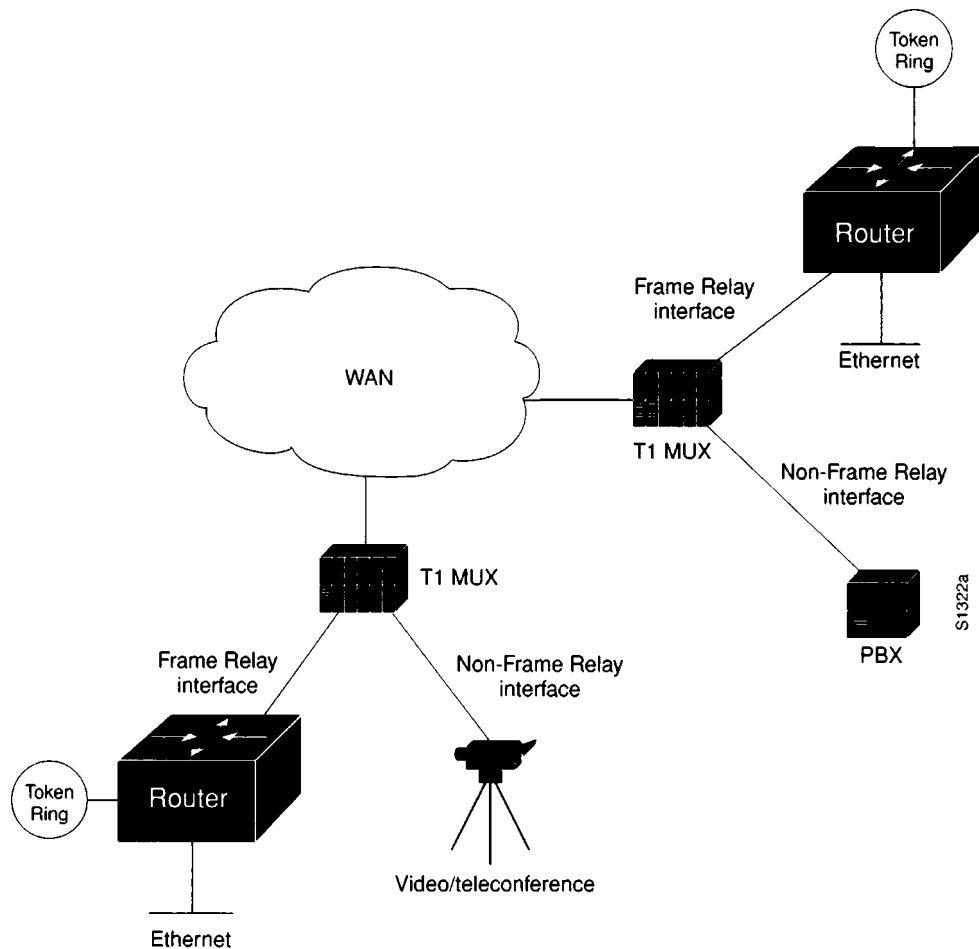


Figure 14-5 Hybrid Frame Relay Network

A public Frame Relay service is deployed by putting Frame Relay switching equipment in the central offices (COs) of a telecommunications carrier. In this case, users may realize economic benefits from traffic-sensitive charging rates, and are relieved from the work necessary to administer and maintain the network equipment and service.

In either type of network, the lines that connect user devices to the network equipment may operate at a speed selected from a broad range of data rates. Speeds between 56 Kbps and 2 Mbps are typical, although both lower and higher speeds are supportable by the Frame Relay technology. Implementations capable of operating over 45 Mbps (DS3) links are expected to be available soon.

Whether in a public or private network, the support of Frame Relay interfaces to user devices does not necessarily dictate that the Frame Relay protocol is used between the network devices. No standards for interconnecting equipment inside a Frame Relay network currently exist. Thus, traditional circuit-switching, packet-switching, or a hybrid approach combining these technologies can be used.

Chapter 15

SMDS

15

Background

Switched Multimegabit Data Service (SMDS) is a packet-switched datagram service designed for very high-speed wide-area data communications. Offering data throughputs that will initially be in the 1-Mbps to 34-Mbps range, SMDS is being deployed in public networks by the carriers in response to two trends. The first of these is the proliferation of distributed processing and other applications that require high performance networking. The second trend is the decreasing cost and high bandwidth potential of fiber media, making support of such applications over a wide area network (WAN) viable.

SMDS is described in a series of specifications produced by Bell Communications Research (Bellcore) and adopted by the telecommunications equipment providers and carriers. One of these specifications describes the *SMDS Interface Protocol (SIP)*, which is the protocol between a user device (referred to as *customer premises equipment*, or *CPE*), and SMDS network equipment.

The SIP is based on an IEEE standard protocol for metropolitan area networks (MANs); that is, the *IEEE 802.6 Distributed Queue Dual Bus (DQDB)* standard. Using this protocol, CPE such as routers can attach to an SMDS network and use SMDS service for high-speed internetworking.

Technology Basics

Figure 15-1 depicts an internetworking scenario using SMDS. As shown in this figure, access to SMDS is provided over either a 1.544-Mbps (*DS-1*, or *Digital Signal 1*) or 44.736-Mbps (*DS-3*, or *Digital Signal 3*) transmission facility. Although SMDS is usually described as a fiber-based service, DS-1 access may be provided over either fiber or over copper-based media with sufficiently good error characteristics. The demarcation point between the carrier's SMDS network and the customer's equipment is referred to as the *subscriber network interface (SNI)*.

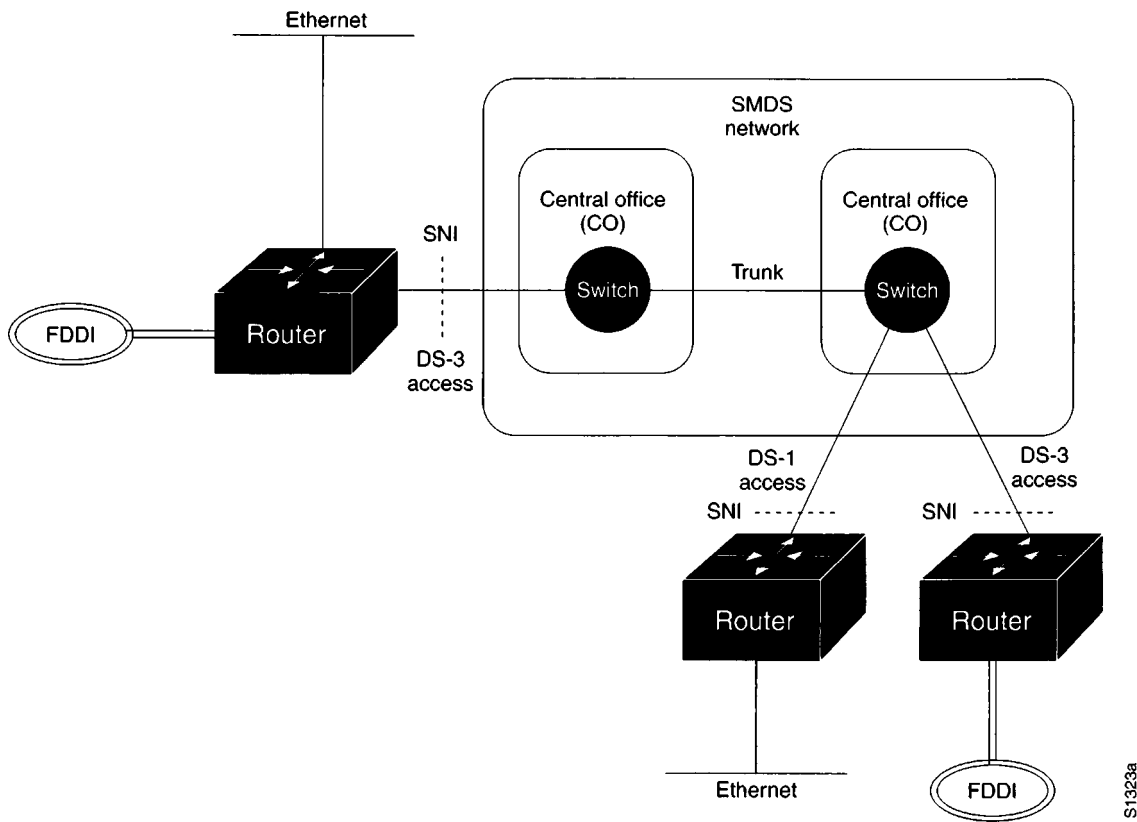


Figure 15-1 SMDS Internetworking Scenario

SMDS data units are capable of containing up to 9,188 octets (bytes) of user information. SMDS is therefore capable of encapsulating entire *IEEE 802.3*, *IEEE 802.4*, *IEEE 802.5*, and *FDDI* packets. The large packet size is consistent with the high-performance objectives of the service.

Addressing

Like other datagram protocols, SMDS data units carry both a source and a destination address. The receiver of a data unit can use the source address to return data to the sender and for functions such as address resolution (discovering the mapping between higher-layer addresses and SMDS addresses). SMDS addresses are 10-digit addresses that resemble conventional phone numbers.

In addition, SMDS supports group addresses that allow a single data unit to be sent and then delivered by the network to multiple recipients. Group addressing is analogous to multicasting on local area networks (LANs) and is a valuable feature in internetworking applications where it is widely used for routing, address resolution, and dynamic discovery of network resources (such as file servers).

SMDS offers several other addressing features. Source addresses are validated by the network to ensure that the address in question is legitimately assigned to the SNI from which it originated. Thus, users are protected against *address spoofing*; that is, a sender pretending to be another user. Source and destination address screening is also possible. Source address screening acts on addresses as data units are leaving the network, while destination address screening acts on addresses as data units are entering the network. If the address is disallowed, the data unit is not delivered. With address screening, a subscriber can establish a private virtual network that excludes unwanted traffic. This provides the subscriber with an initial security screen and promotes efficiency because devices attached to SMDS do not have to waste resources handling unwanted traffic.

Access Classes

To accommodate a range of traffic requirements and equipment capabilities, SMDS supports a variety of access classes. Different access classes determine the various maximum sustained information transfer rates as well as the degree of burstiness allowed when sending packets into the SMDS network.

On DS-3-rate interfaces, access classes are implemented through credit management algorithms. Credit management algorithms track credit balances for each customer interface. Credit is allocated on a periodic basis, up to some maximum. Then, the credit balance is decremented as packets are sent to the network.

The operation of the credit management scheme essentially constrains the customer's equipment to some sustained, or average rate of data transfer. This average rate of transfer is less than the full information carrying bandwidth of the DS-3 access facility. Five access classes, corresponding to sustained information rates of 4, 10, 16, 25, and 34 Mbps, are supported for DS-3 access interface. The credit management scheme is not applied to DS-1-rate access interfaces.

SMDS Interface Protocol (SIP)

Access to the SMDS network is accomplished via SIP. The SIP is based on the DQDB protocol specified by the IEEE 802.6 MAN standard. The DQDB protocol defines a media-access-control scheme allowing many systems to interconnect via two unidirectional logical buses.

As designed by IEEE 802.6, the DQDB standard can be used to construct private, fiber-based MANs supporting a variety of applications including data, voice, and video. This protocol was chosen as the basis for the SIP because it was an open standard, could support all of the SMDS service features, was designed for compatibility with carrier transmission standards, and is aligned with emerging standards for *Broadband ISDN (BISDN)*. As BISDN technology matures and is deployed, the carriers intend to support not only SMDS but broadband video and voice services as well.

To interface to SMDS networks, only the connectionless data portion of the IEEE 802.6 protocol is needed. Therefore, the SIP does not define voice or video application support.

When used to gain access to an SMDS network, operation of the DQDB protocol across the SNI results in an *access DQDB*. The term *access DQDB* distinguishes operation of DQDB across the SNI from operation of DQDB in any other environment (such as inside the SMDS network). A switch in the SMDS network operates as one station on an access DQDB, while customer equipment operates as one or more stations on the access DQDB.

Because the DQDB protocol was designed to support a variety of data and nondata applications and because it is a shared-medium access-control protocol, it is relatively complex. It has two parts:

- The protocol syntax
- The distributed queuing algorithm that constitutes the shared medium access control

CPE Configurations

There are two possible configurations of CPE on the SMDS access DQDB (see Figure 15-2). In a single-CPE configuration, the access DQDB simply connects the switch in the carrier network and one subscriber-owned station (CPE). In a multi-CPE configuration, the access DQDB consists of the switch in the network and multiple interconnected CPE at the subscriber site. In this latter configuration, all CPE must belong to the same subscriber.

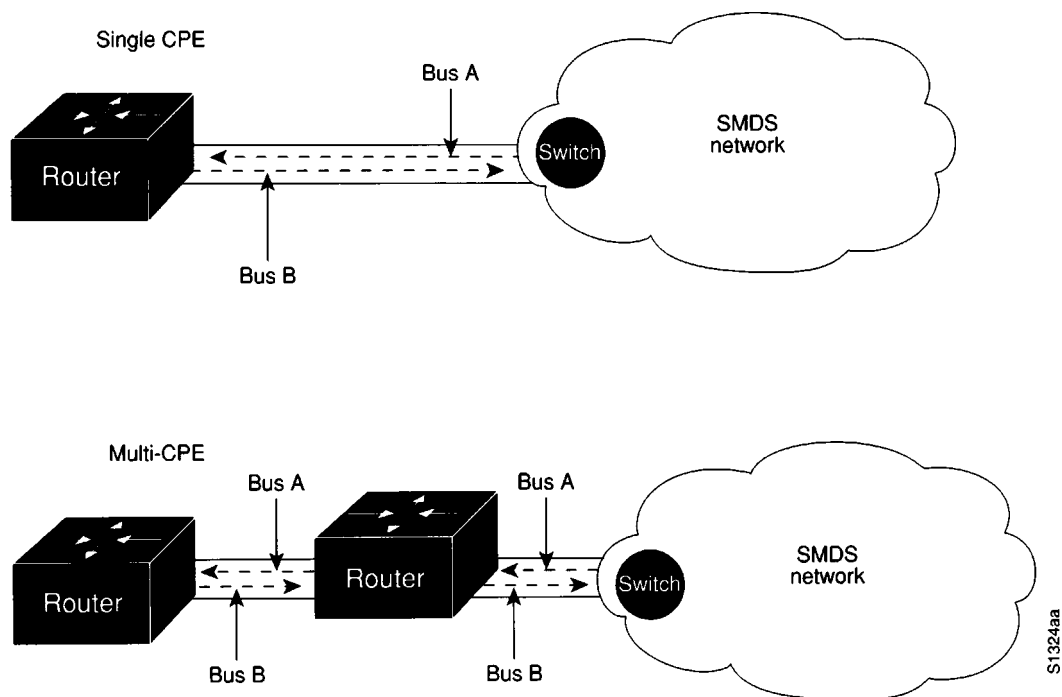
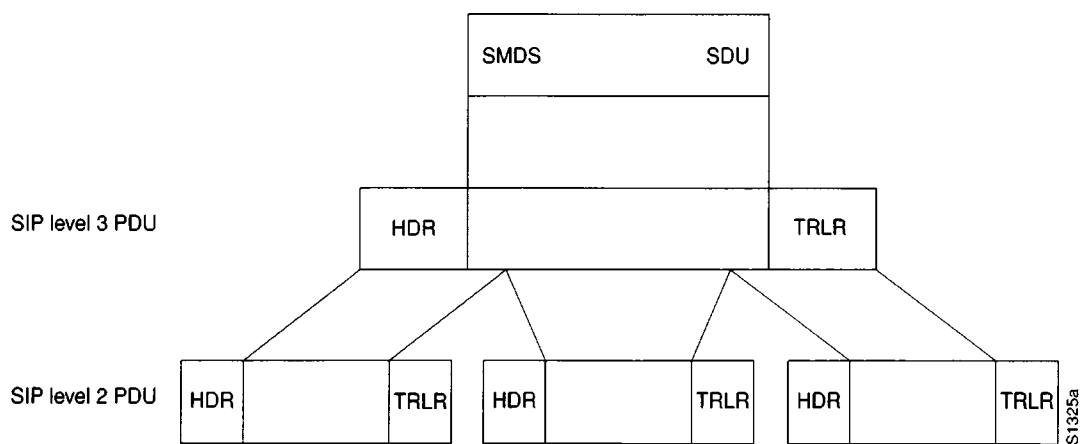


Figure 15-2 Single-CPE and Multi-CPE Configurations

In the single-CPE case, the access DQDB is essentially just a two-node DQDB subnetwork. Each of the nodes (the switch and the CPE) transfer data to the other via a unidirectional logical bus. There is no contention for this bus, since there are no other stations. Because of this, the distributed queuing algorithm need not be used. Without the complexity of the distributed queuing algorithm, SIP for single-CPE configurations is much simpler than SIP for multi-CPE configurations.

SIP Levels

The SIP can be logically partitioned into three levels, as shown in Figure 15-3.



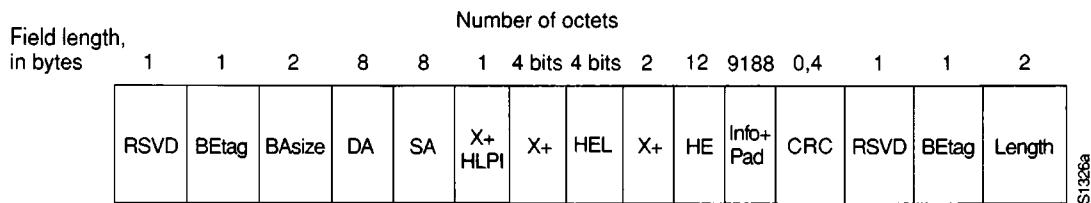
SDU = Service data unit
PDU = Protocol data unit
HDR = Header
TRLR = Trailer

Figure 15-3 Encapsulation of User Information by SIP Levels

Level 3

SIP level 3 operation involves the encapsulation of SMDS *service data units* (*SDUs*) in a level 3 header and trailer. Level 3 *protocol data units* (*PDUs*) are then broken into level 2 PDUs as appropriate to conform to level 2 specifications.

The SIP level 3 PDU is reasonably complex. It is depicted in Figure 15-4.



RSVD = Reserved
 BEtag = Beginning-end tag
 BAsize = Buffer allocation size
 DA = Destination address
 SA = Source address
 X+ = Carried across network unchanged
 HLPI = Higher-layer protocol identifier
 HEL = Header extension length
 HE = Header extension
 Info+Pad = Information + padding (to ensure that this field ends on a 32-bit boundary)
 CRC = Cyclic Redundancy Check

Figure 15-4 SIP Level 3 PDU

Fields marked with X+ in the figure are not used in the provision of SMDS, but are present in the protocol to ensure alignment of the SIP format with the DQDB protocol format. Values placed in these fields by the CPE must be delivered unchanged by the network.

The two *reserved* fields must be populated with zeros. The two *BEtag* fields contain an identical value and are used to form an association between the first and last segments or level 2 PDUs of a SIP level 3 PDU. These fields can be used to detect the condition where the last segment of one level 3 PDU and the first segment of the next level 3 PDU are both lost, resulting in receipt of an invalid level 3 PDU.

The *destination* and *source addresses* consist of two parts: an *address type* and an *address*. In both cases, the address type occupies the four most significant bits of the field. If the address is a destination address, the address type may be either “1100” or “1110.” The former indicates a 60-bit individual address, whereas the latter indicates a 60-bit group address. If the address is a source address, the address type field can only indicate an individual address.

Bellcore Technical Advisories specify how addresses consistent in format with the *North American Numbering Plan (NANP)* are to be encoded in the source and destination address fields. In this case, the four most significant bits of each of the source and destination address subfields contains the value “0001,” which is the internationally defined country code for North America. The next 40 bits contains the binary coded decimal (BCD)-encoded values of the 10-digit SMDS, NANP-aligned addresses. The final 16 (least-significant) bits are populated with ones for padding.

The *higher-layer protocol identifier* field indicates what type of protocol is encapsulated in the information field. This value is important to systems using the SMDS network (such as Cisco routers) but is not processed nor changed by the SMDS network.

The *header extension length (HEL)* field indicates the number of 32-bit words in the header extension field. Currently, the size of this field for SMDS is fixed at 12 bytes. Therefore, the HEL value is always “0011.”

The *header extension* field is currently identified as having two uses. One is to contain an SMDS version number, which is used to determine what version of the protocol is being used. The other use is to convey a carrier selection value providing the ability to select a particular interexchange carrier to carry SMDS traffic from one local carrier network to another. In the future, other information may be defined to be conveyed in the header extension field, if required.

Level 2

Level 3 PDUs are segmented into uniformly-sized (53-octet) level 2 PDUs, often referred to as *slots* or *cells*. The format of the SIP level 2 PDU is shown in Figure 15-5.

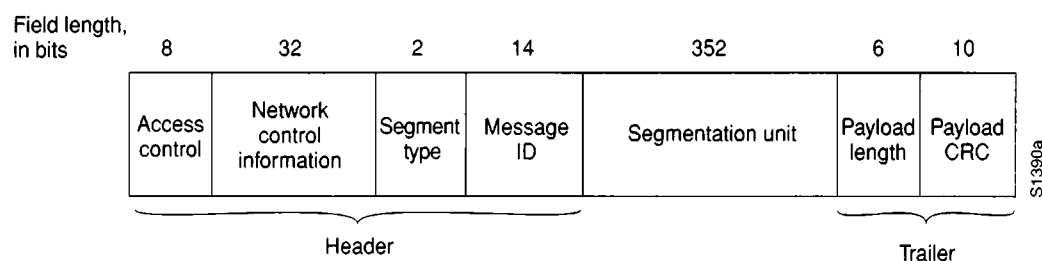


Figure 15-5 SIP Level 2 PDU

The *access control* field of the SIP level 2 PDU contains different values depending on the direction of information flow. If the slot is sent from the switch to the CPE, only the indication of whether the PDU contains information or not is important. If the slot was sent from the CPE to the switch, and if the configuration is multi CPE, this field can also carry request bits that indicate bids for slots on the bus going from the switch to the CPE. Further detail on how these request bits are used to implement the distributed queuing media access control can be obtained from the IEEE 802.6 standard.

The *network control information* field can contain only two possible values. One particular bit pattern is included when the PDU contains information; another is used when it does not.

The *segment type* field indicates whether this level 2 PDU is the beginning slot, the last slot, or a slot from the middle of a level 3 PDU. Segment type values are shown in Figure 15-6.

Value	Meaning
00	Continuation of message (COM)
01	End of message (EOM)
10	Beginning of message (BOM)
11	Single segment message (SSM)

Figure 15-6 Segment Type Values

The *message ID* field allows association of level 2 PDUs with a level 3 PDU. The message ID is the same for all segments of a given level 3 PDU. On a multi-CPE access DQDB, level 3 PDUs originating from different CPE must have different message IDs. This allows the SMDS network receiving interleaved slots from different level 3 PDUs to associate each level 2 PDU with the correct level 3 PDU. Successive level 3 PDUs from the same CPE may have identical message IDs. This presents no ambiguity, since any single CPE must send all level 2 PDUs from one level 3 PDU before it begins sending level 2 PDUs of a different level 3 PDU.

The *segmentation unit* field is the data portion of the PDU. In the event of an empty level 2 PDU, this field is populated with zeros.

The *payload length* field indicates how many bytes of a level 3 PDU are actually contained in the segmentation unit field. If the level 2 PDU is empty, this field is also populated with zeros.

Finally, the *payload CRC* field contains a 10-bit *cyclic redundancy check (CRC)* value used to detect errors over the segment type, message ID, segmentation unit, payload length, and payload CRC fields. This CRC does not cover the access-control or network-control information fields.

Level 1

SIP level 1 provides the physical link protocol, which operates at DS-3 or DS-1 rates between the CPE and the network. SIP level 1 is divided into two parts: the *transmission system* sublayer and the *Physical Layer Convergence Protocol (PLCP)*. The former defines the characteristics and method of attachment to the transmission link, that is, the DS-3 or DS-1. The latter specifies how the level 2 PDUs, or slots, are to be arranged relative to the DS-3 or DS-1 frame, and defines certain management information.

Because it is based on IEEE 802.6, the SIP has the advantage of compatibility with future BISDN interfaces that will support not only data but video and voice applications as well. However, this compatibility does cost some protocol overhead, which must be taken into account when calculating overall data throughput that can be achieved using SIP. Over a DS-3 access DQDB, the total bandwidth available for level 3 PDU user data is approximately 34 Mbps. Over a DS-1 access, approximately 1.2 Mbps can carry user data.

The use of the IEEE 802.6 MAN *media access control (MAC)* protocol as the basis for the SMDS SIP means that local communication between CPE on the same access DQDB is possible. Some of this local communication will be visible to the switch serving the SNI and some will not. The switch therefore must use the destination address of a data unit to differentiate between data units intended for SMDS-based transfer and data units intended for local transmission among multiple CPE sharing an access DQDB.

Network Implementation

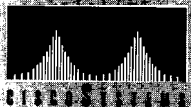
Inside the carrier network, the high-speed packet-switching capability required by SMDS can be provided by a number of different technologies. In the near term, switches based on MAN technology such as the DQDB standard are being included in a number of networks. A series of Technical Advisories produced by Bellcore specify standard requirements on network equipment for such functions as:

- Network operations
- Usage measurement for billing
- Interface between a local carrier network and a long distance carrier network
- Interface between two switches inside the same carrier's network
- Customer network management

As has been noted, the IEEE 802.6 protocol and SIP were intentionally designed to align with the principal BISDN protocol referred to as *Asynchronous Transfer Mode (ATM)*. ATM and IEEE 802.6 belong to a class of protocols often referred to as *fast packet switching* or *cell relay* protocols. These protocols organize information into small, fixed-size cells (Level 2 PDUs in SIP terminology). Fixed-size cells can be processed and switched in hardware at very high speeds. This tightly constrains delay characteristics, making cell relay protocols useful for video and voice applications. As ATM-based switching equipment becomes available, this technology will also be introduced into networks providing SMDS.

Part 4

Routed
Protocols





Chapter 16

AppleTalk

16

Background

In the early 1980s, Apple Computer was preparing to introduce the Macintosh computer. Apple engineers knew that networks would become a critical need rather than an interesting curiosity. They wanted to ensure that a Macintosh-based network was a seamless extension of the revolutionary Macintosh user interface. With these two goals in mind, Apple decided to build a network interface into every Macintosh and to integrate that interface into the desktop environment. Apple's new network architecture was given the name *AppleTalk*.

Although AppleTalk is a proprietary network, Apple has published AppleTalk specifications in an attempt to encourage third-party development. Today, many companies are successfully marketing AppleTalk-based products, including Novell, Inc. and Microsoft Corporation.

The original implementation of AppleTalk, designed for local work groups, is now commonly referred to as *AppleTalk Phase I*. With the installation of over 1.5 million Macintosh computers in the first five years of the product's life, however, Apple found that some large corporations were exceeding the built-in limits of AppleTalk Phase I, so they enhanced the protocol. The enhanced protocols came to be known as *AppleTalk Phase II*. AppleTalk Phase II enhanced AppleTalk's routing capabilities and allowed AppleTalk to run successfully in larger networks.

Technology Basics

AppleTalk was designed as a client-server distributed network system. In other words, users share network resources (such as files and printers) with other users. Computers supplying these network resources are called *servers*; computers using a server's network resources are called *clients*. Interaction with servers is essentially transparent to the user because the computer itself determines the location of the requested material and accesses it without further information from the user. In addition to their ease-of-use, distributed systems also enjoy an economic advantage over peer-to-peer systems because important materials can be located in a few, rather than many locations.

AppleTalk corresponds relatively well to the OSI reference model. In Figure 16-1, AppleTalk protocols are shown adjacent to the OSI layers to which they map. This figure differs from some other depictions of how the AppleTalk protocol stack relates to the OSI model in that it places NBP, ZIP, and RTMP at Layer 3 and AEP at Layer 7. Cisco believes that NBP, ZIP, and RTMP are more closely aligned (functionally) to Layer 3 of the OSI model, even though they use the services of DDP, another Layer 3 protocol. Similarly, Cisco believes that AEP

should be listed as an application-layer protocol because it is commonly used to provide application-layer functionality. Specifically, AEP helps determine the ability of remote nodes to receive incoming connections.

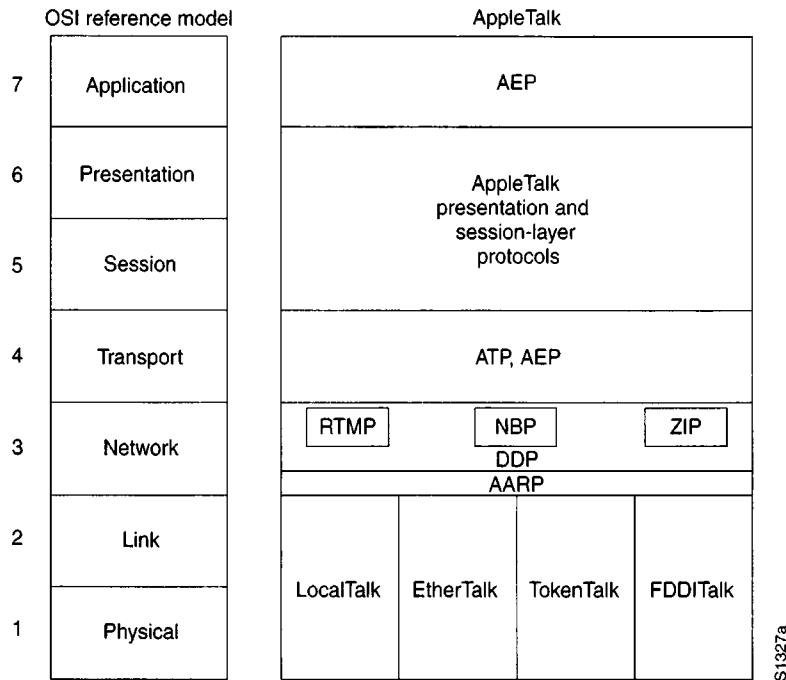


Figure 16-1 AppleTalk and the OSI Reference Model

Media Access

Apple constructed AppleTalk to be link-layer independent. In other words, it can theoretically run on top of any link-layer implementation. Apple supports a variety of link-layer implementations, including Ethernet, Token Ring, FDDI, and LocalTalk. Apple refers to AppleTalk over Ethernet as *EtherTalk*, to AppleTalk over Token Ring as *TokenTalk*, and to AppleTalk over FDDI as *FDDITalk*. For more information on Ethernet, Token Ring, and FDDI technical characteristics, see Chapter 5, "Ethernet/IEEE 802.3," Chapter 6, "Token Ring/IEEE 802.5," and Chapter 7, "FDDI," respectively.

LocalTalk is Apple's proprietary media-access system. It is based on contention access, bus topology, and baseband signaling, and runs on shielded twisted-pair media at 230.4 Kbps. The physical interface is RS-422, a balanced electrical interface supported by RS-449. LocalTalk segments can span up to 300 meters and support a maximum of 32 nodes.

Network Layer

This section describes AppleTalk network-layer concepts and protocols. Included are discussions of protocol address assignment, network entities, and AppleTalk protocols that provide OSI reference model Layer 3 functionality.

Protocol Address Assignment

To ensure minimal network administrator overhead, AppleTalk node addresses are assigned dynamically. When a Macintosh running AppleTalk starts up, it chooses a protocol (network-layer) address and checks to see whether that address is currently in use. If not, the new node has successfully assigned itself an address. If the address is currently in use, the node with the conflicting address sends a message indicating a problem, and the new node chooses another address and repeats the process. Figure 16-2 shows the AppleTalk address selection process.

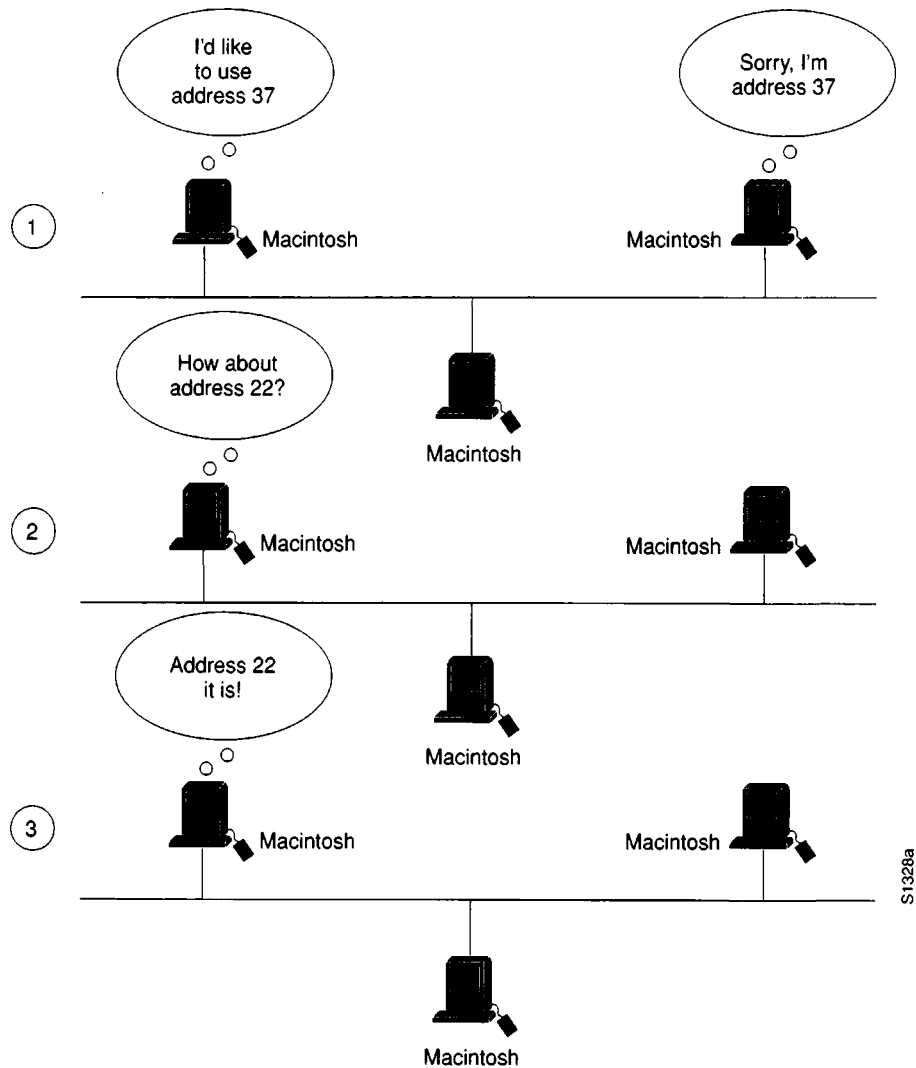


Figure 16-2 AppleTalk Address Selection Process

The actual mechanics of AppleTalk address selection are media-dependent. The *AppleTalk Address Resolution Protocol (AARP)* is used to associate AppleTalk addresses with particular media addresses. AARP also associates other protocol addresses with hardware addresses. When either AppleTalk or any other protocol stack must send a packet to another network node, the protocol address is passed to AARP. AARP first checks an address cache to see whether the protocol address/hardware address relationship is already known. If so, that relationship is passed up to inquiring protocol stack. If not, AARP initiates a broadcast or multicast message inquiring about the hardware address for the protocol address in question. If the broadcast reaches a node with the specified protocol address, that node replies with its hardware address. This information is passed up to the inquiring protocol stack, which uses the hardware address in communications with that node.

Network Entities

AppleTalk identifies several network entities. The most elemental is a *node*, which is simply any device connected to an AppleTalk network. The most common nodes are Macintosh computers and laser printers, but many other types of computers are also capable of AppleTalk communication, including IBM PCs, Digital Equipment Corporation VAX computers and a variety of workstations. The next entity defined by AppleTalk is the *network*. An AppleTalk network is simply a single logical cable. Although the logical cable is frequently a single physical cable, some sites use bridges to interconnect several physical cables. Finally, an AppleTalk *zone* is a logical group of (possibly noncontiguous) networks. These AppleTalk entities are pictured in Figure 16-3.

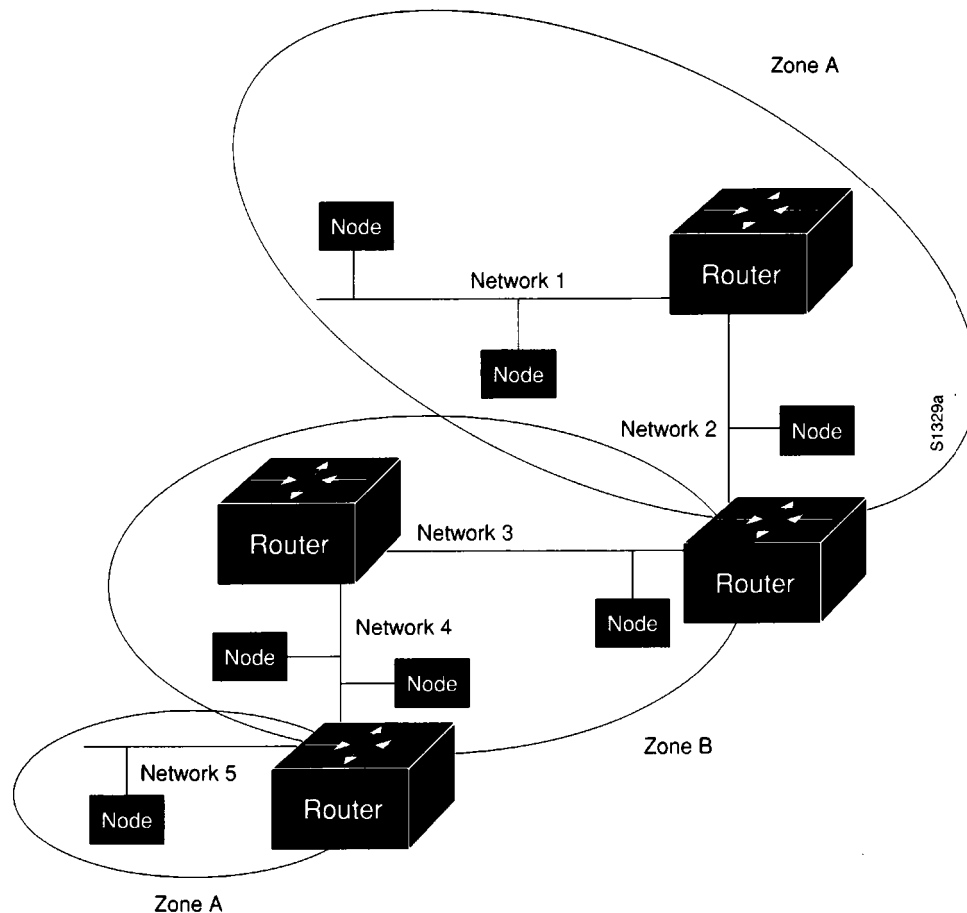


Figure 16-3 AppleTalk Entities

Datagram Delivery Protocol (DDP)

AppleTalk's primary network-layer protocol is the *Datagram Delivery Protocol (DDP)*. DDP provides connectionless service between network sockets. Sockets can be assigned either statically or dynamically.

AppleTalk addresses, which are administered by DDP, consist of two components: a 16-bit *network number* and an 8-bit *node number*. The two components are usually written as decimal numbers, separated by a period (for example, 10.1 means network 10, node 1). When an 8-bit *socket* identifying a particular process is added to the network number and node number, a unique process on a network is specified.

AppleTalk Phase II distinguishes between *nonextended* and *extended* networks. In a nonextended network such as LocalTalk, each AppleTalk node number is unique. Nonextended networks were the sole network type defined in AppleTalk Phase I. In an extended network such as EtherTalk and TokenTalk, each network number/node number combination is unique.

Zones are defined by the AppleTalk network manager during the router configuration process. Each node in an AppleTalk network belongs to a single specific zone. Extended networks can have multiple zones associated with them. Nodes on extended networks can belong to any single zone associated with the extended network.

Routing Table Maintenance Protocol (RTMP)

The protocol that establishes and maintains AppleTalk routing tables is called the *Routing Table Maintenance Protocol (RTMP)*. RTMP routing tables contain an entry for each network that a datagram can reach. Each entry includes the router port that leads to the destination network, the node ID of the next router to receive the packet, the distance in hops to the destination network, and the current state of the entry (good, suspect, or bad). Periodic exchange of routing tables allows the routers in an internet to ensure that they supply current and consistent information. Figure 16-4 shows a sample RTMP table and the corresponding network architecture.

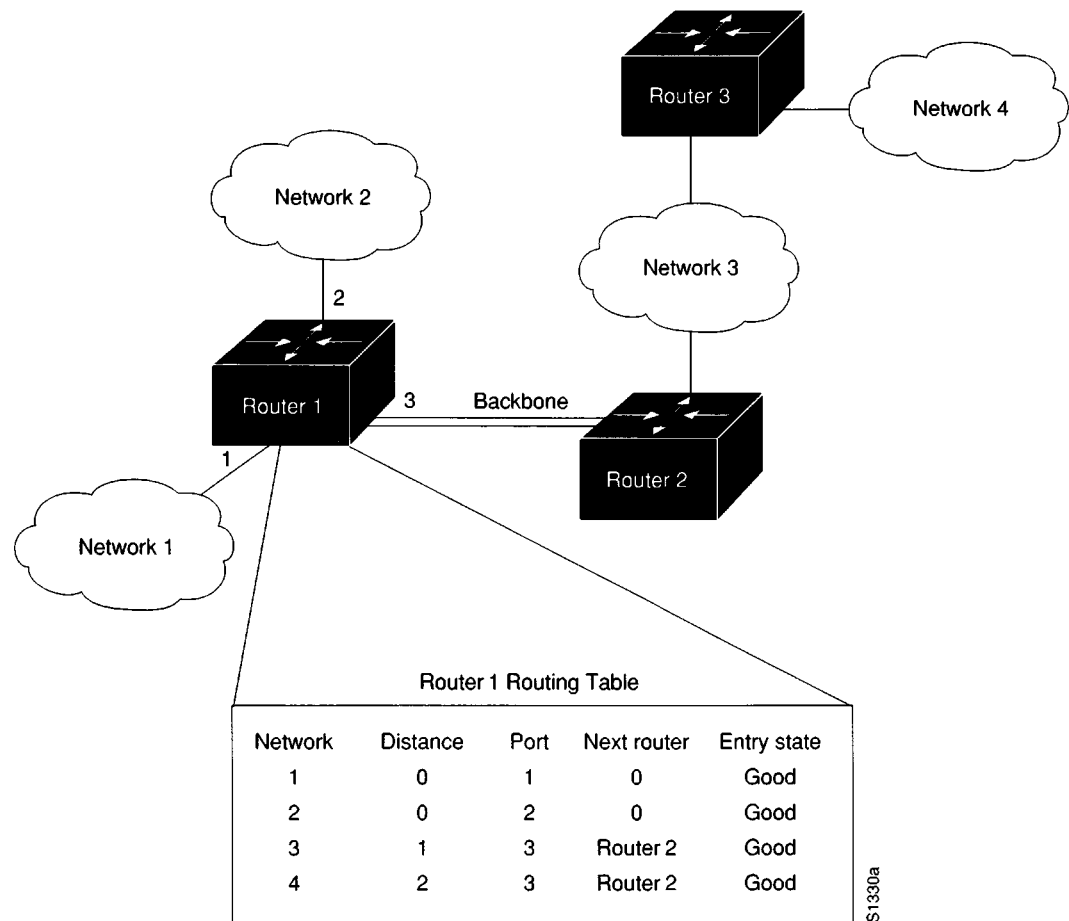


Figure 16-4 Sample AppleTalk Routing Table

AppleTalk's *Name Binding Protocol (NBP)* associates AppleTalk names (expressed as *network-visible entities*, or *NVEs*) with addresses. An NVE is an AppleTalk network-addressable service, such as a socket. NVEs are associated with one or more entity names and attribute lists. Entity names are character strings such as *printer@net1*, while attribute lists specify NVE characteristics.

Named NVEs are associated with network addresses through the process of name binding. Name binding can be done when the user node is first started up or dynamically, immediately before first use. NBP orchestrates the name binding process, which includes name registration, name confirmation, name deletion, and name lookup.

Zones allow name lookup in a group of logically related nodes. To look up names within a zone, an NBP lookup request is sent to a local router, which sends a broadcast request to all networks that have nodes belonging to the target zone. The *Zone Information Protocol (ZIP)* coordinates this effort.

ZIP maintains network number/zone name mappings in *zone information tables (ZITs)*. ZITs are stored in routers, which are the primary users of ZIP, but end nodes use ZIP during the startup process to choose their zone and to acquire internetwork zone information. ZIP uses RTMP routing tables to keep up with network topology changes. When ZIP finds a routing table entry that is not in the ZIT, it creates a new ZIT entry. Figure 16-5 shows a sample ZIT.

Network number	Zone
1	My
2	Your
3	Marketing
4	Documentation
5-5	Sales

S1331a

Figure 16-5 Sample AppleTalk ZIT

Transport Layer

AppleTalk's transport layer is implemented by two primary Apple protocols: *AppleTalk Transaction Protocol (ATP)* and *AppleTalk Data Stream Protocol (ADSP)*. ATP is transaction oriented, while ADSP is data-stream oriented.

AppleTalk Transaction Protocol (ATP)

ATP is one of AppleTalk's transport-layer protocols. ATP is suitable for transaction-based applications such as those found in banks or retail stores.

ATP transactions consist of *requests* (from clients) and *replies* (from servers). Each request/reply pair has a particular *transaction ID*. Transactions occur between two socket clients. ATP uses *exactly-once (XO)* and *at-least-once (ALO)* transactions. XO transactions are required in those situations where accidentally performing the transaction more than once is unacceptable. Bank transactions are examples of such *nonidempotent* situations (situations where repeating a transaction causes problems by invalidating the data involved in the transaction).

ATP is capable of most important transport-layer functions, including data acknowledgment and retransmission, packet sequencing, and fragmentation and reassembly. ATP limits message segmentation to 8 packets, and ATP packets cannot contain more than 578 data bytes.

AppleTalk Data Stream Protocol (ADSP)

ADSP is another important AppleTalk transport-layer protocol. As its name implies, ADSP is data-stream rather than transaction oriented. It establishes and maintains full-duplex data streams between two sockets in an AppleTalk internetwork.

ADSP is a reliable protocol in that it guarantees that data bytes will be delivered in the same order as they were sent and that they are not duplicated. ADSP numbers each data byte to keep track of the individual elements of the data stream.

ADSP also specifies a flow-control mechanism. The destination can essentially slow source transmissions by reducing the size of its advertised receive window.

ADSP also provides an out-of-band control message mechanism. Attention packets are used as the vehicle for movement of out-of-band control messages between two AppleTalk entities. These packets use a separate sequence number stream to differentiate them from normal ADSP data packets.

Upper-Layer Protocols

AppleTalk supports several upper-layer protocols. The *AppleTalk Session Protocol (ASP)* establishes and maintains sessions (logical conversations) between an AppleTalk client and a server. AppleTalk's *Printer Access Protocol (PAP)* is a connection-oriented protocol that establishes and maintains connections between clients and servers. (Use of the term *printer* in this protocol's title is purely historical.) The *AppleTalk Echo Protocol (AEP)* is an extremely simple protocol that generates packets that can be used to test the reachability of various network nodes. Finally, the *AppleTalk Filing Protocol (AFP)* helps clients share server files across a network.

Chapter 17

DECnet

17

Background

Digital Equipment Corporation (Digital) developed the *DECnet* protocol family to provide a well-thought-out way for its computers to communicate with one another. The first version of DECnet, released in 1975, allowed two directly attached PDP-11 minicomputers to communicate. In more recent years, Digital has included support for nonproprietary protocols, but DECnet remains the most important of Digital's network product offerings.

DECnet is currently in its fifth major product release (sometimes called *Phase V* and referred to as *DECnet/OSI* in Digital literature). DECnet Phase V is a proper superset of the OSI protocol suite, supporting all OSI protocols as well as several other proprietary and standard protocols supported in previous versions of DECnet. As with past changes to the protocol, DECnet Phase V is compatible with the previous release (Phase IV, in this case).

Digital Network Architecture (DNA)

Contrary to popular belief, DECnet is not a network architecture at all, but rather a series of products conforming to Digital's *Digital Network Architecture (DNA)*. Like most comprehensive network architectures from large systems vendors, DNA supports a large set of both proprietary and standard protocols. The list of DNA-supported technologies grows constantly as Digital implements new protocols. Figure 17-1 illustrates an incomplete snapshot of DNA and the relationship of some of its components to the OSI reference model.

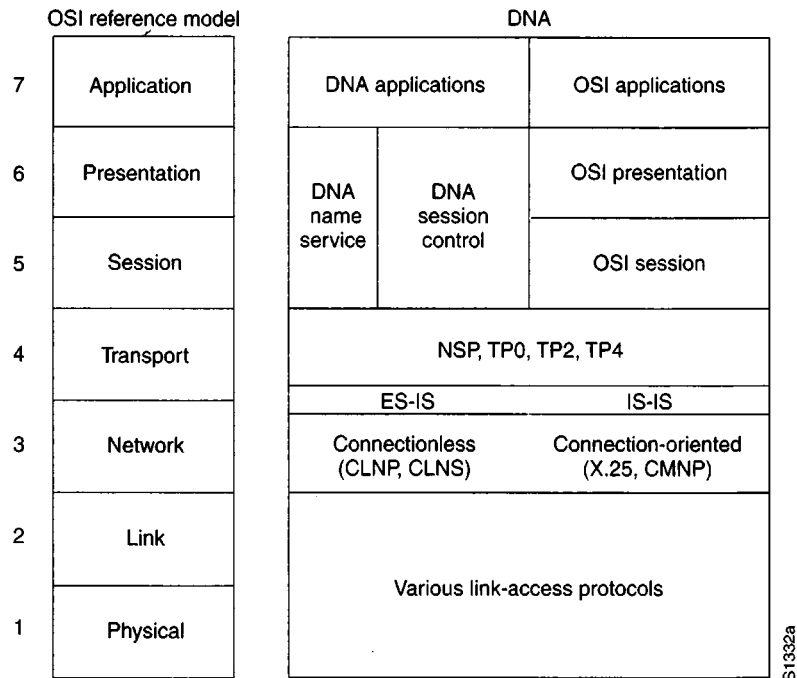


Figure 17-1 DNA and the OSI Reference Model

Media Access

As Figure 17-1 shows, DNA supports a variety of media and link implementations. Among these are well-known standards such as *Ethernet*, *Token Ring*, *Fiber Distributed Data Interface (FDDI)*, *IEEE 802.2*, and *X.25*. See Chapter 5, “Ethernet/IEEE 802.3,” Chapter 6, “Token Ring/IEEE 802.5,” Chapter 7, “FDDI,” Chapter 12, “SDLC and Derivatives,” and Chapter 13, “X.25,” for more information on these protocols. DNA also offers a traditional point-to-point link-layer protocol called *Digital Data Communications Message Protocol (DDCMP)* and a 70-Mbps bus used in the VAXcluster called the *Computer-room Interconnect bus (CI bus)*.

Network Layer

DECnet supports both connectionless and connection-oriented network layers. Both network layers are implemented by OSI protocols. The connectionless implementation uses the *Connectionless Network Protocol (CLNP)* and the *Connectionless Network Service (CLNS)*. The connection-oriented network layer uses the *X.25 Packet-Level Protocol (PLP)*, which is also known as *X.25 level 3*, and the *Connection-Mode Network Protocol (CMNP)*. These OSI protocols are described more completely in Chapter 20, “OSI Protocols.”

Although most of DNA was brought into OSI conformance with DECnet Phase V, DECnet Phase IV routing was already very similar to OSI routing. Phase V DNA routing consists of OSI routing (*ES-IS* and *IS-IS*), plus continued support for the *DECnet Phase IV routing protocol*. *ES-IS* and *IS-IS* are described in Chapter 28, “OSI Routing.”

DECnet Phase IV Routing Frame Format

The DECnet Phase IV routing protocol differs from *IS-IS* in several ways. One difference is in the protocol header. The DNA Phase IV routing layer header appears in Figure 17-2; *IS-IS* packet formats are shown in Chapter 28, “OSI Routing.”

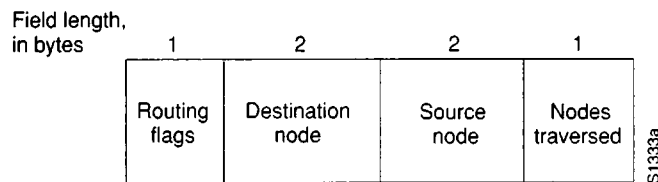


Figure 17-2 DNA Phase IV Routing Layer Header

The first field in a DNA Phase IV routing header is the *routing flags* field, which includes:

- A *return-to-sender* bit which, if set, indicates that the packet is returning to the source.
- A *return-to-sender-request* bit which, if set, indicates that request packets should be returned to the source if they cannot be delivered to the destination.
- An *intraLAN* bit, which is on by default. If the router detects that the two communicating end systems are not on the same subnetwork, it turns the bit off.
- Other bits that indicate header format, whether padding is being used, and other functions.

Following the routing flags field are the *destination node* and *source node* fields, which identify the network addresses of the destination nodes and the source node

The final field in the DNA Phase IV routing header is the *nodes traversed* field, which shows the number of nodes the packet has traversed on its way to the destination. This field allows implementation of a maximum hop count, so that obsolete packets can be removed from the network.

DECnet identifies two types of nodes: *end nodes* and *routing nodes*. Both end nodes and routing nodes can send and receive network information, but only routing nodes can provide routing services for other DECnet nodes.

DECnet routing decisions are based on *cost*, an arbitrary measure assigned by network administrators to be used in comparing various paths through an internetwork environment. Costs are typically based on hop count, media bandwidth, or other measures. The lower the cost, the better the path. When network faults occur, the DECnet Phase IV routing protocol uses cost values to recalculate the best paths to each destination. Figure 17-3 illustrates the calculation of costs in a DECnet Phase IV routing environment.

Best Path to Destination:

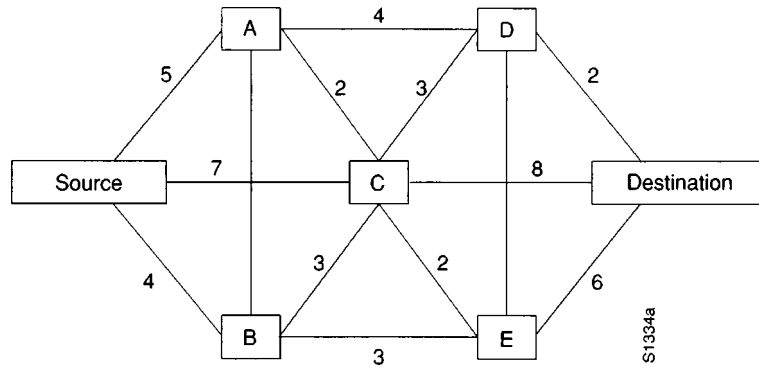
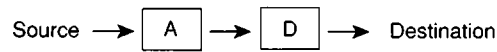


Figure 17-3 DECnet Phase IV Routing Protocol Cost Calculation

Addressing

DECnet addresses are not associated with the physical networks to which the nodes are connected. Instead, DECnet locates hosts using *area/node address* pairs. An area's value ranges from 1 to 63, inclusive. A node address can be between 1 and 1023, inclusive. Therefore, each area can have 1023 nodes and approximately 65,000 nodes can be addressed in a DECnet network. Areas can span many routers, and a single cable can support many areas. Therefore, if a node has several network interfaces, it uses the same area/node address for each interface. Figure 17-4 shows a sample DECnet network with several addressable entities.

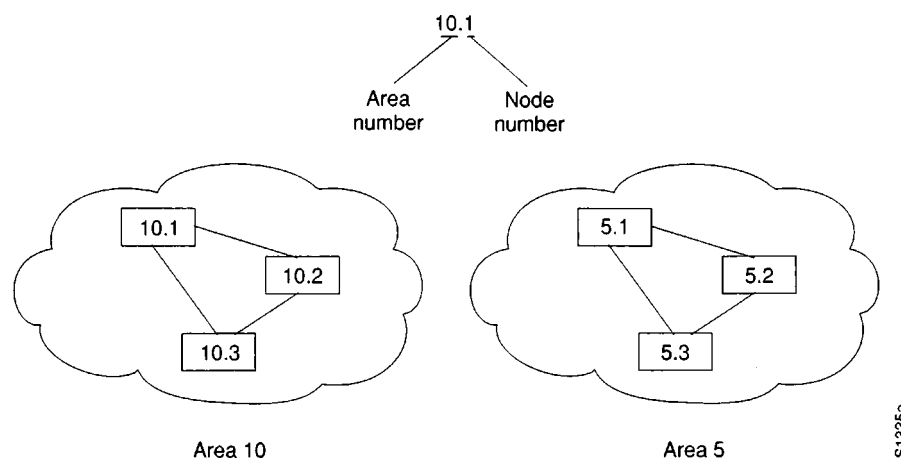


Figure 17-4 DECnet Addresses

DECnet hosts do not use manufacturer-assigned *Media Access Control (MAC)*-layer addresses. Instead, network-level addresses are embedded in the MAC-layer address according to an algorithm that multiplies the area number by 1024 and adds the node number to the product. The resulting 16-bit decimal address is converted to a hexadecimal number and appended to the address AA00.0400 in byte-swapped order, with the least significant byte first. For example, DECnet address 12.75 becomes 12363 (base 10), which equals 304B (base 16). After this byte-swapped address is appended to the standard DECnet MAC address prefix, the resulting address is AA00.0400.4B30.

Routing Levels

DECnet routing nodes are referred to as either *Level 1* or *Level 2* routers. A Level 1 router communicates with end nodes and with other Level 1 routers in a particular area. Level 2 routers communicate with Level 1 routers in the same area and with Level 2 routers in different areas. Together, then, Level 1 and Level 2 routers form a hierarchical routing scheme. This relationship is illustrated in Figure 17-5.

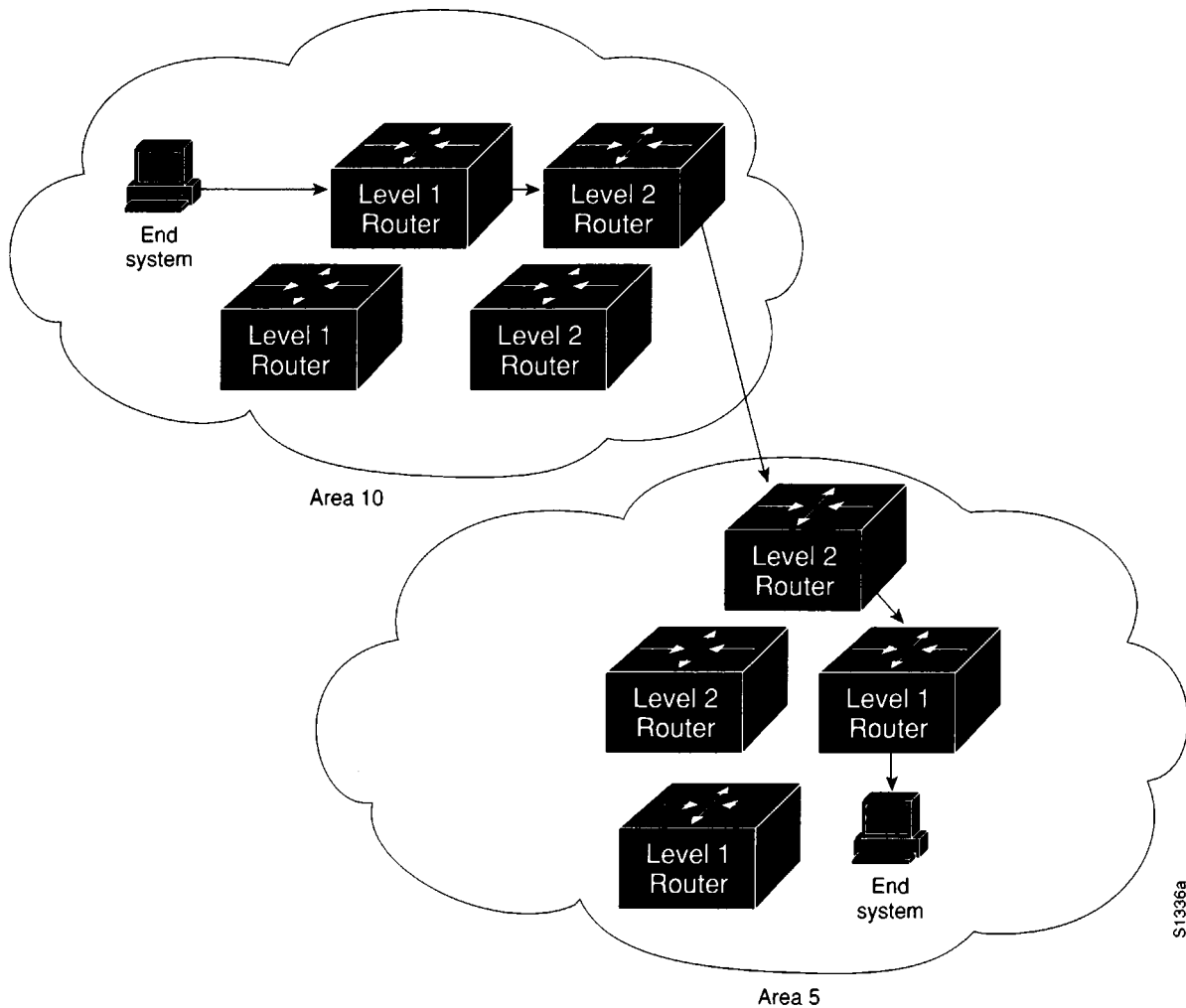


Figure 17-5 DECnet Level 1 and Level 2 Routers

End systems send routing requests to a designated Level 1 router. The Level 1 router with the highest priority is elected to be the designated router. If two routers have the same priority, the one with the larger node number becomes the designated router. A router's priority can be manually configured to force it to become the designated router.

As shown in Figure 17-5, multiple Level 2 routers can exist in any area. When a Level 1 router wishes to send a packet outside of its area, it forwards the packet to a Level 2 router in the same area. In some cases, this Level 2 router may not have the optimal path to the destination, but the mesh network configuration offers a degree of fault tolerance not provided by the simple assignment of one Level 2 router per area.

Transport Layer

The DNA transport layer is implemented by a variety of transports, both proprietary and standard. OSI transports TP0, TP2, and TP4 are supported. These are described in greater detail in Chapter 20, "OSI Protocols."

Digital's own *Network Services Protocol (NSP)* is functionally similar to TP4 in that it offers connection-oriented, flow-controlled service with message fragmentation and reassembly. Two subchannels are supported—one for normal data and one for expedited data and flow control information. Two flow control types are supported—a simple start/stop mechanism where the receiver tells the sender when to terminate and resume data transmission and a more complex flow control technique where the receiver tells the sender how many messages it can accept. NSP can also respond to congestion notifications from the network layer by reducing the number of outstanding messages it will tolerate.

Upper-Layer Protocols

Above the transport layer, DECnet supports its own proprietary upper-layer protocols as well as standard OSI upper-layer protocols. DECnet application protocols use the DNA session control protocol and the DNA name service. OSI application protocols are supported by OSI presentation and session layer implementations. See Chapter 20, "OSI Protocols," for more information on these OSI protocols.

Chapter 18

Internet Protocols

18

Background

In the mid-1970s, the Defense Advanced Research Project Agency (DARPA) became interested in establishing a packet-switched network to provide communications between the research institutions in the United States. DARPA and other government organizations understood the potential of packet-switched technology, and they were just beginning to face the problem virtually all companies with networks now have—that of communication between dissimilar computer systems.

With the goal of heterogeneous connectivity in mind, DARPA funded research by Stanford University and Bolt, Beranek, and Newman (BBN) to create a series of communication protocols. The result of this development effort, completed in the late 1970s, was the Internet protocol suite, of which the *Transmission Control Protocol (TCP)* and the *Internet Protocol (IP)* are the two best-known members.

The Internet protocols can be used to communicate across any set of interconnected networks. They are equally well suited for local area network (LAN) as well as wide area network (WAN) communications. The Internet suite includes not only lower-layer specifications (like TCP and IP), but also specifications for such common applications as mail, terminal emulation, and file transfer. Figure 18-1 shows some of the more important Internet protocols and their relationship to the OSI reference model.

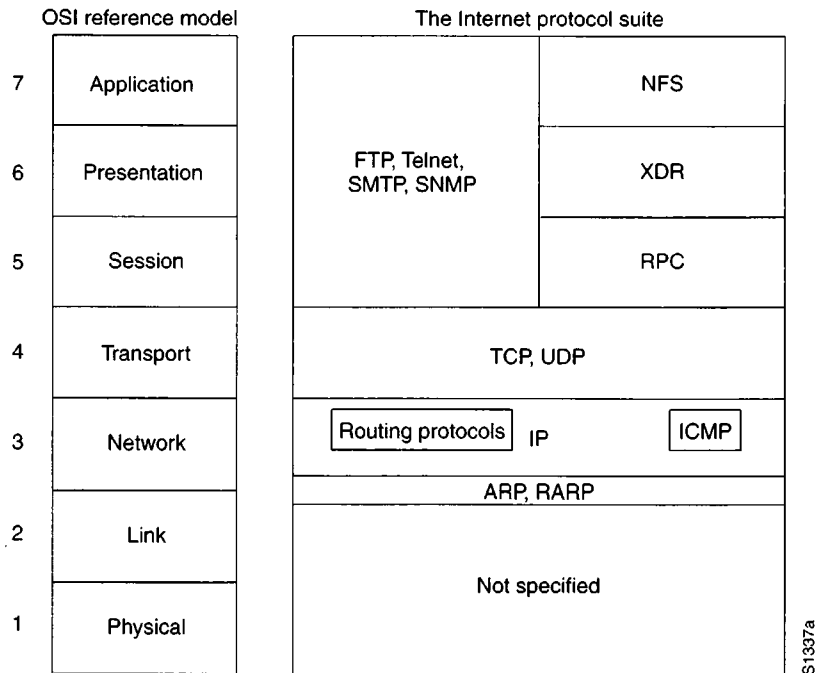
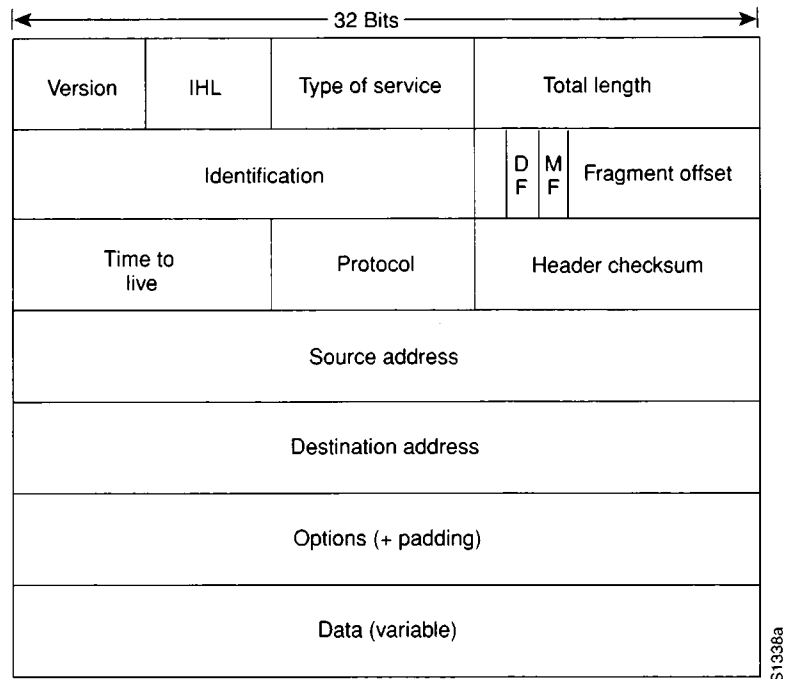


Figure 18-1 Internet Protocol Suite and the OSI Reference Model

Creation and documentation of the Internet protocols more closely resembles an academic research project than anything else. The protocols are specified in documents called *Requests for Comments (RFCs)*. RFCs are published and then reviewed and analyzed by the Internet community. Protocol refinements are published in new RFCs. Taken together, the RFCs provide a colorful history of the people, companies, and trends that shaped the development of what is today the world's most popular open system protocol suite.

Network Layer

IP is the primary Layer 3 protocol in the Internet suite. In addition to internetwork routing, IP provides fragmentation and reassembly of datagrams and error reporting. Along with TCP, IP represents the heart of the Internet protocol suite. The IP packet format is shown in Figure 18-2.



S1938a

Figure 18-2 IP Packet Format

The IP header begins with a *version number*, which indicates the version of IP currently used.

The *IP header length* (IHL) field indicates the datagram header length in 32-bit words.

The *type-of-service* field specifies how a particular upper-layer protocol would like the current datagram to be handled. Datagrams can be assigned various levels of importance through this field.

The *total length* field specifies the length of the entire IP packet, including data and header, in bytes.

The *identification* field contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

The *flags* field (containing a DF bit, an MF bit, and a fragment offset) specifies whether the datagram can be fragmented and whether the current fragment is the last fragment.

The *time-to-live* field maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

The *protocol* field indicates which upper-layer protocol receives incoming packets after IP processing is complete.

The *header checksum* field helps ensure IP header integrity.

The *source* and *destination address* fields specify the sending and receiving nodes.

The *options* field allows IP to support various options, such as security.

The *data* field contains upper-layer information.

Addressing

As with all network-layer protocols, IP's addressing scheme is integral to the process of routing IP datagrams through an internetwork. An IP address is 32 bits in length, divided into either two or three parts. The first part designates the network address, the second part (if present) designates the subnet address, and the final part designates the host address. Subnet addresses are only present if the network administrator has decided that the network should be divided into subnetworks. The lengths of the network, subnet, and host fields are all variable.

IP addressing supports five different network classes. The leftmost bits indicate the network class.

- *Class A* networks are intended mainly for use with a few very large networks, since they provide only 7 bits for the network address field.
- *Class B* networks allocate 14 bits for the network address field and 16 bits for the host address field. This address class offers a good compromise between network and host address space.
- *Class C* networks allocate 22 bits for the network address field. *Class C* networks only provide 8 bits for the host field, however, so the number of hosts per network may be a limiting factor.
- *Class D* addresses are reserved for multicast groups, as described formally in RFC 1112. In class D addresses, the four highest-order bits are set to 1,1,1, and 0.
- *Class E* addresses are also defined by IP, but reserved for future use. In class E addresses, the four highest-order bits are all set to 1.

IP addresses are written in dotted decimal format, for example, 34.0.0.1. Figure 18-3 shows the address formats for class A, B, and C IP networks.

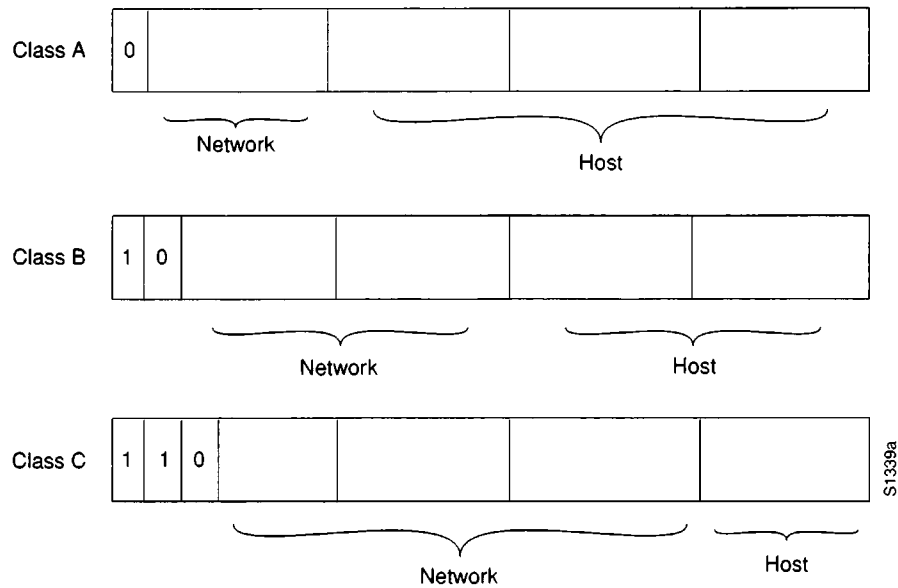


Figure 18-3 Class A, B, and C Address Formats

IP networks can also be divided into smaller units, called *subnets*. Subnets provide extra flexibility for the network administrator. For example, assume that a network has been assigned a class B address and all the nodes on the network currently conform to a class B address format. Then assume that the dotted decimal representation of this network's address is 128.10.0.0 (all zeros in the host field of an address specifies the entire network). Rather than change all the addresses to some other basic network number, the administrator can subdivide the network using subnetting. This is done by borrowing bits from the host portion of the address and using them as a subnet field, as depicted in Figure 18-4.

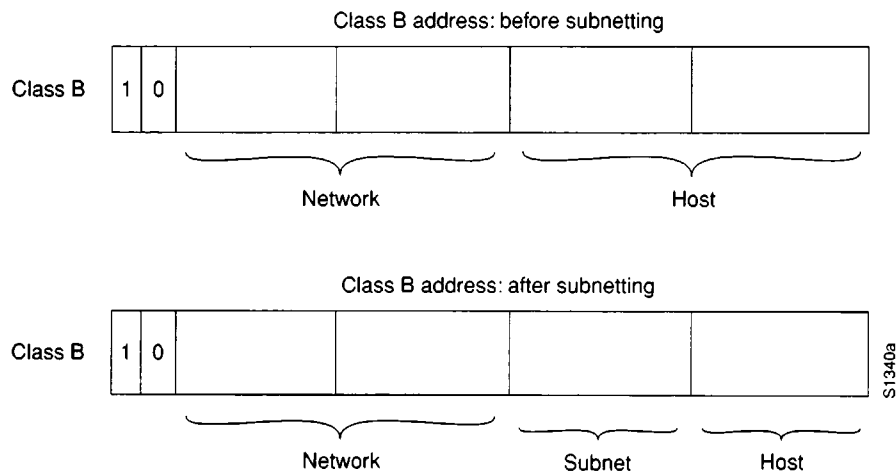


Figure 18-4 Subnet Addresses

If the network administrator has chosen to use eight bits of subnetting, the third octet of a class B IP address provides the subnet number. In our example, address 128.10.1.0 refers to network 128.10, subnet 1, address 128.10.2.0 refers to network 128.10, subnet 2, and so on.

The number of bits borrowed for the subnet address is variable. To specify how many bits are used, IP provides the subnet mask. Subnet masks use the same format and representation technique as do IP addresses. Subnet masks have ones in all bits except those bits that specify the host field. For example, the subnet mask that specifies 8 bits of subnetting for class A address 34.0.0.0 is 255.255.0.0. The subnet mask that specifies 16 bits of subnetting for class A address 34.0.0.0 is 255.255.255.0. Both of these subnet masks are pictured in Figure 18-5.

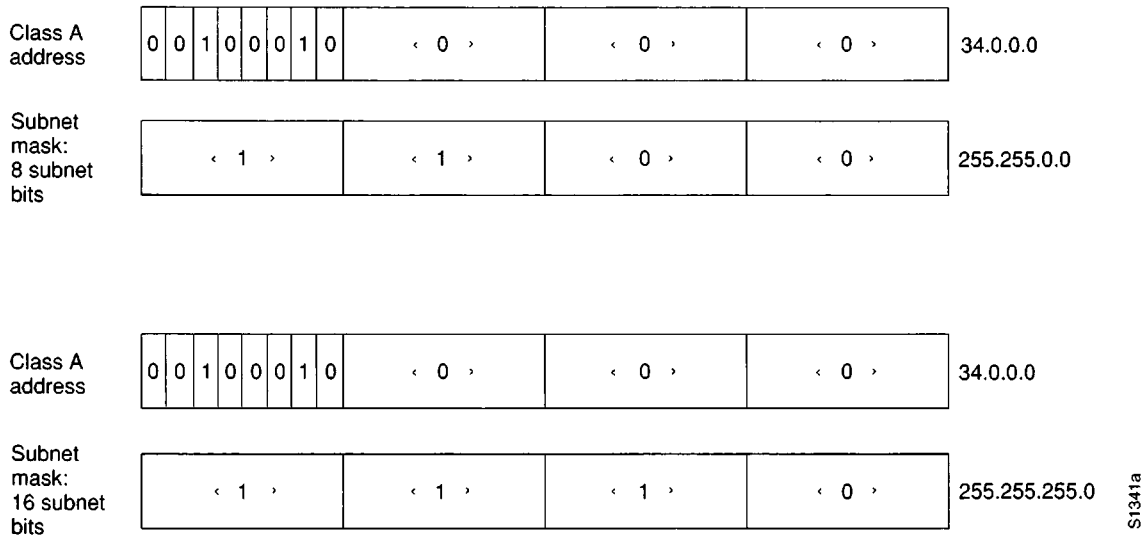


Figure 18-5 Sample Subnet Mask

On some media (such as IEEE 802 LANs), media addresses and IP addresses are dynamically discovered through the use of two other members of the Internet protocol suite: the *Address Resolution Protocol (ARP)* and the *Reverse Address Resolution Protocol (RARP)*. ARP uses broadcast messages to determine the hardware (MAC-layer) address corresponding to a particular internetwork address. ARP is sufficiently generic to allow use of IP with virtually any type of underlying media-access mechanism. RARP uses broadcast messages to determine the internet address associated with a particular hardware address. RARP is particularly important to diskless nodes, which may not know their internetwork address when they boot.

Internet Routing

Routing devices in the Internet have traditionally been called *gateways*—an unfortunate term since, elsewhere in the industry, the term applies to a device with somewhat different functionality. Gateways (which we will call routers from this point on) within the Internet are organized hierarchically. Some routers are used to move information through one particular group of networks under the same administrative authority and control (such an entity is called an *autonomous system*). Routers used for information exchange within autonomous systems are called *interior routers*, and they use a variety of *interior gateway protocols (IGPs)* to accomplish this purpose. Routers that move information between autonomous systems are called *exterior routers*, and they use an exterior gateway protocol for this purpose. The Internet architecture is shown in Figure 18-6.

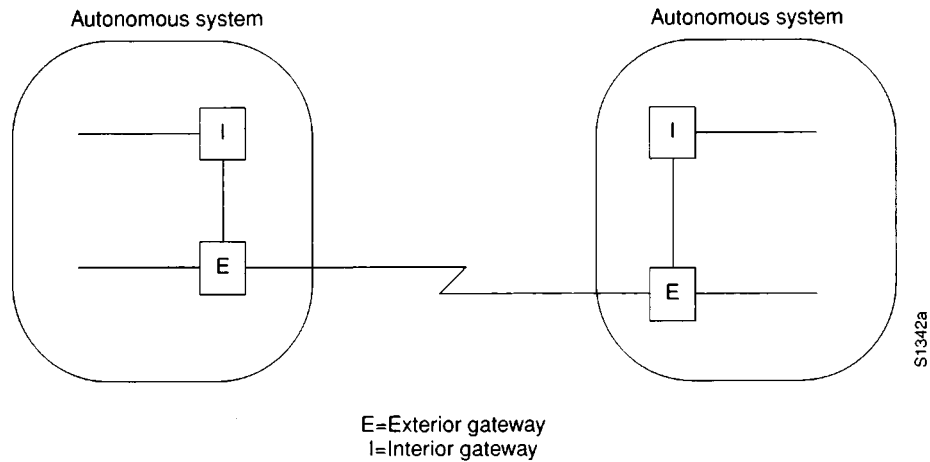


Figure 18-6 Internet Architecture

IP routing protocols are dynamic. *Dynamic routing* calls for routes to be calculated at regular intervals by software in the routing devices. This contrasts with *static routing*, where routes are established by the network administrator and do not change until the network administrator changes them. An IP routing table consists of *destination address/next hop* pairs. A sample entry, shown in Figure 18-7, is interpreted as meaning “to get to network 34.1.0.0 (subnet 1 on network 34), the next stop is the node at address 54.34.23.12.”

Destination address	Next hop
34.1.0.0	54.34.23.12
78.2.0.0	54.34.23.12
147.9.5.0	.
17.12.0.0	.
.	54.32.12.10
.	54.32.12.10
.	.
.	.

Figure 18-7 IP Routing Table

IP routing specifies that IP datagrams travel through internetworks one hop at a time. The entire route is not known at the outset of the journey. Instead, at each stop, the next destination is calculated by matching the destination address within the datagram with an entry in the current node’s routing table. Each node’s involvement in the routing process consists only of forwarding packets based on internal information, without regard for the measure of success that may or may not have been achieved at reaching the final destination. In other words, IP does not provide for error reporting back to the source when routing anomalies occur. This task is left to another Internet protocol: the *Internet Control Message Protocol (ICMP)*.

ICMP

ICMP performs a number of tasks within an IP internetwork. In addition to the principal reason it was created (for reporting routing failures back to the source), ICMP also provides a method for testing node reachability across an internetwork (the ICMP *Echo* and *Reply* messages), a method for stimulating more efficient routing (the ICMP *Redirect* message), a method for informing sources that a datagram has exceeded its allocated time to exist within the internetwork (the ICMP *Time Exceeded* message) and other helpful messages. A more recent addition to ICMP provides a way for new nodes to discover the subnet mask currently used in an internetwork. All in all, ICMP is an integral part of any IP implementation, particularly those that run in routers.

Discussion of specific IP routing protocols occurs in other chapters of this book. For example, RIP is discussed in Chapter 23, "RIP," IGRP is discussed in Chapter 24, "IGRP," OSPF is discussed in Chapter 25, "OSPF," EGP is discussed in Chapter 26, "EGP," and BGP is discussed in Chapter 27, "BGP." IS-IS is also an official IP routing protocol and is discussed in Chapter 28, "OSI Routing."

Transport Layer

The Internet transport layer is implemented by TCP and the *User Datagram Protocol (UDP)*. TCP provides connection-oriented data transport, while UDP operation is connectionless.

Transmission Control Protocol (TCP)

TCP provides full-duplex, acknowledged, and flow-controlled service to upper-layer protocols. It moves data in a continuous, unstructured byte stream where bytes are identified by sequence numbers. TCP can also support numerous simultaneous upper-layer conversations. The TCP packet format is shown in Figure 18-8.

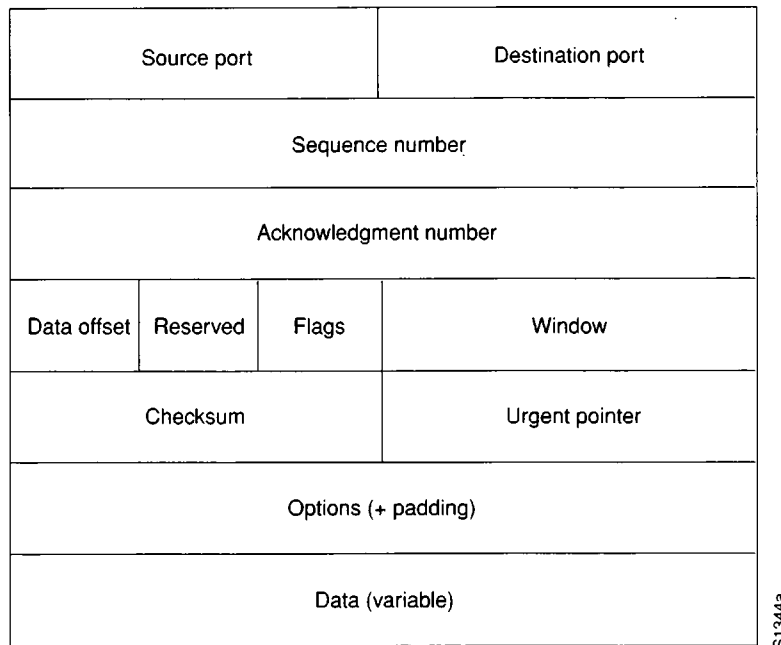


Figure 18-8 TCP Packet Format

The *source port* field identifies the point at which a particular source upper-layer process receives TCP services; the *destination port* field identifies the destination upper-layer process port to TCP services.

The *sequence number* field usually specifies the number assigned to the first byte of data in the current message. Under certain circumstances, it can also be used to identify an initial sequence number to be used in the upcoming transmission.

The *acknowledgment number* field contains the sequence number of the next byte of data the sender of the packet expects to receive.

The *data offset* field indicates the number of 32-bit words in the TCP header.

The *reserved* field is reserved for future use by protocol designers.

The *flags* field carries a variety of control information.

The *window* field specifies the size of the sender's receive window (buffer space available for incoming data).

The *checksum* field indicates whether the header was damaged in transit.

The *urgent pointer* field points to the first urgent data byte in the packet.

The *options* field specifies various TCP options.

User Datagram Protocol (UDP)

UDP is a much simpler protocol than TCP and is useful in situations where TCP's powerful reliability mechanisms are not necessary. The UDP header has only four fields: *source port*, *destination port*, *length*, and *UDP checksum*. The source and destination port fields serve the same functions as they do in the TCP header. The length field specifies the length of the UDP header and data, and the checksum field allows packet integrity checking. The UDP checksum is optional.

Upper-Layer Protocols

The Internet protocol suite includes many upper-layer protocols representing a wide variety of applications, including network management, file transfer, distributed file services, terminal emulation, and electronic mail. Figure 18-9 maps the best-known Internet upper-layer protocols to the applications they support.

Application	Protocols
File transfer	FTP
Terminal emulation	Telnet
Electronic mail	SMTP
Network management	SNMP
Distributed file services	NFS, XDR, RPC X Windows

S1345a

Figure 18-9 Internet Protocol/Application Mapping

The *File Transfer Protocol (FTP)* provides a way to move files between computer systems. *Telnet* allows virtual terminal emulation. The *Simple Network Management Protocol (SNMP)* is a network management protocol used for reporting anomalous network conditions and setting network threshold values. *X Windows* is a popular protocol that permits intelligent terminals to communicate with remote computers as if they were directly attached monitors. *Network File System (NFS)*, *External Data Representation (XDR)*, and *Remote Procedure Call (RPC)* combine to allow transparent access to remote network resources. The *Simple Mail Transfer Protocol (SMTP)* provides an electronic mail transport mechanism. These and other network applications use the services of TCP/IP and other lower-layer Internet protocols to provide users with basic network services.

Chapter 19

NetWare Protocols

19

Background

NetWare is a *network operating system* (NOS) and related support services environment created by Novell, Inc. and introduced to the market in the early 1980s. In those days, networks were small and predominantly homogeneous, local area network (LAN) workgroup communication was new, and the idea of a personal computer (PC) was just becoming popular.

Much of NetWare's networking technology was derived from *Xerox Network Systems* (XNS), a networking system created by Xerox Corporation in the late 1970s. For more information on XNS, see Chapter 22, "XNS."

By the early 1990s, NetWare's NOS market share had risen to between 50 and 75 percent (depending on the market research group performing the study). With over 500,000 NetWare networks installed worldwide and an accelerating movement to connect networks to other networks, NetWare and its supporting protocols often coexist on the same physical channel with many other popular protocols, including TCP/IP, DECnet, and AppleTalk.

Technology Basics

As a NOS environment, NetWare specifies the upper five layers of the OSI reference model. It provides file and printer sharing, support for various applications such as electronic mail transfer and database access, and other services. Like other NOSs such as the *Network File System* (NFS) from Sun Microsystems, Inc. and *LAN Manager* from Microsoft Corporation, NetWare is based on a *client-server architecture*. In such architectures, *clients* (sometimes called workstations) request certain services such as file and printer access from *servers*.

Originally, NetWare clients were small PCs, while servers were slightly more powerful PCs. As NetWare became more popular, it was ported to other computing platforms. Currently, NetWare clients and servers may be represented by virtually any kind of computer system, from PCs to mainframes.

A primary characteristic of the client-server system is that remote access is transparent to the user. This is accomplished through *remote procedure calls*, a process where a local computer program running on a client sends a procedure call to a remote server. The server executes the remote procedure call and returns the requested information to the local computer client.

Figure 19-1 illustrates a simplified view of NetWare's best-known protocols and their relationship to the OSI reference model. With appropriate drivers, NetWare can run on any media-access protocol. The figure lists those media-access protocols currently supported with NetWare drivers.

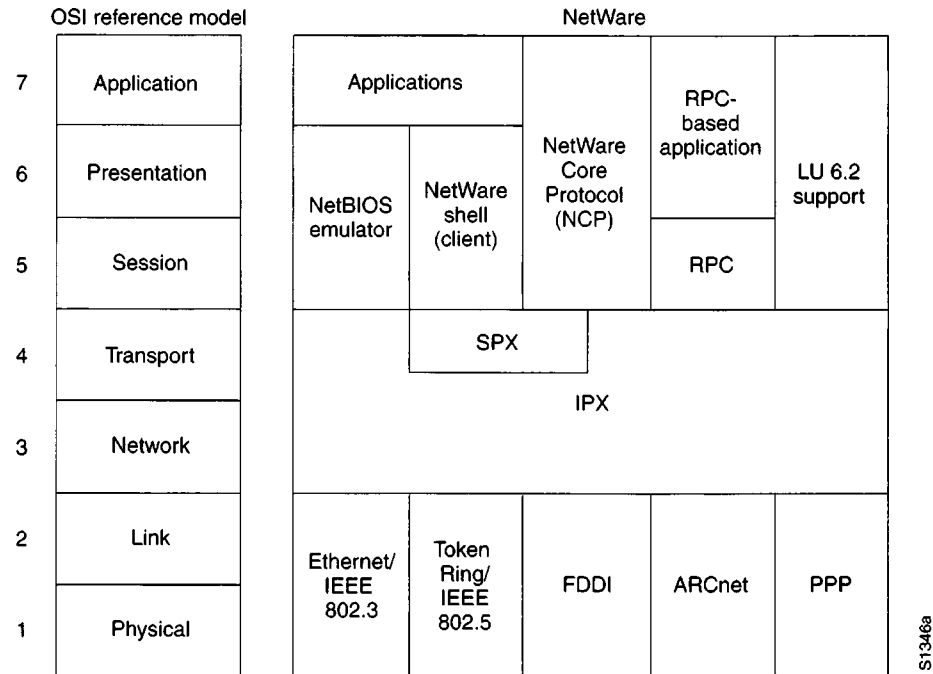


Figure 19-1 NetWare and the OSI Reference Model

Media Access

NetWare runs on *Ethernet/IEEE 802.3*, *Token Ring/IEEE 802.5*, *Fiber Distributed Data Interface (FDDI)*, and *ARCnet*. For information on Ethernet/IEEE 802.3, see Chapter 5, "Ethernet/IEEE 802.3." For information on Token Ring/IEEE 802.5, see Chapter 6, "Token Ring/IEEE 802.5." For information on FDDI, see Chapter 7, "FDDI." NetWare also works over synchronous wide area network (WAN) links using the *Point-to-Point Protocol (PPP)*. PPP is discussed in more detail in Chapter 10, "PPP."

ARCnet is a simple network system that supports all three primary media (*twisted pair*, *coaxial cable*, and *fiber-optic cable*) and two topologies (*bus* and *star*). It was developed by Datapoint Corporation and introduced in 1977. Although ARCnet has not attained the popularity enjoyed by Ethernet and Token Ring, its low cost and flexibility have resulted in many loyal supporters.

Network Layer

Internet Packet Exchange (IPX) is Novell's original network-layer protocol. When a device to be communicated with is located on a different network, IPX routes the information through any intermediate networks that might be present to the destination. Figure 19-2 shows the IPX packet format.

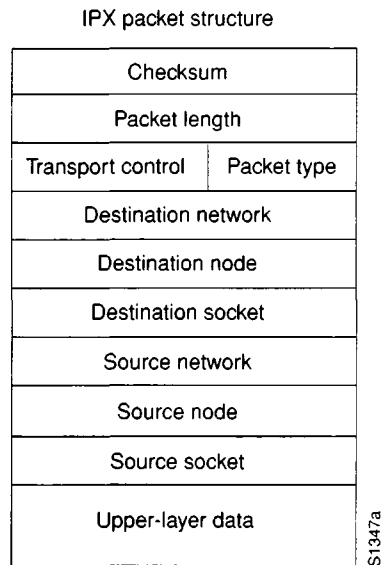


Figure 19-2 IPX Packet Format

The IPX packet begins with a 16-bit *checksum* field that is set to ones.

A 16-bit *length* field specifies the length, in bytes, of the complete IPX datagram. IPX packets can be any length up to the media maximum transfer unit (MTU) size. There is no packet fragmentation.

Following the length field is an 8-bit *transport control* field, which indicates the number of routers the packet has passed through. When the value of this field reaches 15, the packet is discarded under the assumption that a routing loop might be occurring.

The 8-bit *packet type* field specifies the upper-layer protocol to receive the packet's information. Two common values for this field are 5, which specifies *Sequenced Packet Exchange (SPX)*, and 17, which specifies the *NetWare Core Protocol (NCP)*.

Destination address information occupies the next three fields. These fields specify the destination network, host, and socket (process).

Following these are three *source address* fields, specifying the source network, host, and socket.

The *data* field follows the destination and source fields. It contains information for upper-layer processes.

Although IPX was derived from XNS, it has several unique features. From the standpoint of routing, the encapsulation mechanisms of these two protocols is the most important difference. Encapsulation is the process of packaging upper-layer protocol information and data into a *frame*. Frames are logical groups of information very much like words in a telephone conversation. XNS uses standard Ethernet encapsulation, whereas IPX packets are encapsulated in Ethernet Version 2.0 or IEEE 802.3 frames without the IEEE 802.2 information that typically accompanies these frames. Figure 19-3 illustrates Ethernet, standard IEEE 802.3, and IPX encapsulation.

Note: NetWare 4.0 supports encapsulation of IPX packets in IEEE 802.3 frames.

Ethernet	Standard 802.3	IPX
Destination address	Destination address	Destination address
Source address	Source address	Source address
Type	Length	Length
Upper-layer data	802.2 Header	IPX data
	802.2 Data	
CRC	CRC	CRC

S1348a

Figure 19-3 Ethernet, IEEE 802.3, and IPX Encapsulation Formats

To route packets in an internetwork, IPX uses a dynamic routing protocol called the *Routing Information Protocol (RIP)*. Like XNS, RIP derived from work done at Xerox for the XNS protocol family. Today, RIP is the most commonly used *interior gateway protocol (IGP)* in the Internet community, an international network environment providing connectivity to virtually every university and research institute and many commercial organizations in the United States, and many foreign organizations as well. For more detailed information on RIP, see Chapter 23, “RIP.”

In addition to the difference in encapsulation mechanisms, Novell also added a protocol called the *Service Advertisement Protocol (SAP)* to its IPX protocol family. SAP allows nodes that provide services (such as file servers and print servers) to advertise their addresses and the services they provide.

Novell also supports IBM’s LU 6.2 *network addressable unit (NAU)*. LU 6.2 allows peer-to-peer connectivity across IBM communication environments. Using NetWare’s LU 6.2 capability, NetWare nodes can exchange information across an IBM network. NetWare packets are encapsulated within LU 6.2 packets for transit across the IBM network.

Transport Layer

Sequenced Packet Exchange (SPX) is the most commonly used NetWare transport protocol. Novell derived this protocol from XNS's *Sequenced Packet Protocol (SPP)*. As with the *Transmission Control Protocol (TCP)* and many other transport protocols, SPX is a reliable, connection-oriented protocol that supplements the datagram service provided by Layer 3 protocols.

Novell also offers *Internet Protocol (IP)* support in the form of *User Datagram Protocol (UDP)*/IP encapsulation of other Novell packets, such as SPX/IPX packets. IPX datagrams are encapsulated inside UDP/IP headers for transport across an IP-based internetwork. For more information on UDP and the Internet protocols in general, see Chapter 18, "Internet Protocols."

Upper-Layer Protocols

NetWare supports a wide variety of upper-layer protocols, but several are somewhat more popular than others. The *NetWare shell* runs in clients (often called workstations in the NetWare community) and intercepts application I/O calls to determine whether they require network access for satisfaction. If so, the NetWare shell packages the requests and sends them to lower-layer software for processing and network transmission. If not, they are simply passed to local I/O resources. Client applications are unaware of any network access required for completion of application calls. *NetWare Remote Procedure Call (NetWare RPC)* is another more general redirection mechanism supported by Novell.

The *NetWare Core Protocol (NCP)* are a series of server routines designed to satisfy application requests coming from, for example, the NetWare shell. Services provided by NCP include file access, printer access, name management, accounting, security, and file synchronization.

NetWare also supports the *Network Basic I/O System (NetBIOS)* session-layer interface specification from IBM and Microsoft. NetWare's NetBIOS emulation software allows programs written to the industry-standard NetBIOS interface to run within the NetWare system.

NetWare application-layer services include *NetWare Message Handling Service (NetWare MHS)*, *Btrieve*, *NetWare Loadable Modules (NLMs)*, and various IBM connectivity features. NetWare MHS is a message delivery system that provides electronic mail transport. *Btrieve* is Novell's implementation of the binary tree (btree) database access mechanism. NLMs are implemented as add-on modules that attach into the NetWare system. NLMs for alternate protocol stacks, communication services, database services, and many other services are currently available from Novell and third parties.

Chapter 20

OSI Protocols

20

Background

In the early days of intercomputer communication, networking software was created in a haphazard, ad hoc fashion. When networks grew sufficiently popular, some recognized the need to standardize the by-products of network software and hardware development. Standardization, it was believed, would allow vendors to create hardware/software systems that could communicate with one another, even if the underlying architectures were dissimilar. With this goal, the International Organization for Standardization (ISO) began development of the *Open Systems Interconnection (OSI)* reference model. The OSI reference model was completed and released in 1984.

Today, the OSI reference model (discussed in detail in Chapter 1, “Introduction to Internetworking”) is the world’s most prominent networking architecture model. It is also the most popular tool for learning about networks. The OSI protocols, on the other hand, have had a long gestation period. While OSI implementations are not unheard of, the OSI protocols have not yet attained the popularity of many proprietary (for example, DECnet and AppleTalk) and de facto (for example, the Internet protocols) standards.

Technology Basics

The world of OSI networking has a unique terminology.

- *End system (ES)* refers to any nonrouting network device.
- *Intermediate system (IS)* refers to a router.
- *Area* is a group of contiguous networks and attached hosts that are specified to be an area by a network administrator or similar person.
- *Domain* is a collection of connected areas. Routing domains provide full connectivity to all end systems within them.

Media Access

Like several other modern 7-layer protocol stacks, the OSI stack includes many of today's popular media-access protocols. This allows other protocol stacks to exist alongside OSI on the same media. OSI includes *IEEE 802.2*, *IEEE 802.3*, *IEEE 802.5*, *FDDI*, *X.21*, *V.35*, *X.25*, and others. Most of these OSI media-access protocols are discussed elsewhere in this publication.

Network Layer

OSI offers both a connectionless and a connection-oriented network layer service. The connectionless service is described in ISO 8473 (usually referred to as *Connectionless Network Protocol* or *CLNP*). The connection-oriented service (sometimes called *Connection-Oriented Network Service*, or *CONS*) is described in ISO 8208 (*X.25 Packet-Level Protocol*, sometimes referred to as *Connection-Mode Network Protocol*, or *CMNP*) and ISO 8878 (which describes how to use ISO 8208 to provide OSI connection-oriented service). An additional document, ISO 8881, describes how to run the X.25 Packet-Level Protocol over IEEE 802 LANs. OSI also specifies several routing protocols, which are discussed in Chapter 28, "OSI Routing." X.25 is discussed in Chapter 13, "X.25."

In addition to the previously mentioned protocol and service specifications, other relevant OSI network-layer documents include:

- ISO 8648—This document is usually referred to as the *internal organization of the network layer (IONL)*. It describes how the network layer can be broken into three separate and distinct sublayers to allow support for different subnetwork types.
- ISO 8348—This document is usually called the *network service definition*. It describes the connection-oriented and connectionless services provided by the OSI network layer. Network layer addressing is also defined in this document. The connectionless-mode service definition and the addressing definition were previously published as separate addenda to ISO 8348; however, the 1993 version of ISO 8348 folds all of the addenda into a single document.
- ISO TR 9575—This document describes the framework, concepts, and terminology used in OSI routing protocols.
- ISO TR 9577—This document describes how to discriminate between multiple network layer protocols running on the same medium. Such discrimination is necessary because, unlike other protocols, OSI network layer protocols are not discriminated through a protocol ID or similar field at the data link layer.

Connectionless Service

As its name suggests, CLNP is a connectionless datagram protocol used to carry data and error indications. Functionally, it is quite similar to the *Internet Protocol (IP)* described in Chapter 18, “Internet Protocols.” It has no facilities for error detection or correction, relying on the transport layer to provide these services as appropriate. It has only one phase, called *data transfer*. Each invocation of a service primitive is independent of all other invocations, requiring all address information to be completely contained within the service primitive.

While CLNP defines the actual protocol performing typical network-layer functions, *Connectionless Network Service (CLNS)* describes a service provided to the transport layer in which a request to transfer data receives “best effort” delivery. Such delivery does not guarantee that data will not be lost, corrupted, misordered, or duplicated. Connectionless service assumes that the transport layer will correct such problems as necessary. CLNS does not provide any form of connection information or state and connection setup is not performed. Because CLNS provides transport layers with the service interface to CLNP, CLNS and CLNP are often discussed together.

Connection-Oriented Service

OSI connection-oriented network service is specified by ISO 8208 and ISO 8878. OSI uses the X.25 Packet-Level Protocol for connection-oriented data movement and error indications. Six services are provided to transport-layer entities (one for connection establishment, one for connection release, and four for data transfer). Services are invoked by some combination of four *primitives*: *request*, *indication*, *response*, and *confirmation*. The four primitives interact as shown in Figure 20-1.

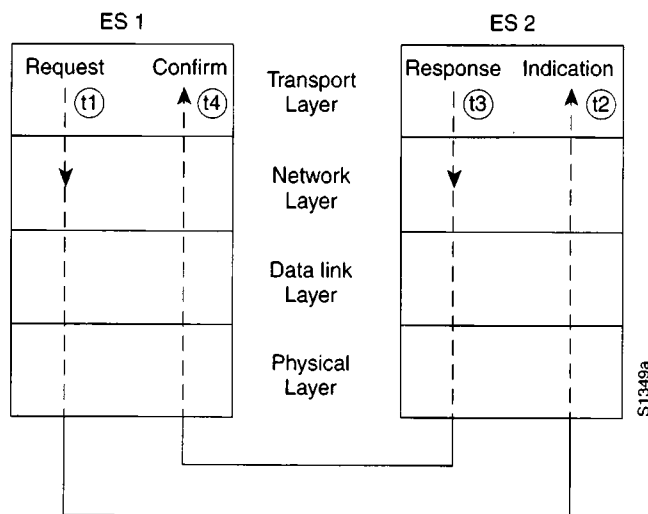


Figure 20-1 OSI Primitives

At time t1, ES 1's transport layer sends a request primitive to ES 1's network layer. This request is placed onto ES 1's subnetwork by lower-layer subnetwork protocols, and is eventually received by ES 2, which sends the information up to the network layer. At time t2, ES 2's network layer sends an indication primitive to its transport layer. After required upper-layer processing for this packet is complete, ES 2 initiates a response to ES 1 by using a response primitive sent from the transport layer to the network layer. The response, which occurs at time t3, travels back to ES 1, which sends the information up to the network layer where a confirm primitive is generated and sent to the transport layer at time t4.

Addressing

The OSI network service is provided to the transport layer through a conceptual point on the network/transport layer boundary known as a *network service access point (NSAP)*. There is one NSAP per transport entity.

Each NSAP can be individually addressed in the global OSI internetwork through its *NSAP address* (often colloquially and imprecisely known as an NSAP). Thus, an OSI end system will typically have multiple NSAP addresses. These addresses will usually differ only in the last byte, known as the *n-selector*.

It is also useful to address a system's network layer without being associated with a specific transport entity, for instance, when a system participates in routing protocols or when addressing an intermediate system (router). Such addressing is done via a special network address known as a *network entity title (NET)*. A NET is structurally identical to an NSAP address, but uses the special n-selector value "00." Most end systems and intermediate systems have only a single NET, unlike IP routers which usually have one address per interface. However, an intermediate system that is participating in multiple areas or domains can choose to have multiple NETs.

NETs and NSAP addresses are hierarchical addresses. Addressing hierarchies facilitate both administration (by allowing multiple levels of address administration) and routing (by encoding network topology information). An NSAP address is first separated into two parts: the *initial domain part (IDP)* and the *domain specific part (DSP)*. The IDP is then further divided into the *authority and format identifier (AFI)* and the *initial domain identifier (IDI)*.

The AFI provides information about the structure and content of the IDI and DSP fields, including whether the IDI is of variable length and whether the DSP uses decimal or binary notation. The IDI specifies an entity that can assign values to the DSP portion of the address.

The DSP is then further subdivided by the authority responsible for its administration. Typically, another administrative authority identifier follows, allowing address administration to be further delegated to sub-authorities. Following that comes information used for routing, such as the routing domain, the area within the routing domain, the station ID within the area, and the selector within the station. Figure 20-2 illustrates the OSI address format.

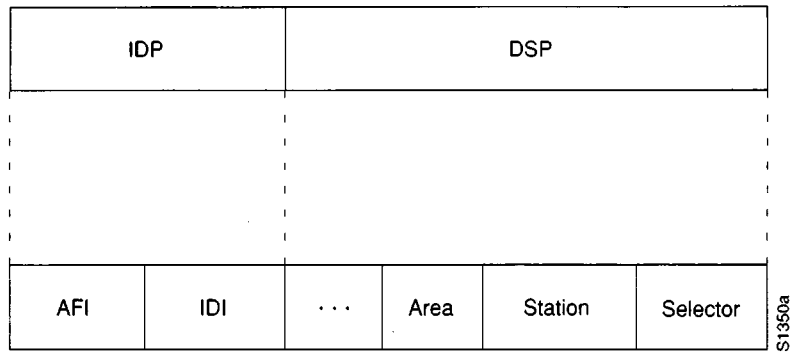


Figure 20-2 OSI Address Format

Transport Layer

As with the OSI network layer, both connectionless and connection-oriented transport services are offered. There are actually five connection-oriented OSI transport protocols: *TP0*, *TP1*, *TP2*, *TP3*, and *TP4*. All but *TP4* work only with OSI's connection-oriented network service. *TP4* works with both connection-oriented and connectionless services.

TP0 is the simplest OSI connection-oriented transport protocol. Of the classical transport layer functions, it performs only segmentation and reassembly. This means that *TP0* will note the smallest maximum size *protocol data unit (PDU)* supported by the underlying subnetworks, and will break the transport packet into smaller pieces that are not too big for network transmission.

In addition to segmentation and reassembly, *TP1* offers basic error recovery. It numbers all PDUs and resends those that are unacknowledged. *TP1* can also reinitiate connections when excessive unacknowledged PDUs occur.

TP2 can multiplex and demultiplex data streams over a single virtual circuit. This capability makes *TP2* particularly useful over public data networks (PDNs), where each virtual circuit incurs a separate charge. Like *TP0* and *TP1*, *TP2* also segments and reassembles PDUs.

TP3 combines the features of *TP1* and *TP2*.

TP4 is the most popular OSI transport protocol. *TP4* is similar to the Internet protocol suite's *Transmission Control Protocol (TCP)* and, in fact, was based on *TCP*. In addition to *TP3*'s features, *TP4* provides reliable transport service. It assumes a network in which problems are not detected.

Upper-Layer Protocols

Principle OSI upper-layer protocols are shown in Figure 20-3.

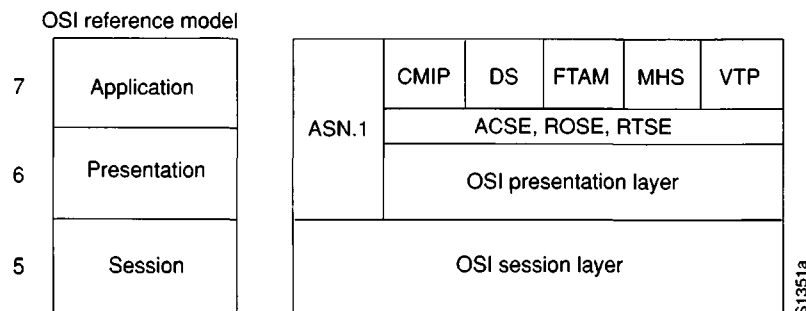


Figure 20-3 Principle OSI Upper-Layer Protocols

Session Layer

The OSI session-layer protocol turns the data streams provided by the lower four layers into sessions by implementing various control mechanisms. These mechanisms include accounting, conversation control (that is, determining who can talk when), and session parameter negotiation.

Session conversation control is implemented by use of a *token*, the possession of which provides the right to communicate. The token can be requested, and ESs can be given priorities that provide for unequal token use.

Presentation Layer

The OSI presentation layer is typically just a pass-through protocol for information from adjacent layers. Although many people believe that *Abstract Syntax Notation 1 (ASN.1)* is OSI's presentation-layer protocol, ASN.1 is used for expressing data formats in a machine-independent format. This allows communication between applications on diverse computer systems (ESs) in a manner transparent to the applications.

Application Layer

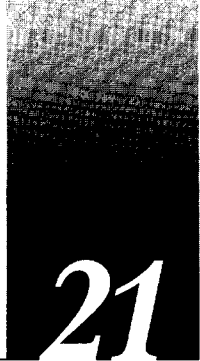
The OSI application layer includes actual applications as well as *application service elements (ASEs)*. ASEs allows easy communication from applications to lower layers. The three most important ASEs are *Association Control Service Element (ACSE)*, *Remote Operations Service Element (ROSE)*, and *Reliable Transfer Service Element (RTSE)*. ACSE associates application names with one another in preparation for application-to-application communication. ROSE implements a generic request-reply mechanism that permits remote operations in a manner similar to that of *remote procedure calls (RPCs)*. RTSE aids reliable delivery by making session-layer constructs easy to use.

Five OSI applications receive the most attention:

- *Common Management Information Protocol (CMIP)*—OSI's network management protocol. Like SNMP (see Chapter 32, "SNMP," for more information) and NetView (see Chapter 33, "IBM Network Management," for more information), it allows exchange of management information between ESs and management stations (which are also ESs).
- *Directory Services (DS)*—Derived from the Consultative Committee for International Telegraph and Telephone (CCITT) X.500 specification, this service provides distributed database capabilities useful for upper-layer node identification and addressing.
- *File Transfer, Access, and Management (FTAM)*—OSI's file transfer service. In addition to classical file transfer, for which FTAM provides numerous options, FTAM also offers distributed file access facilities in the spirit of NetWare from Novell, Inc. or Network File System (NFS) from Sun Microsystems, Inc.
- *Message Handling Systems (MHS)*—Provides an underlying transport mechanism for electronic messaging applications and other applications desiring store-and-forward services. Although they accomplish similar purposes, MHS is not to be confused with Novell's NetWare MHS (see Chapter 19, "NetWare Protocols," for more information).
- *Virtual Terminal Protocol (VTP)*—Provides terminal emulation. In other words, it allows a computer system to appear to a remote ES as if it were a directly attached terminal. With VTP, users can (for example) run remote jobs on mainframes.

Chapter 21

Banyan VINES



Background

Banyan Virtual Network System (VINES) implements a distributed network system based on a proprietary protocol family derived from Xerox's *Xerox Network Systems (XNS)* protocols (see Chapter 22, "XNS"). A distributed system environment permits user-transparent exchange of information between clients (user computers) and servers (specially designated computers that provide services such as file and print service). Along with Novell's *NetWare*, IBM's *LAN Server*, and Microsoft's *LAN Manager*, VINES is one of the best-known distributed system environments for microcomputer-based networks.

Technology Basics

The VINES protocol stack is pictured in Figure 21-1.

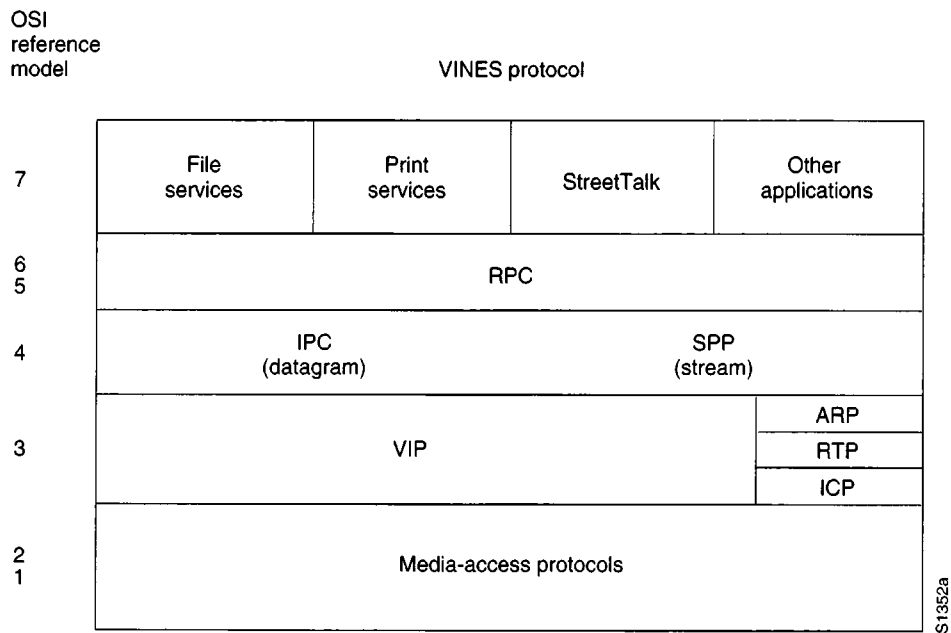


Figure 21-1 VINES Protocol Stack

Media Access

The lower two layers of the VINES stack are implemented with a variety of well-known media-access mechanisms, including *High-level Data Link Control (HDLC)* (see Chapter 12, “SDLC and Derivatives”), *X.25* (see Chapter 13, “X.25”), *Ethernet* (see Chapter 5, “Ethernet/IEEE 802.3”), and *Token Ring* (see Chapter 6, “Token Ring/IEEE 802.5”).

Network Layer

VINES uses the *VINES Internetwork Protocol (VIP)* to perform Layer 3 activities (including internetwork routing). VINES also supports its own *Address Resolution Protocol (ARP)*, its own version of the *Routing Information Protocol (RIP)* called the *Routing Update Protocol (RTP)*, and the *Internet Control Protocol (ICP)*, which provides exception handling and special routing cost information. ICP, RTP, and ARP packets are encapsulated in a VIP header.

VINES Internetwork Protocol (VIP)

VINES network-layer addresses are 48-bit entities subdivided into network (32 bits) and subnetwork (16 bits) portions. The network number is better described as a server number, since it is derived directly from the server’s *key* (a hardware module that identifies a unique number and the software options for that server). The subnetwork portion of a VINES address is better described as a host number, since it is used to identify hosts on VINES networks. Figure 21-2 illustrates the VINES address format.

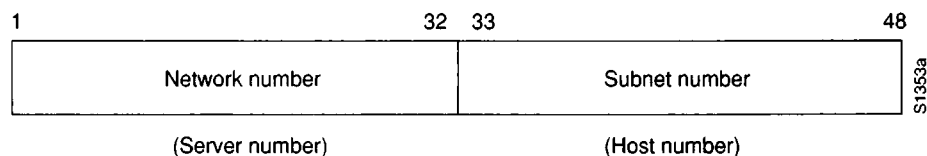


Figure 21-2 VINES Address Format

The network number identifies a VINES logical network, which is represented as a two-level tree with the root at a *service node*. Service nodes, which are usually servers, provide address resolution and routing services to *clients*, which represent the leaves of the tree. The service node assigns VIP addresses to clients.

When a client is powered on, it broadcasts a request for servers. All servers that hear the request respond. The client chooses the first response and requests a subnetwork (host) address from that server. The server responds with an address consisting of its own network address (derived from its key), concatenated with a subnetwork (host) address of its own choosing. Client subnetwork addresses are typically assigned sequentially, starting with 8001H. Server subnetwork addresses are always 1. The VINES address selection process is shown in Figure 21-3.

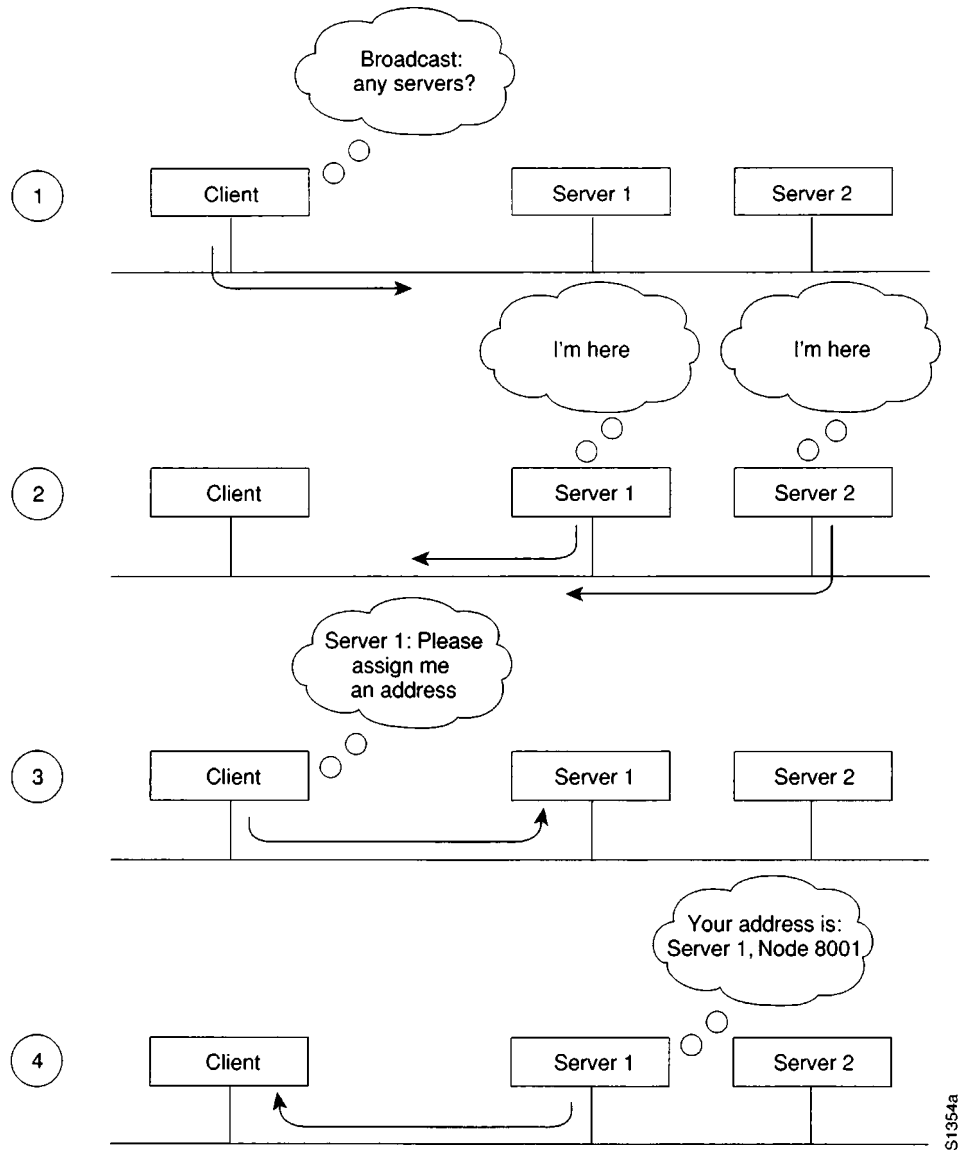


Figure 21-3 VINES Address Selection Process

Dynamic address assignment is not unique in the industry (AppleTalk also uses this process), but it is certainly not as common as static address assignment. Since addresses are chosen exclusively by a particular server (whose address is unique as a result of the uniqueness of the hardware key), there is very little chance of a duplicate address (a potentially devastating problem on *Internet Protocol (IP)* and other networks).

In the VINES network scheme, all servers with multiple interfaces are essentially routers. Clients always choose their own server as a first-hop router, even if another server on the same cable provides a better route to the ultimate destination. Clients can learn about other routers by receiving redirect messages from their own server. Since clients rely on their servers for first-hop routing, VINES servers maintain routing tables to help them find remote nodes.

VINES routing tables consist of host/cost pairs, where host corresponds to a network node that can be reached and cost corresponds to a delay, expressed in milliseconds, to get to that node. RTP helps VINES servers find neighboring clients, servers, and routers.

Periodically, all clients advertise both their network-layer and their MAC-layer addresses with the equivalent of a *hello* packet. Hello packets indicate that the client is still operating and network-ready. The servers themselves send routing updates to other servers periodically. Routing updates alert other routers to changes in node addresses and network topology.

When a VINES server receives a packet, it checks to see if the packet is destined for another server, or a broadcast. If the current server is the destination, the server handles the request appropriately. If another server is the destination, the current server either forwards the packet directly (if the server is a neighbor) or routes it to the next server/router in line. If the packet is a broadcast, the current server checks to see if the packet came from the least cost path. If not, the packet is discarded. If so, the packet is forwarded on all interfaces except the one on which the packet was received. This approach helps diminish the number of broadcast storms, a common problem in other network environments. The VINES routing algorithm is shown in Figure 21-4.

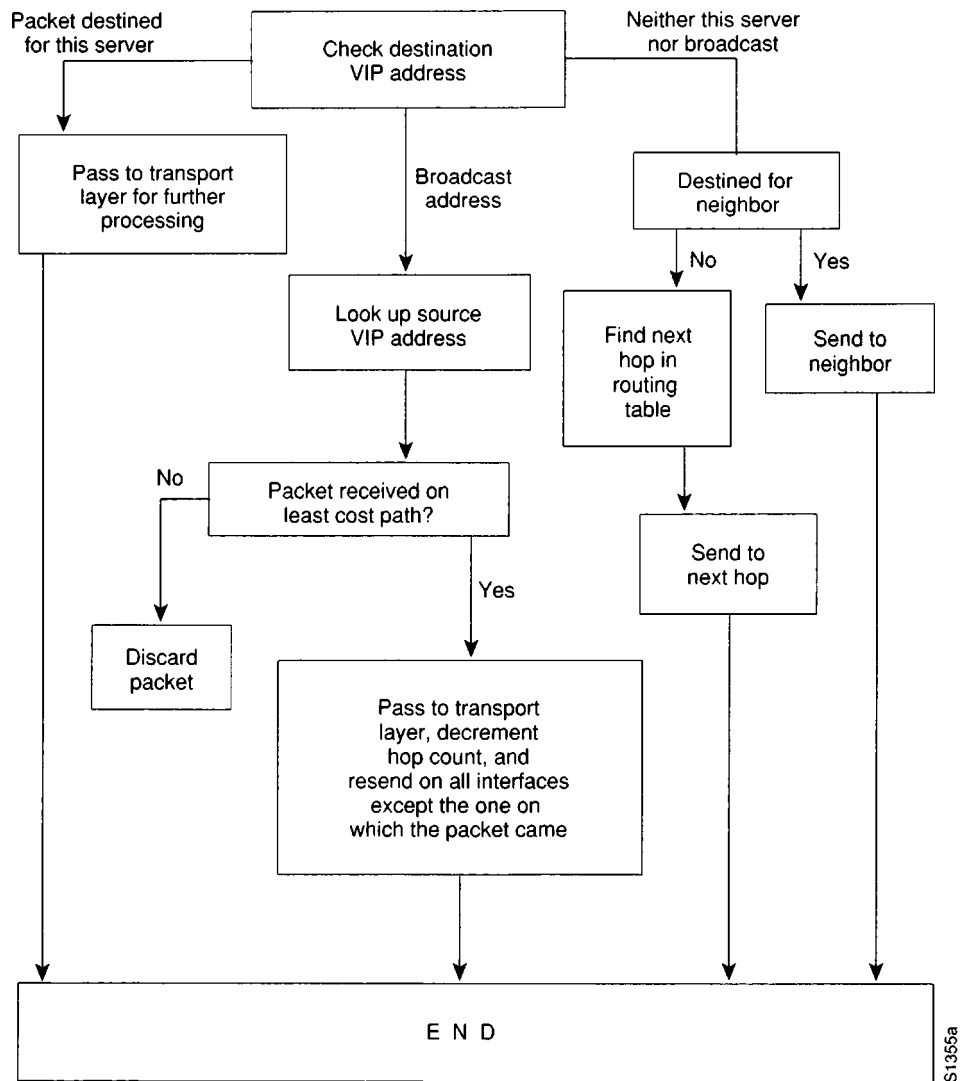


Figure 21-4 VINES Routing Algorithm

The VIP packet format is shown in Figure 21-5.

Field length, in bytes	2	2	1	1	4	2	4	2	Variable
Check-sum	Packet length	Transport control	Protocol type	Dest. network number	Dest. subnetwork number	Source network number	Source subnetwork number	Data	

Figure 21-5 VIP Packet Format

The VIP packet begins with a *checksum* field, used to detect packet corruption.

Following the checksum is the *packet length* field, which indicates the length of the entire VIP packet.

The next field is the *transport control* field, which consists of several subfields. If the packet is a broadcast packet, two subfields are provided: the *class* subfield (bits 1 through 3) and the *hop-count* subfield (bits 4 through 7). If the packet is not a broadcast packet, four subfields are provided: the *error* subfield, the *metric* subfield, the *redirect* subfield, and the *hop count* subfield. The class subfield specifies the type of node that should receive the broadcast. For this purpose, nodes are broken into various categories having to do with the type of node and the type of link the node is on. By specifying the type of nodes to receive broadcasts, the class subfield reduces the disruption caused by broadcasts. The hop count subfield represents the number of hops (router traversals) the packet has been through. The error subfield specifies whether the ICP protocol should send an exception notification packet to the packet's source if a packet turns out to be unroutable. The metric subfield is set to one by a transport entity when it needs to learn the routing cost of moving packets between a service node and a neighbor. The redirect subfield specifies whether the router should generate a redirect (when appropriate).

Next, the *protocol type* field indicates the network- or transport-layer protocol for which the metric or exception notification packet is destined.

Following the protocol type field are the VIP address fields. The *destination network number* and *destination subnetwork number* fields are followed by the *source network number* and *subnetwork number* fields.

Routing Update Protocol (RTP)

RTP distributes network topology information. Routing update packets are broadcast periodically by both client and service nodes. These packets inform neighbors of a node's existence and also indicate whether the node is a client or a service node. Service nodes also include, in each routing update packet, a list of all known networks and the cost factors associated with reaching those networks.

Two routing tables are maintained: a *table of all known networks* and a *table of neighbors*. For service nodes, the table of all known networks contains an entry for each known network except the service node's own network. Each entry contains a network number, a routing metric, and a pointer to the entry for the next hop to the network in the table of neighbors. The table of neighbors contains an entry for each neighbor service node and client node. Entries include a network number, a subnetwork number, the media-access protocol (for example, Ethernet) used to reach that node, a LAN address (if the medium connecting the neighbor is a LAN), and a neighbor metric.

RTP specifies four packet types:

- Routing update packets—Issued periodically to notify neighbors of an entity's existence.
- Routing request packets—Exchanged by entities when they need to learn the network's topology quickly.
- Routing response packets—Contain topological information and are used by service nodes to respond to routing request packets.
- Routing redirect packets—Provide better path information to nodes using inefficient paths.

RTP packets have a four-byte header consisting of a one-byte *operation type* field, a one-byte *node type* field, a one-byte *controller type* field, and a one-byte *machine type* field. The operation type field indicates the packet type. The node type field indicates whether the packet came from a service node or a nonservice node. The controller type field indicates whether the controller in the node transmitting the RTP packet has a multibuffer controller. This field is used to help regulate data flow between network nodes. Finally, the machine type field indicates whether the processor in the RTP sender is fast or slow. Like the controller type field, the machine type field is also used for pacing.

Address Resolution Protocol (ARP)

ARP protocol entities are classified as either *address resolution clients* or *address resolution services*. Address resolution clients are usually implemented in client nodes, whereas address resolution services are typically provided by service nodes.

ARP packets have an 8-byte header consisting of a 2-byte *packet type*, a 4-byte *network number*, and a 2-byte *subnetwork number*. There are four packet types: a *query request*, which is a request for an ARP service, a *service response*, which is a response to a query request, an *assignment request*, which is sent to an ARP service to request a VINES internetwork address, and an *assignment response*, which is sent by the ARP service as a response to the assignment request. The network number and subnet number fields only have meaning in an assignment response packet.

ARP clients and services implement the following algorithm when a client starts up. First, the client broadcasts query request packets. Then, each service that is a neighbor of the client responds with a service response packet. The client then issues an assignment request packet to the first service that responded to its query request packet. The service responds with an assignment response packet containing the assigned internet address.

Internet Control Protocol (ICP)

ICP defines *exception notification* and *metric notification* packets. Exception notification packets provide information about network-layer exceptions; metric notification packets contain information about the final transmission used to reach a client node.

Exception notifications are sent when a VIP packet cannot be routed properly and the error subfield in the VIP header's transport control field is enabled. These packets also contain a field identifying the particular exception by its error code.

ICP entities in service nodes generate metric notification messages when the metric subfield in the VIP header's transport control field is enabled and the destination address in the service node's packet specifies one of the service node's neighbors.

Transport Layer

VINES provides three transport-layer services:

- *Unreliable datagram service*—Sends packets that are routed on a best-effort basis but not acknowledged at the destination.
- *Reliable message service*—A virtual-circuit service that provides reliable sequenced and acknowledged delivery of messages between network nodes. A reliable message may be transmitted in a maximum of four VIP packets.
- *Data stream service*—Supports the controlled flow of data between two processes. The data stream service is an acknowledged virtual circuit service that supports the transmission of unlimited-size messages.

Upper-Layer Protocols

As a distributed network, VINES uses the *remote procedure call (RPC)* model for communication between clients and servers. RPC is the foundation of distributed service environments. The *NetRPC* protocol (Layers 5 and 6) provides a high-level programming language that allows access to remote services in a manner transparent to both the user and the application.

At Layer seven, VINES offers file-service and print-service applications, as well as the *StreetTalk* name/directory service protocol. One of VINES' trademark protocols, *StreetTalk* provides a globally consistent name service for an entire internetwork.

VINES also provides an integrated applications development environment under several operating systems, including DOS and UNIX. This development environment allows third parties to develop both clients and services that run in the VINES environment.

Chapter 22

XNS

22

Background

The Xerox Network Systems (XNS) protocols were created by the Xerox Corporation in the late 1970s and early 1980s. They were designed to be used across a variety of communication media, processors, and office applications. Several XNS protocols resemble the *Internet Protocol (IP)* and *Transmission Control Protocol (TCP)* protocols developed by the Defense Advanced Research Projects Agency (DARPA) for the U.S. Department of Defense (DoD). See Chapter 18, “Internet Protocols,” for more information on these and related protocols. All XNS protocols meet the basic design objectives of the OSI reference model.

Due to its availability and early entry into the market, XNS was adopted by most of the early LAN companies, including Novell, Inc., Ungermann-Bass, Inc. (now a part of Tandem Computers), and 3Com Corporation. Each of these companies have since made various changes to the XNS protocols. Novell added the *Service Access Protocol (SAP)* to permit resource advertisement and modified the OSI Layer 3 protocols (which Novell renamed *IPX*, for *Internetwork Packet Exchange*) to run on IEEE 802.3 networks rather than Ethernet. Ungermann-Bass modified RIP to support delay as well as hop count. Other small changes were also made. Over time, the PC networking XNS implementations have become more popular than XNS as it was designed by Xerox.

Technology Basics

Although the design objectives are the same, the XNS concept of a protocol hierarchy is somewhat different than that provided by the OSI reference model. A rough comparison is shown in Figure 22-1.

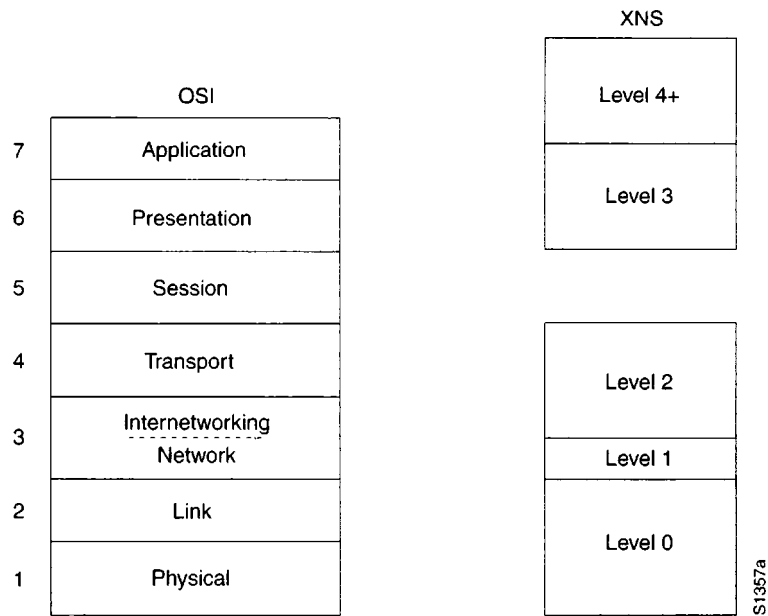


Figure 22-1 XNS and the OSI Reference Model

As seen in Figure 22-1, Xerox provided a five-level model of packet communications. Level 0 corresponds roughly to OSI Layers 1 and 2, handling link access and bit stream manipulation. Level 1 corresponds roughly to the portion of OSI Layer 3 that pertains to network traffic. Level 2 corresponds roughly to the portion of OSI Layer 3 that pertains to internetwork routing, and to OSI Layer 4, which handles interprocess communication. Levels 3 and 4 correspond roughly to the upper two layers of the OSI model, handling data structuring, process-to-process interaction and applications. XNS has no protocol corresponding to OSI Layer 5 (the session layer).

Media Access

Although XNS documentation mentions X.25, Ethernet, and HDLC, XNS does not expressly define what it refers to as a level zero protocol. Like many other protocol suites, XNS leaves media access an open issue, implicitly allowing any such protocol to host the transport of XNS packets over a physical medium.

Network Layer

The XNS network-layer protocol is called the *Internet Datagram Protocol (IDP)*. IDP performs standard Layer 3 functions, including logical addressing and end-to-end datagram delivery across an internetwork. The format of an IDP packet is shown in Figure 22-2.

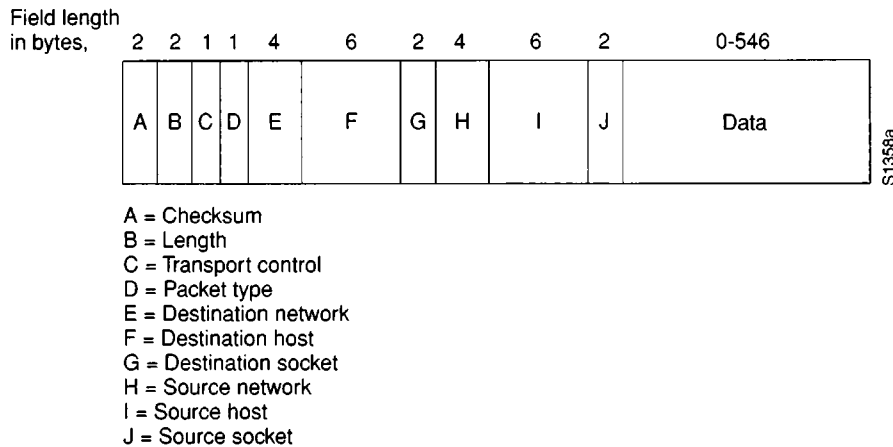


Figure 22-2 IDP Packet Format

The first field in an IDP packet is a 16-bit *checksum* field, which helps gauge the integrity of the packet after it traverses the internetwork.

Following the checksum is a 16-bit *length* field, which carries the complete length (including checksum) of the current datagram.

After the length field is an 8-bit *transport control* field and an 8-bit *packet type* field. The transport control field contains *hop count* and *maximum packet lifetime (MPL)* subfields. The hop count subfield is initialized to zero by the source and incremented by one as the datagram passes through a router. When the hop count field reaches 16, the datagram is discarded on the assumption that a routing loop is occurring. The MPL subfield provides the maximum amount of time, in seconds, that a packet can remain on the internetwork.

Following the transport control field is an 8-bit *packet type* field. This field specifies the format of the data field.

XNS source and destination network addresses each have three fields: a 32-bit *network number* that uniquely identifies a network in an internetwork, a 48-bit *host number* that is unique across all hosts ever manufactured, and a 16-bit *socket number* that uniquely identifies a socket (process) within a particular host. IEEE 802 addresses are equivalent to host numbers, so hosts that are connected to more than one IEEE 802 network have the same address on each segment. This makes network numbers redundant, but nevertheless useful for routing. Certain socket numbers are *well-known*, meaning that the service performed by the software using them is statically defined. All other socket numbers are reusable.

XNS supports *unicast* (point-to-point), *multicast*, and *broadcast* packets. Multicast and broadcast addresses are further divided into *directed* and *global* types. Directed multicasts deliver packets to members of the multicast group on the network specified in the destination

multicast network address. Directed broadcasts deliver packets to all members of a specified network. Global multicasts deliver packets to all members of the group within the entire internet, whereas global broadcasts deliver packets to all internet addresses. One bit in the host number indicates a single versus a multicast address. All ones in the host field indicates a broadcast address.

To route packets in an internetwork, XNS uses a dynamic routing scheme called the *Routing Information Protocol (RIP)*. Today, RIP is the most commonly used *interior gateway protocol (IGP)* in the Internet community, an international network environment providing connectivity to virtually every university and research institute and many commercial organizations in the United States. See Chapter 23, “RIP,” for more information on RIP.

Transport Layer

OSI transport-layer functions are implemented by several protocols. Each of the following protocols is described in the XNS specification as a level two protocol.

The *Sequenced Packet Protocol (SPP)* provides reliable, connection-based, flow-controlled packet transmission on behalf of client processes. It is similar in function to the Internet protocol suite’s *Transmission Control Protocol (TCP)* and the OSI protocol suite’s *Transport Protocol 4 (TP4)*. See Chapter 18, “Internet Protocols,” for more information on TCP and Chapter 20, “OSI Protocols,” for more information on TP4.

Each SPP packet includes a *sequence number*, which is used to order packets and to determine if any have been duplicated or missed. SPP packets also contain two 16-bit *connection identifiers*. One connection identifier is specified by each end of the connection. Together, the two connection identifiers uniquely identify a logical connection between client processes.

SPP packets cannot be longer than 576 bytes. Client processes can negotiate use of a different packet size during connection establishment, but SPP does not define the nature of this negotiation.

The *Packet Exchange Protocol (PEP)* is a request-response protocol designed to have greater reliability than simple datagram service (as provided by IDP, for example), but less reliability than SPP. PEP is functionally similar to the Internet protocol suite’s *User Datagram Protocol (UDP)*. See Chapter 18, “Internet Protocols,” for more information on UDP. PEP is single-packet based, providing retransmissions but no duplicate packet detection. As such, it is useful in applications where request-response transactions are idempotent (repeatable without context damage), or where reliable transfer is executed at another layer.

The *Error Protocol (EP)* can be used by any client process to notify another client process that a network error has occurred. This protocol is used, for example, in situations where an SPP implementation has identified a duplicate packet.

Upper-Layer Protocols

XNS offers several upper-layer protocols. The *Printing* protocol provides print services. The *Filing* protocol provides file-access services. The *Clearinghouse* protocol provides name services. Each of these three protocols runs on top of the *Courier* protocol, which provides conventions for data structuring and process interaction.

XNS also defines level four protocols. These are application protocols, but since they have little to do with actual communication functions, the XNS specification does not include any pertinent definitions.

Finally, the *Echo Protocol* is used to test the reachability of XNS network nodes. It is used to support functions such as that provided by the **ping** command found in UNIX and other environments. The XNS specification describes the Echo Protocol as a level-two protocol.



Chapter 23

RIP

23

Background

Routing Information Protocol (RIP) is a routing protocol originally designed for Xerox *PARC Universal Protocol* (where it was called GWINFO) and used in the *Xerox Network Systems (XNS)* protocol suite. RIP became associated with both UNIX and *Transmission Control Protocol/Internet Protocol (TCP/IP)* in 1982 when the Berkeley Standard Distribution (BSD) version of UNIX began shipping with a RIP implementation referred to as *routed* (pronounced “route dee”). Still a very popular routing protocol in the Internet community, RIP is formally defined in the XNS *Internet Transport Protocols* publication (1981) and in *Request for Comments (RFC) 1058* (1988).

RIP has been widely adopted by personal computer (PC) manufacturers for use in their networking products. For example, AppleTalk’s routing protocol (*Routing Table Maintenance Protocol*, also known as *RTMP*) is a modified version of RIP. RIP was also the basis for the routing protocols of Novell, 3Com, Ungermann-Bass, and Banyan. Novell and 3Com RIP is basically standard Xerox RIP. Ungermann-Bass and Banyan made minor modifications to RIP to serve their own needs.

Routing Table Format

Each entry in a RIP routing table provides a variety of information, including the ultimate destination, the next hop on the way to that destination, and a *metric*. The metric indicates the distance in number of hops to the destination. Other information can also be present in the routing table, including various timers associated with the route. A typical RIP routing table is shown in Figure 23-1.

Destination	Next-Hop	Distance	Timers	Flags
Network A	Router 1	3	t1, t2, t3	x, y
Network B	Router 2	5	t1, t2, t3	x, y
Network C	Router 1	2	t1, t2, t3	x, y
.
.
.

S1359a

Figure 23-1 Typical RIP Routing Table

RIP maintains only the best route to a destination. When new information provides a better route, this information replaces old route information. Network topology changes can provoke changes to routes, causing, for example, a new route to become the best route to a particular destination. When network topology changes occur, these changes are reflected in routing update messages. For example, when a router detects a link failure or a router failure, it recalculates its routes and sends routing update messages. Each router receiving a routing update message that includes a change updates its tables and propagates the change.

Packet Format (IP Implementations)

Figure 23-2 shows the RIP packet format for IP implementations, as specified by RFC 1058.

Note: Figure 23-2 shows the RIP format used for IP networks in the Internet. Some other RIP variations make slight modifications to the format and/or to the field names listed here, but the basic routing algorithm is functionally the same.

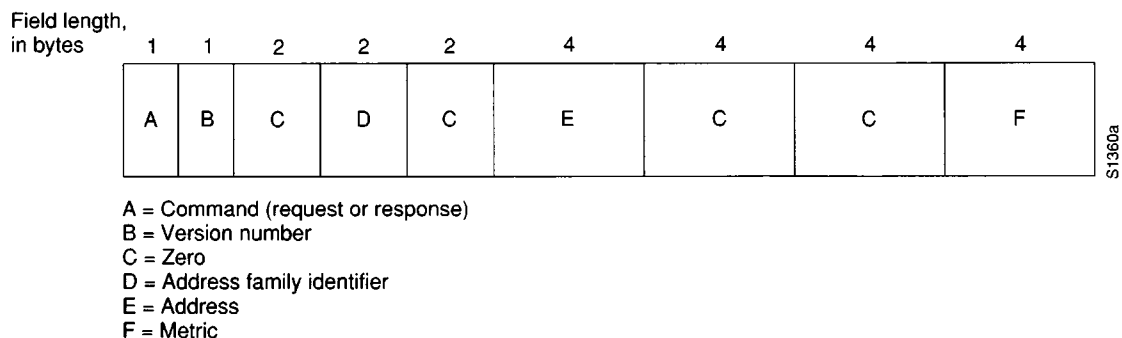


Figure 23-2 RIP Packet Format

The first field in an IP RIP packet is the *command* field. This field carries an integer indicating either a request or a response. The request command requests the responding system to send all or part of its routing table. Destinations for which a response is requested are listed later in the packet. The response command represents a reply to a request or, more frequently, an unsolicited regular routing update. In the response packet, a responding system includes all or part of its routing table. Regular routing update messages include the entire routing table.

The *version* field specifies the RIP version being implemented. With the potential for many RIP implementations in an internetwork, this field can be used to signal different, potentially incompatible implementations.

Following a 16-bit field of all zeros is the *address family identifier* field. This field specifies the particular address family being used. On the Internet (a large, international network connecting research institutions, government institutions, universities, and private businesses), this address family is typically IP (value = 2), but other network types may also be represented.

After another 16-bit field of zeros is a 32-bit *address* field. In Internet RIP implementations, this field typically contains an IP address.

Following two more 32-bit fields of zeros is the RIP *metric*. This metric is a *hop count*. It indicates how many internetwork hops (routers) must be traversed before the destination can be reached.

Up to 25 occurrences of the address family identifier through metric fields are permitted to occur in any single IP RIP packet. In other words, up to 25 destinations may be listed in any single RIP packet. Multiple RIP packets are used to convey information from larger routing tables.

Like other routing protocols, RIP uses certain timers to regulate its performance. The RIP *routing update timer* is generally set to 30 seconds, ensuring that each router will send a complete copy of its routing table to all neighbors every 30 seconds. The *route invalid timer* determines how much time must expire without having heard about a particular route before that route is considered invalid. When a route is marked invalid, neighbors are notified of this fact. This notification must occur prior to expiration of the *route flush timer*. When the route flush timer expires, the route is removed from the routing table. Typical initial values for these timers are 90 seconds for the route invalid timer and 270 seconds for the route flush timer.

Stability Features

RIP specifies a number of features designed to make its operation more stable in the face of rapid network topology changes. These include a hop count limit, *hold-downs*, *split-horizons*, and *poison reverse updates*.

Hop Count Limit

RIP permits a maximum hop count of 15. Any destination greater than 15 hops away is tagged as unreachable. RIP's maximum hop count greatly restricts its use in large internetworks, but prevents a problem called *count to infinity* from causing endless network routing loops. The count to infinity problem is shown in Figure 23-3.

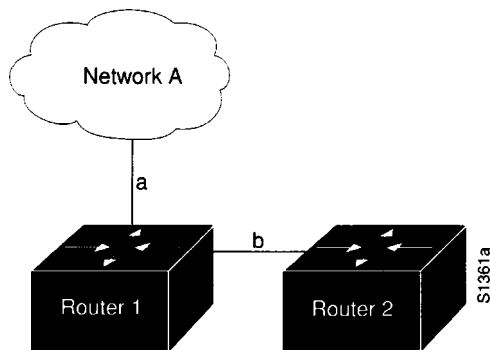


Figure 23-3 Count-To-Infinity Problem

In Figure 23-3, consider what will happen if Router 1's (R1's) link (link a) to Network A fails. R1 examines its information and sees that Router 2 (R2) has a one-hop link to Network A. Since R1 knows it is directly connected to R2, it advertises a two-hop path to Network A and begins routing all traffic to Network A through R2. This creates a routing loop. When R2 sees that R1 can now get to Network A in 2 hops, it changes its own routing table entry to show that it has a 3-hop path to Network A. This problem, and the routing loop, will continue infinitely, or until some external boundary condition is imposed. That boundary condition is RIP's hop-count maximum. When the hop count exceeds 15, the route is marked unreachable. Over time, the route is removed from the table.

Hold-Downs

Hold-downs are used to prevent regular update messages from inappropriately reinstating a route that has gone bad. When a route goes down, neighboring routers will detect this. These routers then calculate new routes and send out routing update messages to inform their neighbors of the route change. This activity begins a wave of routing updates that filter through the network.

Triggered updates do not instantly arrive at every network device. It is therefore possible that a device that has yet to be informed of a network failure may send a regular update message (indicating that a route that has just gone down is still good) to a device that has just been notified of the network failure. In this case, the latter device now contains (and potentially advertises) incorrect routing information.

Hold-downs tell routers to hold down any changes that might affect recently removed routes for some period of time. The hold-down period is usually calculated to be just greater than the period of time necessary to update the entire network with a routing change. Hold-down prevents the count-to-infinity problem.

Split Horizons

Split horizons takes advantage of the fact that it is never useful to send information about a route back in the direction from which it came. To illustrate, consider Figure 23-4.

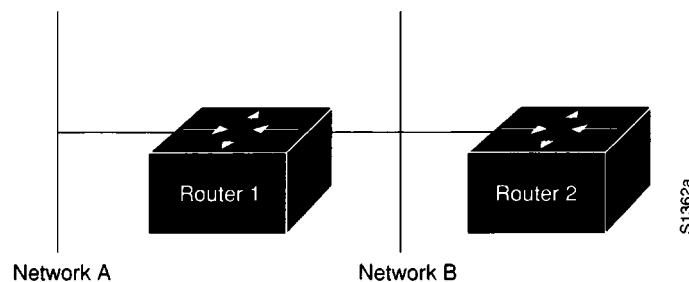


Figure 23-4 Split Horizons

Router 1 (R1) initially advertises that it has a route to Network A. There is no reason for Router 2 (R2) to include this route in its update back to R1, as R1 is closer to Network A. The split horizon rule says that R2 should strike this route from any updates it sends to R1.

The split horizon rule helps prevent two-node routing loops. For example, consider the case where R1's interface to Network A goes down. Without split horizons, R2 continues to inform R1 that it can get to Network A through R1. If R1 does not have sufficient intelligence, it might actually pick up R2's route as an alternative to its failed direct connection, causing a routing loop. Although hold downs should prevent this, split horizon provides extra algorithm stability.

Poison Reverse Updates

Whereas split horizons should prevent routing loops between adjacent routers, poison reverse updates are intended to defeat larger routing loops. The idea is that increases in routing metrics generally indicate routing loops. Poison reverse updates are then sent to remove the route and place it in hold down.

Chapter 24

IGRP

24

Background

The Interior Gateway Routing Protocol (IGRP) is a routing protocol developed in the mid-1980s by Cisco Systems, Inc. Cisco's principle goal in creating IGRP was to provide a robust protocol for routing within an *autonomous system (AS)* having arbitrarily complex topology and consisting of media with diverse bandwidth and delay characteristics. An AS is a collection of networks under common administration that share a common routing strategy. ASs are typically given a unique 16-bit number that is assigned by the *Defense Data Network (DDN) Network Information Center (NIC)*.

In the mid-1980s, the most popular intra-AS routing protocol was the *Routing Information Protocol (RIP)*. Although quite useful for routing within small-to-moderate sized, relatively homogeneous internetworks, RIP's limits were being pushed by network growth. In particular, RIP's small hop-count limit (15) restricted the size of internetworks, and its single metric (hop count) did not allow for much routing flexibility in complex environments. For more information on RIP, see Chapter 23, "RIP." The popularity of Cisco routers and the robustness of IGRP have encouraged many organizations with large internetworks to replace RIP with IGRP.

Cisco's initial IGRP implementation worked in *Internet Protocol (IP)* networks. IGRP was designed to run in any network environment, however, and Cisco soon ported it to run in *Open System Interconnection (OSI) Connectionless Network Protocol (CLNP)* networks.

Technology

IGRP is a *distance-vector, interior-gateway protocol (IGP)*. Distance-vector routing protocols call for each router to send all or a portion of their routing table in a routing update message at regular intervals to each of its neighbor routers. As routing information proliferates through the network, routers can calculate distances to all nodes within the internetwork.

Distance-vector routing protocols are often contrasted with link-state routing protocols, which send local connection information to all nodes in the internetwork. See Chapter 25, "OSPF," and Chapter 28, "OSI Routing," respectively, for a discussion of *Open Shortest Path First (OSPF)* and *Intermediate System to Intermediate System (IS-IS)*, two popular link-state routing algorithms.

IGRP uses a combination (vector) of metrics. *Internetwork delay, bandwidth, reliability, and load* are all factored into the routing decision. Network administrators can set the weighting factors for each of these metrics. IGRP uses either the administrator-set or the default weightings to automatically calculate optimal routes.

IGRP provides a wide range for its metrics. For example, reliability and load can take on any value between 1 and 255, bandwidth can take on values reflecting speeds from 1200 to 10 gigabits per second, while delay can take on any value from 1 to 2 to the 24th power. Wide metric ranges allow satisfactory metric setting in internetworks with widely varying performance characteristics. Most importantly, the metric components are combined in a user-definable algorithm. As a result, network administrators can influence route selection in an intuitive fashion.

To provide additional flexibility, IGRP permits multipath routing. Dual equal-bandwidth lines may run a single stream of traffic in round-robin fashion, with automatic switchover to the second line if one line goes down. Also, multiple paths can be used even if the metrics for the paths are different. For example, if one path is three times better than another by virtue of its metric being three times lower, the better path will be used three times as often. Only routes with metrics that are within a certain range of the best route are used as multiple paths.

Packet Format

The IGRP packet format is shown in Figure 24-1.

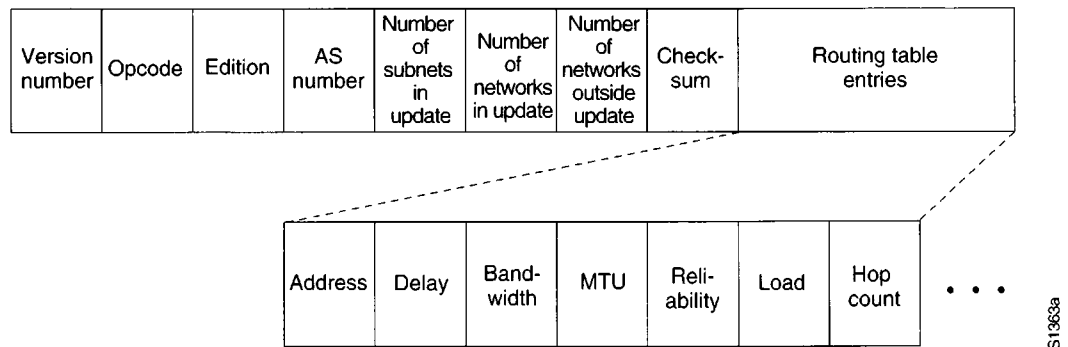


Figure 24-1 IGRP Packet Format

The first field of the IGRP packet contains a *version number*. This version number indicates the version of IGRP being used, and helps signal different, potentially incompatible implementations.

Following the version field is an *opcode* field. This field indicates the packet type. An opcode equal to one indicates an update packet; an opcode equal to two indicates a request packet.

Request packets are used by the source to request a routing table from another router. These packets consist only of a header containing the version, opcode, and AS number fields. Update packets contain a header, followed immediately by routing table entries. No limit on routing table entries is imposed except that the packet cannot be larger than 1500 bytes, including IP header. Multiple packets can be sent if this is insufficient for coverage of a routing table.

After the version field is an *edition* field. This field contains a serial number that is incremented when the routing table changes in some way. This edition value is used to allow routers to avoid processing updates containing information that they have already seen.

Following the edition field is a field containing the *AS number*. This field is required because Cisco routers can span multiple ASs. Multiple ASs (or IGRP processes) in one router keep AS routing information separate.

The next three fields indicate the number of subnets in the update packet, the number of major networks in the update packet, and the number of external networks in the update packet. These fields are present because IGRP update messages have three portions: interior to the subnet, interior to the current AS, and exterior to the current AS. Only subnets of the network associated with the address to which the update is being sent are included. Major networks (that is, nonsubnets) are put into the interior-to-the-AS portion of the packet unless they are expressly flagged as exterior. Networks are flagged as exterior if information about them arrived in the exterior portion of a message from another router.

The final field in the IGRP header is the *checksum* field. This field contains a checksum for the IGRP header and any update information contained in the packet. Checksum computation allows a receiving router to check the validity of the incoming packet.

Update messages contain a series of seven data fields for each routing table entry. The first such field contains the three significant bytes of the *address* (in the case of an IP address). The next five fields contain metric values. The first of these indicates the *delay*, in tens of microseconds. The range covered is 10 microseconds to 167 seconds. Following the delay field is the *bandwidth* field. Bandwidth is given in units of 1 kilobit per second, and covers a range from a 1200 bit/second line to 10 gigabits/second. Next is the *MTU* field, which provides the MTU size, in bytes. After the MTU field is the *reliability* field, which indicates the percentage of packets successfully transmitted and received. Finally, there is the *load* field, which indicates the percentage of the channel occupied. The last field in each routing entry is the *hop count* field. Although hop count is not expressly used in metric determination, it is still carried in the IGRP packet and incremented after packet processing, allowing hop count to be used to suppress loops.

Stability Features

IGRP contains a number of features designed to enhance its own stability. These include *hold-downs*, *split horizons*, and *poison reverse updates*.

Hold-Downs

Hold-downs are used to prevent regular update messages from inappropriately reinstating a route that may have gone bad. When a router goes down, neighboring routers will detect this via the lack of regularly scheduled update messages. These routers then calculate new routes and send routing update messages to inform their neighbors of the route change. This activity begins a wave of triggered updates that filter through the network.

These triggered updates do not instantly arrive at every network device. It is therefore possible that a device that has yet to be informed of a network failure may send a regular update message (indicating that a route that has just gone down is still good) to a device that has just been notified of the network failure. In this case, the latter device will now contain (and potentially advertise) incorrect routing information.

Hold-downs tell routers to hold down any changes that might affect routes for some period of time. The hold-down period is usually calculated to be just greater than the period of time necessary to update the entire network with a routing change.

Split Horizons

Split horizons derive from the fact that it is never useful to send information about a route back in the direction from which it came. To illustrate, consider Figure 24-2.

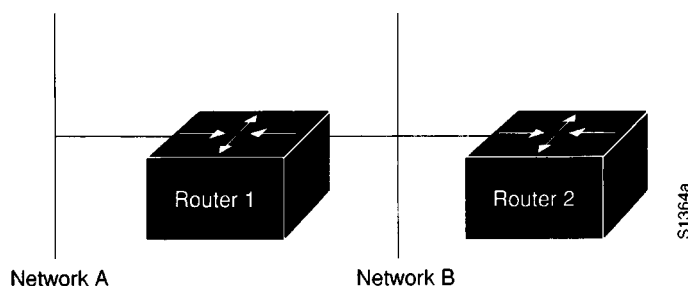


Figure 24-2 Split Horizons

Router 1 (R1) initially advertises that it has a route to Network A. There is no reason for Router 2 (R2) to include this route in its update back to R1, as R1 is closer to Network A. The split-horizon rule says that R2 should strike this route from any updates it sends to R1.

The split-horizon rule helps prevent routing loops. For example, consider the case where R1's interface to Network A goes down. Without split horizons, R2 continues to inform R1 that it can get to Network A (through R1!). If R1 does not have sufficient intelligence, it may actually pick up R2's route as an alternative to its failed direct connection, causing a routing loop. Although hold downs should prevent this, split horizons are implemented in IGRP because they provide extra algorithm stability.

Poison Reverse Updates

Whereas split horizons should prevent routing loops between adjacent routers, poison reverse updates are intended to defeat larger routing loops. Increases in routing metrics generally indicate routing loops. Poison reverse updates are then sent to remove the route and place it in hold down. In Cisco's implementation of IGRP, poison reverse updates are sent if a route metric has increased by a factor of 1.1 or greater.

Timers

IGRP maintains a number of timers and variables containing time intervals. These include an update timer, an invalid timer, a hold time period, and a flush timer. The update timer specifies how frequently routing update messages should be sent. The IGRP default for this variable is 90 seconds. The invalid timer specifies how long a router should wait, in the absence of routing update messages about a specific route, before declaring that route invalid. The IGRP default for this variable is three times the update period. The hold time variable specifies the hold down period. The IGRP default for this variable is three times the update timer period plus ten seconds. Finally, the flush timer indicates how much time should pass before a route should be flushed from the routing table. The IGRP default is seven times the routing update period.

Chapter 25

OSPF

25

Background

Open Shortest Path First (OSPF) is a routing protocol developed for *Internet Protocol (IP)* networks by the *interior gateway protocol (IGP)* working group of the Internet Engineering Task Force (IETF). The working group was formed in 1988 to design an IGP based on the *shortest path first (SPF)* algorithm for use in the *Internet*, a large, international network connecting research institutions, government institutions, universities, and private businesses. Like the *Interior Gateway Routing Protocol (IGRP)* (see Chapter 24, “IGRP,” for more information), OSPF was created because the *Routing Information Protocol (RIP)* was, in the mid-1980s, increasingly unable to serve large, heterogeneous internetworks. See Chapter 23, “RIP” for more information on RIP.

OSPF was derived from several research efforts, including:

- Bolt, Beranek, and Newman’s (BBN’s) SPF algorithm developed for the *Arpanet* (a landmark packet-switching network developed in the early 1970s by BBN) in 1978
- Radia Perlman’s research on fault-tolerant broadcasting of routing information (1988)
- BBN’s work on area routing (1986)
- An early version of OSI’s IS-IS routing protocol

See Chapter 28, “OSI Routing,” for more information on IS-IS.

As indicated by its acronym, OSPF has two primary characteristics. The first is that it is open, in that its specification is in the public domain. The OSPF specification is published as *Request For Comments (RFC) 1247*. The second principle characteristic is that it is based on the SPF algorithm. The SPF algorithm is sometimes referred to as the *Dijkstra algorithm*, after the person credited with its creation.

Technology Basics

OSPF is a *link-state* routing protocol. As such, it calls for the sending of *link-state advertisements* (LSAs) to all other routers within the same hierarchical area. Information on attached interfaces, metrics used, and other variables is included in OSPF LSAs. As OSPF routers accumulate link-state information, they use the SPF algorithm to calculate the shortest path to each node.

As a link-state algorithm, OSPF contrasts with RIP and IGRP, which are *distance-vector* routing protocols. Routers running the distance-vector algorithm send all or a portion of their routing tables in routing update messages, but only to their neighbors.

Routing Hierarchy

Unlike RIP, OSPF can operate within a hierarchy. The largest entity within the hierarchy is the *autonomous system* (AS). An AS is a collection of networks under a common administration, sharing a common routing strategy. OSPF is an intra-AS (interior gateway) routing protocol, although it is capable of receiving routes from and sending routes to other ASs.

An AS can be divided into a number of *areas*. An area is a group of contiguous networks and attached hosts. Routers with multiple interfaces can participate in multiple areas. These routers, which are called *area border routers*, maintain separate topological databases for each area.

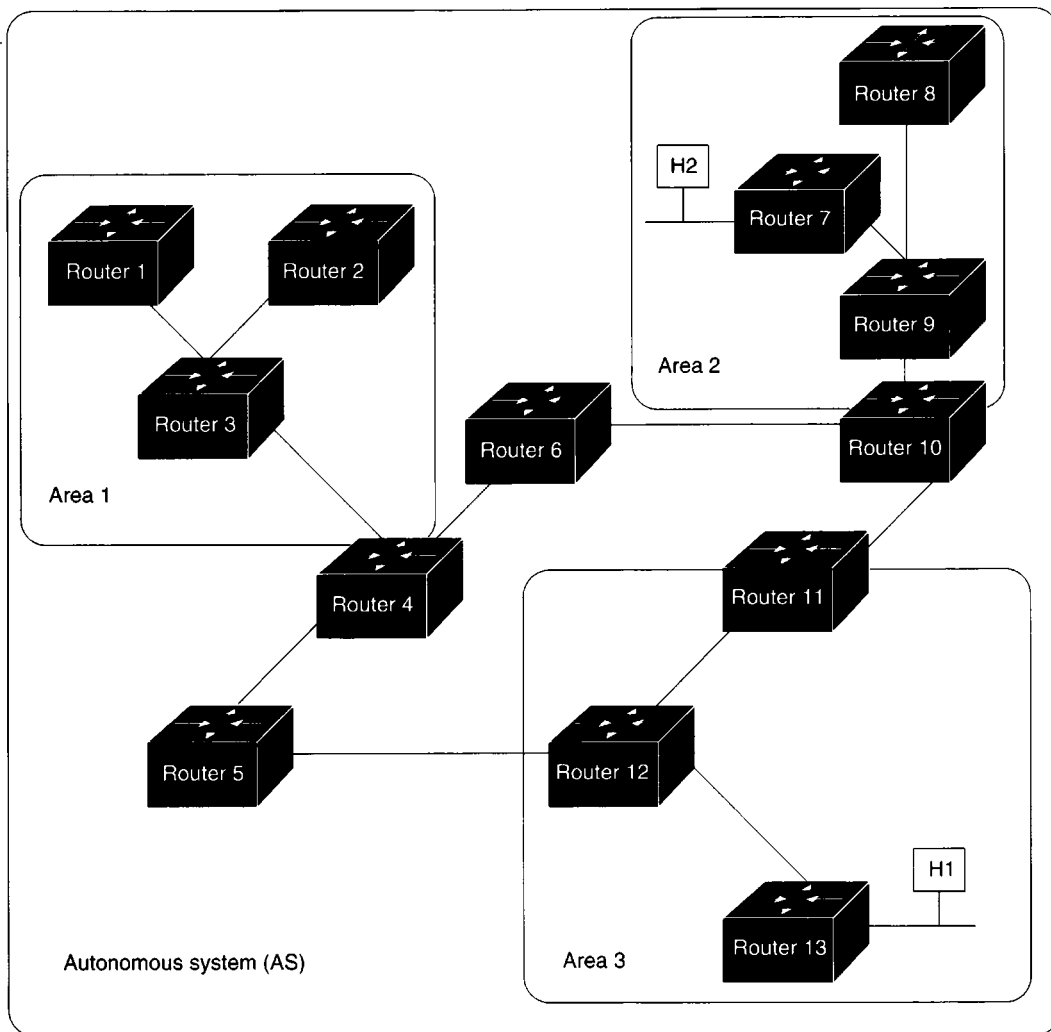
A *topological database* is essentially an overall picture of networks in relationship to routers. The topological database contains the collection of LSAs received from all routers in the same area. Because routers within the same area share the same information, they have identical topological databases.

The term *domain* is sometimes used to describe a portion of the network in which all routers have identical topological databases. Domain is frequently used interchangeably with AS.

An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than if the AS were not partitioned.

Area partitioning creates two different types of OSPF routing, depending on whether the source and destination are in the same or different areas. Intra-area routing occurs when the source and destination are in the same area; interarea routing occurs when they are in different areas.

An OSPF *backbone* is responsible for distributing routing information between areas. It consists of all area border routers, networks not wholly contained in any area, and their attached routers. Figure 25-1 shows an example of an internetwork with several areas.



S1365a

Figure 25-1 Hierarchical OSPF Internetwork

In this figure, Routers 4, 5, 6, 10, 11, and 12 make up the backbone. If host H1 in Area 3 wishes to send a packet to host H2 in Area 2, the packet is sent to Router 13, which forwards the packet to Router 12, which sends the packet to Router 11. Router 11 forwards the packet along the backbone to area border router Router 10, which sends the packet through two intra-area routers (Router 9 and Router 7) until it can be forwarded to host H2.

The backbone itself is an OSPF area, so all backbone routers use the same procedures and algorithms to maintain routing information within the backbone that any area router would. The backbone topology is invisible to all intra-area routers, as are individual area topologies to the backbone.

Areas can be defined in such a way that the backbone is not contiguous. In this case, backbone connectivity must be restored through *virtual links*. Virtual links are configured between any backbone routers that share a link to a nonbackbone area, and function as if they were direct links.

AS border routers running OSPF learn about exterior routes through *exterior gateway protocols (EGPs)* such as *Exterior Gateway Protocol (EGP)* or *Border Gateway Protocol (BGP)*, or through configuration information. See Chapter 26, “EGP” and Chapter 27, “BGP,” respectively, for more information on these protocols.

The SPF Algorithm

The SPF routing algorithm is the basis for OSPF operations. When an SPF router is powered up, it initializes its routing protocol data structures and then waits for indications from lower-layer protocols that its interfaces are functional.

Once assured that its interfaces are functioning, a router uses the OSPF *Hello protocol* to acquire *neighbors*. Neighbors are routers with interfaces to a common network. The router sends hello packets to its neighbors and receives their hello packets. In addition to helping acquire neighbors, hello packets also act as keepalives to let routers know that other routers are still functional.

On *multi-access networks* (networks supporting more than two routers), the Hello protocol elects a *designated router* and a backup designated router. The designated router is responsible, among other things, for generating LSAs for the entire multi-access network. Designated routers allow a reduction in network traffic and in the size of the topological database.

When the link-state databases of two neighboring routers are synchronized, the routers are said to be *adjacent*. On multi-access networks, the designated router determines which routers should become adjacent. Topological databases are synchronized between pairs of adjacent routers. Adjacencies control the distribution of routing protocol packets. These packets are sent and received only on adjacencies.

Each router periodically sends an LSA. LSAs are also sent when a router's state changes. LSAs include information on a router's adjacencies. By comparing established adjacencies to link states, failed routers can be quickly detected, and the network's topology altered appropriately. From the topological database generated from LSAs, each router calculates a shortest-path tree, with itself as root. The shortest-path tree, in turn, yields a routing table.

Packet Format

All OSPF packets begin with a 24-byte header, as shown in Figure 25-2.

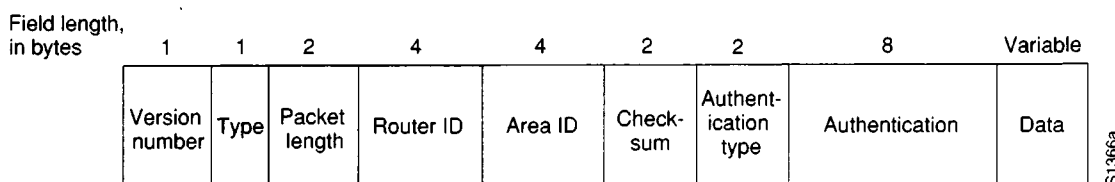


Figure 25-2 OSPF Header Format

The first field in the OSPF header is the OSPF *version number*. The version number identifies the particular OSPF implementation being used.

Following the version number is the *type* field. There are five OSPF packet types:

- *Hello*—Sent at regular intervals to establish and maintain neighbor relationships.
- *Database Description*—Describes the contents of the topological database, and are exchanged when an adjacency is being initialized.
- *Link-State Request*—Requests pieces of a neighbor's topological database. They are exchanged after a router has discovered (through examination of database description packets) that parts of its topological database are out of date.
- *Link-State Update*—Responses to link-state request packets. They are also used for the regular dispersal of LSAs. Several LSAs may be included within a single packet.
- *Link-State Acknowledgment*—Acknowledges link-state update packets. Link-state update packets must be explicitly acknowledged to ensure that link-state flooding throughout an area is a reliable process.

Each LSA in a link-state update packet contains a type field. There are four LSA types:

- *Router links advertisements (RLAs)*—Describe the collected states of the router's links to a specific area. A router sends an RLA for each area to which it belongs. RLAs are flooded throughout the entire area, and no further.
- *Network links advertisements (NLAs)*—Sent by the designated routers. They describe all the routers that are attached to a multi-access network, and are flooded throughout the area containing the multi-access network.
- *Summary links advertisements (SLAs)*—Summarize routes to destinations outside an area, but within the AS. They are generated by area border routers, and are flooded throughout the area. Only intra-area routes are advertised into the backbone. Both intra- and inter-area routes are advertised into the other areas.
- *AS external links advertisements*—Describe a route to a destination that is external to the AS. AS external links advertisements are originated by AS boundary routers. This type of advertisement is the only type that is forwarded everywhere in the AS; all others are forwarded only within specific areas.

Following the OSPF packet header's type field is a *packet length* field. This field provides the packet's length, including the OSPF header, in bytes.

The *router ID* field identifies the packet's source.

The *area ID* field identifies the area to which the packet belongs. All OSPF packets are associated with a single area.

A standard IP *checksum* field checks the entire packet contents for potential damage suffered in transit.

After the checksum is the *authentication type* field. "Simple password" is an example of an authentication type. All OSPF protocol exchanges are authenticated. The authentication type is configurable on a per-area basis.

Following the authentication type is the *authentication* field. This field is 64 bits in length, and contains authentication information.

Additional OSPF Features

Additional OSPF features include equal-cost, *multipath routing* and routing based on upper-layer *type of service (TOS)* requests. TOS-based routing supports those upper-layer protocols that can specify particular types of service. For example, an application might specify that certain data is urgent. If OSPF has high-priority links at its disposal, these can be used to transport the urgent datagram.

OSPF supports one or more metrics. If only one metric is used, it is considered to be arbitrary, and TOS is not supported. If more than one metric is used, TOS is optionally supported through the use of a separate metric (and, therefore, a separate routing table) for each of the eight combinations created by the three IP TOS bits (the *delay*, *throughput*, and *reliability* bits). For example, if the IP TOS bits specify low delay, low throughput, and high reliability, OSPF calculates routes to all destinations based on this TOS designation.

IP subnet masks are included with each advertised destination, enabling *variable-length subnet masks*. With variable-length subnet masks, an IP network can be broken into many subnets of various sizes. This provides network administrators with extra network configuration flexibility.

Chapter 26

EGP

26

Background

The Exterior Gateway Protocol (EGP) is an interdomain reachability protocol used in the *Internet*, an international network connecting universities, governments institutions, research organizations, and private commercial concerns. EGP is documented in *Request For Comments (RFC) 904*, published in April, 1984.

As the first exterior gateway protocol to gain widespread acceptance in the Internet, EGP served a valuable purpose. Unfortunately, EGP's weaknesses have become more apparent as the Internet has grown and matured. Because of these weaknesses, EGP is currently being phased out of the Internet, and is being replaced by other exterior gateway protocols such as the *Border Gateway Protocol (BGP)* and the *Inter-Domain Routing Protocol (IDRP)*. For more information on these protocols, see Chapter 27, "BGP," and Chapter 28, "OSI Routing."

Technology Basics

EGP was originally designed to communicate reachability to and from the ARPANET core routers. Information was passed from individual source nodes in distinct Internet administrative domains called *autonomous systems (ASs)* up to the core routers, which passed the information through the backbone until it could be passed down to the destination network within another AS. This relationship between EGP and other ARPANET components is shown in Figure 26-1.

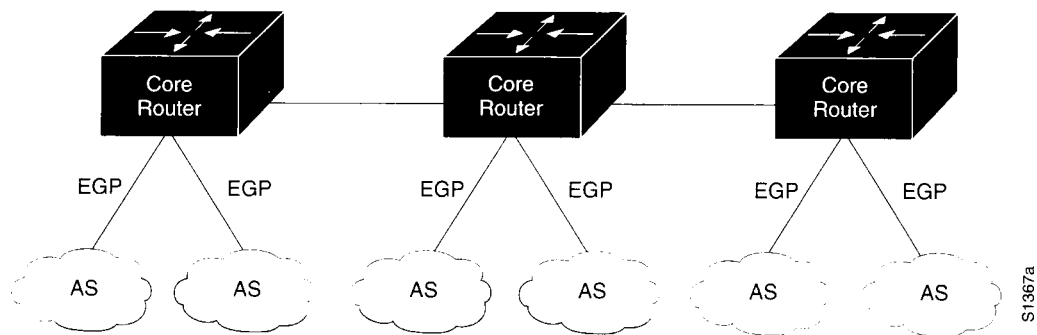


Figure 26-1 EGP and the ARPANET

Although EGP is a dynamic routing protocol, it uses a very simple design. It does not use metrics and therefore cannot make true intelligent routing decisions. EGP routing updates contain network reachability information. In other words, they specify that certain networks are reachable through certain routers.

EGP has three primary functions. First, routers running EGP establish a set of *neighbors*. These neighbors are simply routers with which an EGP router wishes to share reachability information; there is no implication of geographic proximity. Second, EGP routers poll their neighbors to see if they are alive. Third, EGP routers send update messages containing information about the reachability of networks within their ASs.

Packet Format

The EGP packet is shown in Figure 26-2.

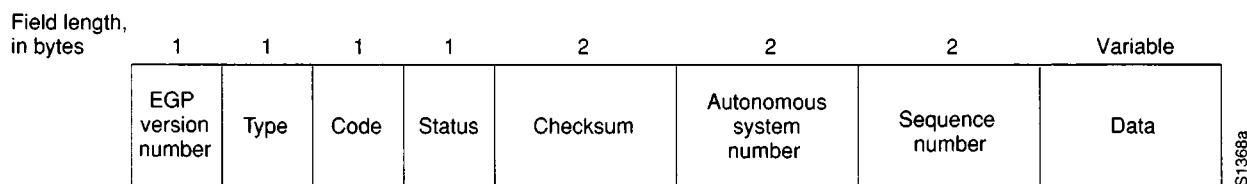


Figure 26-2 EGP Packet Format

The first field in the EGP packet header is the *EGP version number* field. This field identifies the current EGP version and is checked by receivers to determine whether there is a match between the sender and receiver version numbers.

The next field is the *type* field, which identifies the message type. EGP defines five separate message types. These are shown in Figure 26-3.

Message	Function
Neighbor acquisition	Establishes/de-establishes neighbors
Neighbor reachability	Determines if neighbors are alive
Poll	Determines reachability of a particular network
Update	Provides routing updates
Error	Indicates error conditions

S1368a

Figure 26-3 EGP Message Types

Following the type field is the *code* field. This field distinguishes among message subtypes.

Next is the *status* field, which contains message-dependent status information. Status codes include *insufficient resources*, *parameter problem*, *protocol violation*, and others.

After the status field is the *checksum* field. The checksum is used to detect possible problems that may have developed with the packet in transit.

An *autonomous system number* field follows the checksum. This identifies the AS to which the sending router belongs.

Finally, the last field in the EGP packet header is the *sequence number* field. This field allows two EGP routers exchanging messages to match requests with replies. The sequence number is initialized to zero when a neighbor is established and incremented by one with each request-response transaction.

Additional fields follow the EGP header. The contents of these fields vary, depending on the message type (as specified by the type field).

Message Types

Additional fields follow the EGP header. The contents of these fields vary depending on the message type (as specified by the type field).

Neighbor Acquisition

The *neighbor acquisition* message includes a *hello interval* and a *poll interval* field. The hello interval field specifies the interval period for testing whether neighbors are alive. The poll interval field specifies the routing update frequency.

Neighbor Reachability

The *neighbor reachability* message adds no extra fields to the EGP header. These messages use the code field to indicate whether the message is a hello message or a response to a hello message. Separating the reachability assessment function from the routing update function reduces network traffic because network reachability changes usually occur more often than routing parameter changes. Only after a specified percentage of reachability messages have not been received does an EGP node declare a neighbor to be down.

Poll

To provide correct routing between ASs, EGP must know the relative location of remote hosts. The *poll* message allows EGP routers to acquire reachability information about the networks on which these hosts reside. These messages only have one field beyond the common header—the *IP source network* field. This field specifies the network to be used as reference point for the request.

Routing Update

Routing update messages provide a way for EGP routers to indicate the locations of various networks within their ASs. In addition to the common header, these messages include many additional fields. The *number of interior gateways* field indicates the number of interior gateways appearing in the message. The *number of exterior gateways* field indicates the number of exterior gateways appearing in the message. The *IP source network* field provides the IP address of the network from which reachability is measured. Following this field is a series of *gateway blocks*. Each gateway block provides the IP address of a gateway and a list of networks and distances associated with reaching those networks.

Within the gateway block, EGP lists networks by distances. In other words, at distance three, there may be four networks. These networks are then listed by address. The next group of networks may be those that are distance four away, and so on.

EGP does not interpret the distance metrics that are contained within the routing update messages. In essence, EGP uses the distance field to indicate whether a path exists; the distance value can only be used to compare paths if those paths exist wholly within a particular AS. For this reason, EGP is more a reachability protocol than a routing protocol. This restriction also places topology limitations on the structure of the Internet. Specifically, an EGP portion of the Internet must be a tree structure in which a core gateway is the root, and there are no loops among other ASs within the tree. This restriction is a primary limitation of EGP, and provides an impetus for its gradual replacement by other, more capable exterior gateway protocols.

Error

Error messages identify various EGP error conditions. In addition to the common EGP header, EGP error messages provide a *reason* field, followed by an *error message header*. Typical EGP errors (reasons) include *bad EGP header format*, *bad EGP data field format*, *excessive polling rate*, and the *unavailability of reachability information*. The error message header consists of the first three 32-bit words of the EGP header.

Chapter 27

BGP



Background

Exterior gateway protocols are designed to route between routing domains. In the terminology of the Internet (an international network connecting universities, governments institutions, research organizations, and private commercial concerns), a routing domain is called an *autonomous system (AS)*. The first exterior gateway protocol to achieve widespread acceptance in the Internet was the *Exterior Gateway Protocol (EGP)*. See Chapter 26, “EGP,” for more information on EGP. Although a useful technology, EGP has several weaknesses, including the fact that it is more a reachability protocol than a routing protocol.

The Border Gateway Protocol (BGP) represents an attempt to address the most serious of EGP’s problems. BGP is an inter-AS routing protocol created for use in the Internet. Unlike EGP, BGP was designed to detect routing loops. BGP may be thought of as a next-generation EGP. Indeed, BGP and other inter-AS routing protocols are (slowly) replacing EGP in the Internet. BGP Version 3 is specified in *Request For Comments (RFC) 1163*.

Technology Basics

Although designed as an inter-AS protocol, BGP can be used both within and between ASs. Two BGP neighbors communicating between ASs must reside on the same physical network. BGP routers within the same AS communicate with one another to ensure that they have a consistent view of the AS and to determine which BGP router within that AS will serve as the connection point to/from certain external ASs.

Some ASs are merely pass-through channels for network traffic. In other words, some ASs carry network traffic that did not originate within and is not destined for them. BGP must interact with whatever intra-AS routing protocols exist within these pass-through ASs.

BGP update messages consist of network number/AS path pairs. The AS path contains the string of ASs through which the specified network may be reached. These update messages are sent over the TCP transport mechanism to ensure reliable delivery.

The initial data exchange between two routers is the entire BGP routing table. Incremental updates are sent out as the routing tables change. Unlike some other routing protocols, BGP does not require periodic refresh of the entire routing table. Instead, routers running BGP retain the latest version of each peer routing table. Although BGP maintains a routing table with all feasible paths to a particular network, it only advertises the primary (optimal) path in its update messages.

The BGP metric is an arbitrary unit number specifying the degree of preference of a particular path. These metrics are typically assigned by the network administrator through configuration files. Degree of preference may be based on any number of criteria, including AS count (paths with a smaller AS count are generally better), type of link (is the link stable? fast? reliable?), and other factors.

Packet Format

The BGP packet format is shown in Figure 27-1.

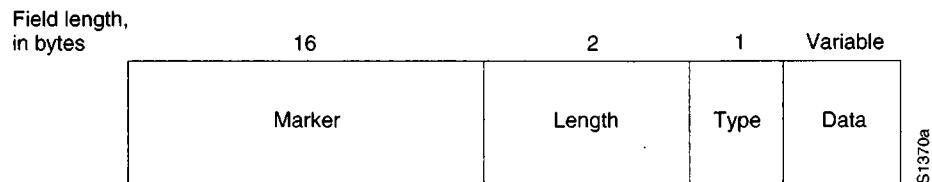


Figure 27-1 BGP Packet Format

BGP packets have a common 19-byte header consisting of three fields.

The *marker* field is 16 bytes long and contains a value that the receiver of the message can predict. This field is used for authentication.

The *length* field contains the total length of the message, in bytes.

The *type* field specifies the message type.

Messages

Four message types are specified in RFC 1163:

- Open
- Update
- Notification
- Keepalive

After a transport protocol connection is established, the first message sent by each side is an open message. If the open message is acceptable to the receiver, a keepalive message confirming the open is sent back. Upon successful confirmation of the open message, updates, keepalives, and notifications may be exchanged.

Open

In addition to the common BGP packet header, open messages define several fields. The *version* field provides a BGP version number, and allows the receiver to check that it is running the same version as the sender. The *autonomous system* field provides the AS number of the sender. The *hold time* field indicates the maximum number of seconds that may elapse without receipt of a message before the transmitter is assumed to be dead. The *authentication code* field indicates the authentication type being used (if any). The *authentication data* field contains actual authentication data (if any).

Update

BGP update messages provide routing updates to other BGP systems. Information in these messages is used to construct a graph describing the relationships of the various ASs. In addition to the common BGP header, update messages have several additional fields. These fields provide routing information by listing path attributes corresponding to each network.

BGP currently defines five attributes:

- *Origin*—Can take on one of three values: *IGP*, *EGP*, and *incomplete*. The IGP attribute means that the network is part of the AS. The EGP attribute means that the information was originally learned from the EGP protocol. BGP implementations would be inclined to prefer IGP routes over EGP routes, since EGP fails in the presence of routing loops. The incomplete attribute is used to indicate that the network is known via some other means.
- *AS path*—Provides the actual list of ASs on the path to the destination.
- *Next hop*—Provides the IP address of the router that should be used as the next hop to the networks listed in the update message.
- *Unreachable*—If present, indicates that a route is no longer reachable.
- *Inter-AS metric*—Provides a way for a BGP router to advertise its cost to destinations within its own AS. This information can be used by routers external to the advertiser's AS to select an optimal route into the AS to a particular destination.

Keepalive

Keepalive messages do not contain any additional fields beyond those in the common BGP header. These messages are sent often enough to keep the hold-time timer from expiring.

Notification

Notification messages are sent when an error condition has been detected and one router wishes to tell another why it is closing the connection between them. Aside from the common BGP header, notification messages have an *error code* field, an *error subcode* field, and *error data*. The error code field indicates the type of error, and can be one of the following:

- *Message header error*—Indicates a problem with the message header such as an unacceptable message length, an unacceptable marker field value, or an unacceptable message type.
- *Open message error*—Indicates a problem with an open message such as an unsupported version number, an unacceptable AS number or IP address, or an unsupported authentication code.
- *Update message error*—Indicates a problem with the update message. Examples include a malformed attribute list, an attribute list error, and an invalid next-hop attribute.
- *Hold time expired*—Indicates a hold time expiration, after which a BGP node will be declared dead.

Chapter 28

OSI Routing

28

Background

Several routing protocols have been or are being developed under the auspices of the *International Organization for Standardization (ISO)*. ISO refers to the *Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol (IS-IS)* as ISO 10589. The American National Standards Institute (ANSI) X3S3.3 (network and transport layers) committee was the motivating force behind ISO standardization of IS-IS. Other ISO protocols associated with routing include ISO 9542 (*End System to Intermediate System, or ES-IS*) and ISO 10747 (*IS-IS Inter-Domain Routing Protocol, or IDRP*). Both of these protocols will be mentioned briefly in this chapter, but the focus is on the intradomain version of IS-IS.

IS-IS is based on work originally done at Digital Equipment Corporation for Phase V DECnet. Although IS-IS was created to route in ISO *Connectionless Network Protocol (CLNP)* networks, a version has since been created to support both CLNP and *Internet Protocol (IP)* networks. This variety of IS-IS is usually referred to as *Integrated IS-IS* and has also been called *Dual IS-IS*. Integrated IS-IS is also discussed briefly.

Terminology

The world of OSI internetworking has a unique terminology. The term *end system (ES)* refers to any nonrouting network node; the term *intermediate system (IS)* refers to a router. These terms are the basis for the OSI protocols ES-IS (which allows ESs and ISs to discover each other) and IS-IS (which provides routing between ISs). Several other important OSI internetworking terms are defined as follows:

- *Area*—A group of contiguous networks and attached hosts that are specified to be an area by a network administrator or similar person.
- *Domain*—A collection of connected areas. Routing domains provide full connectivity to all end systems within them.
- *Level 1 routing*—Routing within a Level 1 area.
- *Level 2 routing*—Routing between Level 1 areas.

Figure 28-1 shows the relationship between these terms.

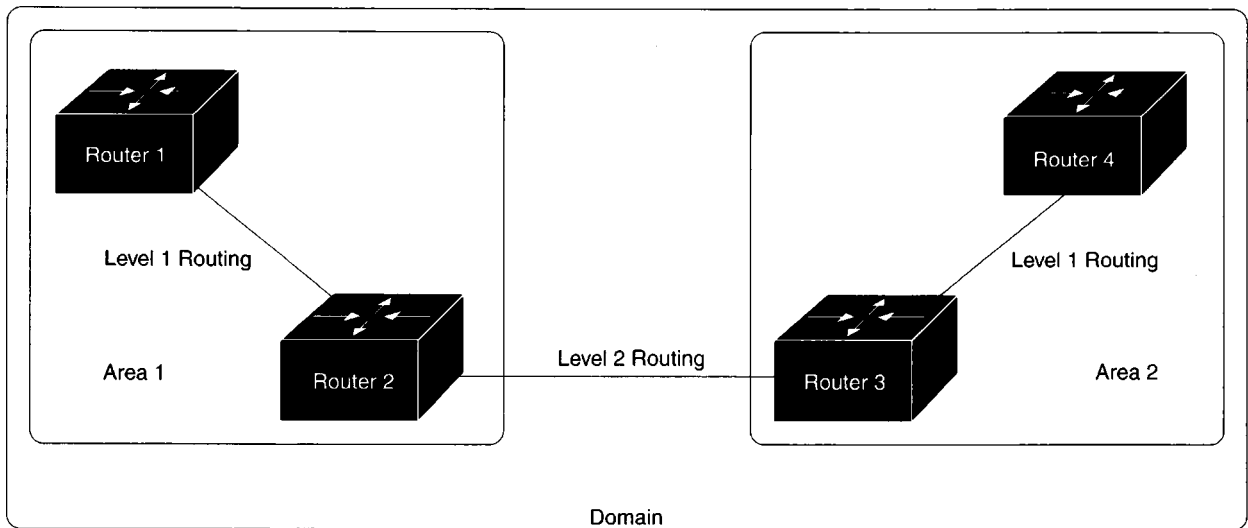


Figure 28-1 Hierarchies in OSI Internetworks

From a purely technological standpoint, IS-IS is quite similar to the *Open Shortest Path First (OSPF)* routing protocol (see Chapter 25, “OSPF,” for more information on OSPF). Both protocols are link-state protocols. Both offer a variety of features not provided by the *Routing Information Protocol (RIP)*, including *routing hierarchies*, *path splitting*, *type-of-service (TOS)* support, *authentication*, support for multiple network-layer protocols, and (with Integrated IS-IS) support for *variable length subnet masks*.

ES-IS

ES-IS is more a discovery protocol than a routing protocol. Through ES-IS, ESs and ISs learn about each other. This process is known as *configuration*. And, because configuration must happen before routing between ESs can occur, ES-IS is discussed here.

ES-IS distinguishes between three different types of subnetworks:

- *Point-to-point subnetworks*—Provide a point-to-point link between two systems. Many wide area network (WAN) serial links are point-to-point networks.
- *Broadcast subnetworks*—Direct a single physical message to all nodes on the subnetwork. Ethernet and IEEE 802.3 are examples of broadcast subnetworks. For more information about Ethernet and IEEE 802.3, see Chapter 5, “Ethernet/IEEE 802.3.”
- *General-topology subnetworks*—Support an arbitrary number of systems. However, unlike broadcast subnetworks, the cost of an n-way transmission scales directly with the subnetwork size on a general-topology subnetwork. X.25 is an example of a general-topology subnetwork. For more information about X.25, see Chapter 13, “X.25.”

Configuration information is transmitted at regular intervals through two types of messages. *ES hello messages (ESHs)* are generated by ESs and sent to every IS on the subnetwork. *IS hello messages (ISHs)* are generated by ISs and sent to all ESs on the subnetwork. These hello messages are primarily intended to convey the subnetwork and network-layer addresses of the systems that generate them.

Where possible, ES-IS attempts to send configuration information to many systems simultaneously. On broadcast subnetworks, ES-IS hello messages are sent to all ISs through a special multicast address. ISs send hello messages to a special multicast address designating all end systems. When operating on a general-topology subnetwork, ES-IS generally does not transmit configuration information, due to the high cost of multicast transmissions.

ES-IS conveys both network-layer addresses and subnetwork addresses. OSI network-layer addresses identify either the *network service access point (NSAP)*, which is the interface between Layer three and Layer four, or the *network entity title (NET)*, which is the network layer entity in an OSI IS. OSI subnetwork addresses (sometimes called *subnetwork point of attachment addresses*, or *SNPAs*) are the points at which an ES or IS is physically attached to a subnetwork. The SNPA address uniquely identifies each system attached to the subnetwork. In an Ethernet network, for example, the SNPA is the 48-bit *media access control (MAC)* address. Part of the configuration information transmitted by ES-IS is the NSAP-to-SNPA or NET-to-SNPA mapping.

Figure 28-2 shows the frame formats of both ESH and ISH packets.

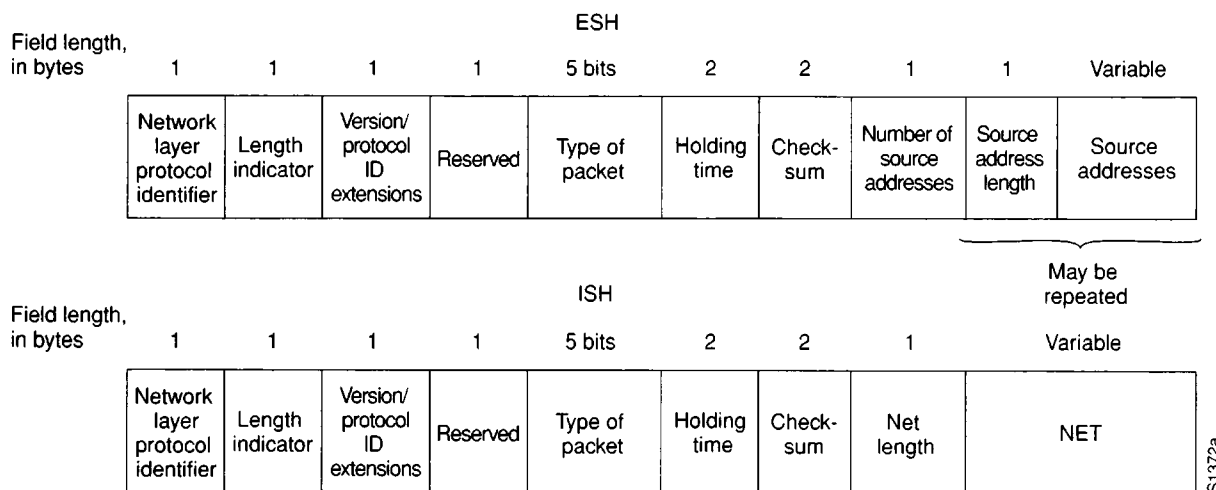


Figure 28-2 ESH and ISH Packet Formats

IS-IS

IS-IS is a link-state routing protocol. As such, it floods the network with link-state information in order to build a complete, consistent picture of network topology.

Routing Hierarchy

To simplify router design and operation, IS-IS distinguishes between level 1 and level 2 ISs. Level 1 ISs know how to communicate with other level 1 ISs in the same area. Level 2 ISs know how to communicate with ISs in other areas. To summarize, level 1 ISs form level 1 areas; level 2 ISs route between level 1 areas.

Level 2 ISs form an intradomain routing backbone. In other words, level 2 ISs can get to other level 2 ISs by traversing only level 2 ISs. The backbone simplifies design because level 1 ISs now only need to know how to get to the nearest level 2 IS. The backbone routing protocol can also change without impacting the intra-area routing protocol.

Inter-ES Communication

OSI routing is accomplished as follows. Each ES lives in a particular area. ESs discover the nearest IS by listening to ISH packets. When an ES wants to send a packet to another ES, it sends the packet to one of the ISs on its directly-attached network. The router looks up the destination address and forwards the packet along the best route. If the destination ES is on the same subnetwork, the local IS will know this from listening to ESHs, and will forward the packet appropriately. In this case, the IS may also provide a *redirect* (RD) message back to the source to tell it that a more direct route is available. If the destination address is an ES on another subnetwork in the same area, the IS will know the correct route, and will forward the packet appropriately. If the destination address is an ES in another area, the level 1 IS sends the packet to the nearest level 2 IS. Forwarding through level 2 ISs continues until the packet reaches a level 2 IS in the destination area. Within the destination area, ISs forward the packet along the best path until the destination ES is reached.

Each IS generates an update specifying the ESs and ISs to which it is connected, as well as the associated metrics. The update is sent to all neighboring ISs, which forward (flood) it to their neighbors, and so on. Sequence numbers terminate the flood and distinguish old updates from new ones. Because all ISs receive link-state updates from all other ISs, each IS can build a complete full topology database. When the topology changes, new updates are sent.

Metrics

IS-IS uses a single required default metric with a maximum path value of 1024. The metric is arbitrary and typically assigned by a network administrator. Any single link can have a maximum value of 64. Path lengths are calculated by summing link values. Maximum metric values were set at these levels to provide the granularity to support various link types, while at the same time ensuring that the shortest path algorithm used for route computation would be reasonably efficient.

IS-IS also defines three additional metrics (costs) as an option for those administrators who feel they are necessary. The *delay* cost reflects the amount of delay on the link. The *expense* cost reflects the communications cost associated with using the link. The *error* cost reflects the error rate of the link.

IS-IS maintains a mapping of these four metrics to the *quality of service* (QoS) option in the CLNP packet header. Using these mappings, IS-IS can compute routes through the internetwork.

Packet Format

IS-IS uses three basic packet formats:

- *IS-IS hello packets*
- *Link state packets (LSPs)*
- *Sequence numbers packets (SNPs)*

Each of the three IS-IS packets has a complex format with three different logical parts. The first part is an 8-byte fixed header shared by all three packet types. The second part is a packet-type-specific portion with a fixed format. The third logical part is also packet-type-specific, but is of variable length. The logical format of IS-IS packets is shown in Figure 28-3.

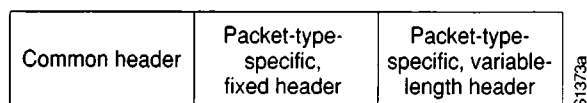


Figure 28-3 IS-IS Logical Packet Format

Each of the three packet types shares a common header, as shown in Figure 28-4.

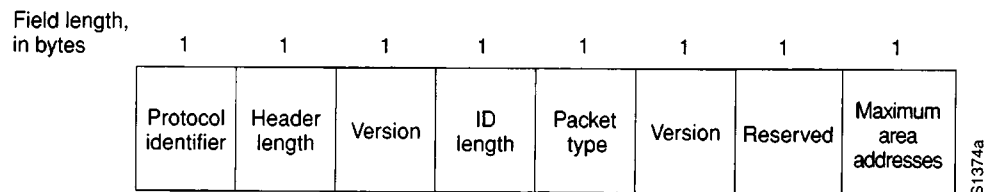


Figure 28-4 IS-IS Common Header Format

The first field in the IS-IS common header is the *protocol identifier*, which identifies the IS-IS protocol. This field contains a constant (131).

The next field in the common header is the *header length* field. This field contains the fixed header length. The length is always equal to 8 bytes, but is included so that IS-IS packets wouldn't differ significantly from CLNP packets.

Following the length field is the *version* field, which is equal to one in the current IS-IS specification.

After the version field is the *ID length* field, which specifies the size of the ID portion of an NSAP if between one and eight, inclusive. If the field contains a zero, the ID portion equals six bytes. If the field contains 255 (all ones), the ID portion is zero bytes.

Next, the *packet type* field specifies the type of IS-IS packet (hello, LSP, or SNP)

After the packet type field, the *version* field is repeated.

Following the second version field is a *reserved* field which is equal to zero and ignored by the receiver.

The final field in the common header is the *maximum area addresses* field. This field specifies the number of addresses permitted in this area.

Following the common header, each packet type has a different additional fixed portion, followed by a variable portion.

Integrated IS-IS

Integrated IS-IS is a version of IS-IS that uses a single routing algorithm to support more network-layer protocols than just CLNP. Integrated IS-IS is sometimes called *Dual IS-IS*, after a version designed for IP and CLNP networks.

Several fields are added to IS-IS packets to allow IS-IS to support additional network layers. These fields inform routers about:

- The reachability of network addresses from other protocol suites
- Which protocols are supported by which routers
- Other information required by a specific protocol suite

Integrated IS-IS represents one of two ways of supporting multiple network layer protocols in a router, the other being the *ships in the night* approach. Ships in the night advocates the use of a completely separate and distinct routing protocol for each network protocol, so that the multiple routing protocols essentially exist independently (with different types of routing information passing like ships in the night). The ability to route multiple network-layer protocols through tables calculated by a single routing protocol saves some router resources.

Inter-Domain Routing Protocol (IDRP)

IDRP is the OSI protocol designed to move information between routing domains. As such, it is designed to operate seamlessly with CLNP, ES-IS, and IS-IS. IDRP is based on the *Border Gateway Protocol (BGP)*, an interdomain routing protocol that originated in the IP community. For more information on BGP, see Chapter 27, “BGP.”

IDRP introduces several new terms, including:

- *Border intermediate system (BIS)*—An IS that participates in interdomain routing. As such, it uses IDRP.
- *Routing domain (RD)*—A group of ESs and ISs operating under the same set of administrative rules, including the sharing of a common routing plan.
- *Routing domain identifier (RDI)*—A unique routing domain (RD) identifier.
- *Routing information base (RIB)*—The routing database used by IDRP. RIBs are built by each BIS from information received from within the RD and from other BISs. A RIB contains the set of routes chosen for use by a particular BIS.
- *Confederation*. A group of routing domains (RDs). The confederation appears to RDs outside the confederation as a single RD. A confederation’s topology is not visible to RDs outside the confederation. Confederations help reduce network traffic by acting as internetwork firewalls, and may be nested within one another.

An IDRP route is a sequence of RDIs. Some of these RDIs can be confederations. Each BIS is configured to know the RD and confederations to which it belongs, and learns about other BISs, RDs, and confederations through information exchanges with each neighbor. As with distance-vector routing, routes to a particular destination accumulate outward from the destination. Only routes that satisfy a BIS’s local policies and have been selected for use will be passed on to other BISs. Route recalculation is partial and occurs when one of three events occurs: an incremental routing update with new routes is received, a BIS neighbor goes down, or a BIS neighbor comes up.

IDRP features include:

- Support for CLNP QOS
- Loop suppression, through the keeping track of all RDs traversed by a route
- Reduction of route information and processing, through the use of confederations, the compression of RD path information, and other means
- Reliability, through the use of a built-in reliable transport
- Security, through the use of cryptographic signatures on a per-packet basis
- Route servers
- RIB refresh packets

Part 6

**Bridging
Technologies**

Chapter 29

Transparent Bridging

Background

Transparent bridges (TBs) were first developed at Digital Equipment Corporation (Digital) in the early 1980s. Digital submitted its work to the Institute of Electrical and Electronic Engineers (IEEE), which incorporated the work into the IEEE 802.1 standard. TBs are very popular in Ethernet/IEEE 802.3 networks

Technology Basics

TBs are so named because their presence and operation is transparent to network hosts. Upon being powered on, TBs learn the network's topology by analyzing the source address of incoming frames from all attached networks. If, for example, a bridge sees a frame arrive on line 1 from Host A, the bridge concludes that Host A can be reached through the network connected to line 1. Through this process, TBs build a table such as the one in Figure 29-1.

Host address	Network number
15	1
17	1
12	2
13	2
18	1
9	1
14	3
.	.
.	.
.	.

S1375a

Figure 29-1 Transparent Bridging Table

The bridge uses its table as the basis for traffic forwarding. When a frame is received on one of the bridge's interfaces, the bridge looks up the frame's destination address in its internal table. If the table contains an association between the destination address and any of the bridge's ports aside from the one on which the frame was received, the frame is forwarded out the indicated port. If no association is found, the frame is flooded to all ports except the inbound port. Broadcasts and multicasts are also flooded in this way.

TBs successfully isolate intrasegment traffic, thereby reducing the traffic seen on each individual segment. This usually improves network response times as seen by the user. The extent to which traffic is reduced and response times are improved depends on the volume of intersegment traffic relative to the total traffic as well as the volume of broadcast and multicast traffic.

Bridging Loops

Without a bridge-to-bridge protocol, the TB algorithm fails when there are multiple paths of bridges and LANs between any two LANs in the internetwork. Figure 29-2 illustrates such a bridging loop.

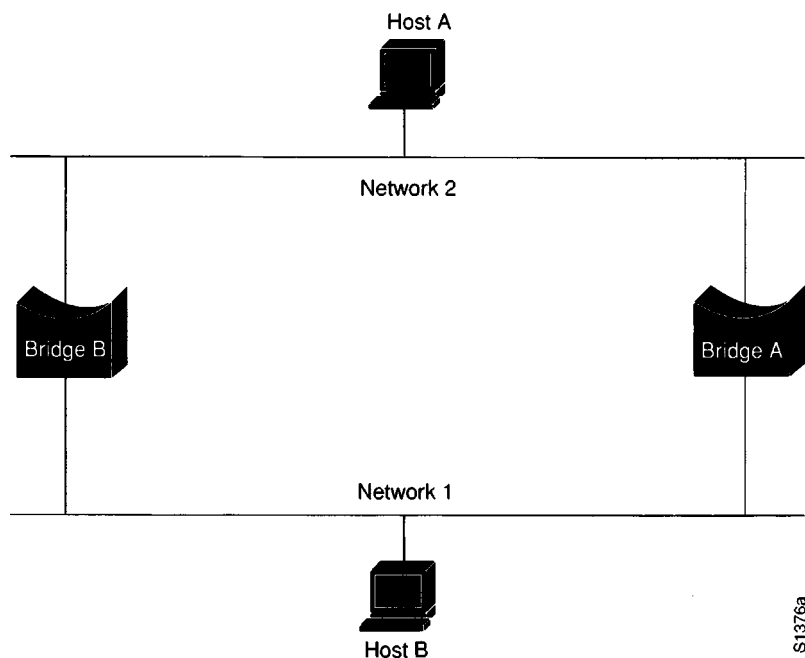


Figure 29-2 Inaccurate Forwarding and Learning in Transparent Bridging Environments

Suppose Host A sends a frame to Host B. Both bridges receive the frame and correctly conclude that Host A is on Network 2. Unfortunately, after Host B receives two copies of Host A's frame, both bridges will again receive the frame on their Network 1 interfaces because all hosts receive all messages on broadcast LANs. In some cases, the bridges will then change their internal tables to indicate that Host A is on Network 1. If so, when Host B replies to Host A's frame, both bridges will receive and subsequently drop the replies, since their tables will indicate that the destination (Host A) is on the same network segment as the frame's source.

In addition to basic connectivity problems such as the one just described, the proliferation of broadcast messages in networks with loops represents a potentially serious network problem. Referring again to Figure 29-2, assume that Host A's initial frame is a broadcast. Both bridges will forward the frames endlessly, using all available network bandwidth and blocking the transmission of other packets on both segments.

A topology with loops such as that pictured in Figure 29-2 can be useful as well as potentially harmful. A loop implies the existence of multiple paths through the internetwork. A network with multiple paths from source to destination can increase overall network fault tolerance through improved topological flexibility.

Spanning-Tree Algorithm (STA)

An algorithm was created to preserve the benefits of loops while eliminating their problems. This algorithm was initially documented by Digital, a key Ethernet vendor. Digital's new algorithm was subsequently revised by the IEEE 802 committee and published in the IEEE 802.1d specification as the *spanning-tree algorithm (STA)*.

The STA designates a loop-free subset of the network's topology by placing those bridge ports that, if active, would create loops into a standby (blocking) condition. Blocking bridge ports can be activated in the event of primary link failure, providing a new path through the internetwork.

The STA uses a conclusion from graph theory as a basis for constructing a loop-free subset of the network's topology. Graph theory states the following:

For any connected graph consisting of nodes and edges connecting pairs of nodes, there is a spanning tree of edges that maintains the connectivity of the graph but contains no loops.

Figure 29-3 illustrates how the STA eliminates loops. The STA calls for each bridge to be assigned a unique identifier. Typically, this identifier is one of the bridge's *Media Access Control (MAC)* addresses plus a priority. Each port in every bridge is also assigned a unique (within that bridge) identifier (typically, its own MAC address). Finally, each bridge port is associated with a path cost. The path cost represents the cost of transmitting a frame onto a LAN through that port. In Figure 29-3, path costs are noted on the lines emanating from each bridge. Path costs are usually defaulted, but can be assigned manually by network administrators.

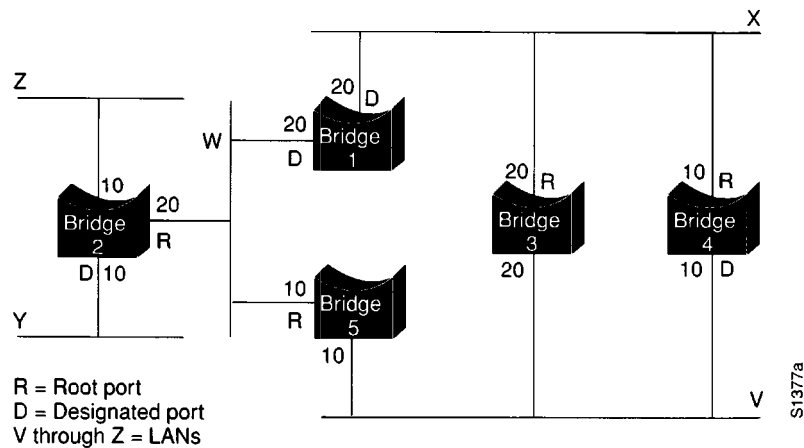


Figure 29-3 TB Network Before Running STA

The first activity in spanning-tree computation is the selection of the *root bridge*, which is the bridge with the lowest value bridge identifier. In Figure 29-3, the root bridge is Bridge 1. Next, the *root port* on all other bridges is determined. A bridge's root port is the port through which the root bridge may be reached with the least aggregate path cost. This value (the least aggregate path cost to the root) is called the *root path cost*.

Finally, *designated bridges* and their *designated ports* are determined. A designated bridge is the bridge on each LAN that provides the minimum root path cost. A LAN's designated bridge is the only bridge allowed to forward frames to and from the LAN for which it is the designated bridge. A LAN's designated port is that port which connects it to the designated bridge.

In some cases, two or more bridges may have the same root path cost. For example, in Figure 29-3, Bridges 4 and 5 can both reach Bridge 1 (the root bridge) with a path cost of 10. In this case, the bridge identifiers are used again, this time to determine the designated bridges. Bridge 4's LANV port is selected over Bridge 5's LANV port.

Using this process, all but one of the bridges directly connected to each LAN are eliminated, thereby removing all two-LAN loops. The STA also eliminates loops involving more than two LANs, while still preserving connectivity. Figure 29-4 shows the results of applying the STA to the network depicted in Figure 29-3. Figure 29-4 shows the tree topology more clearly. Comparing this figure to the prespanning-tree figure shows that the STA has placed both Bridge 3's and Bridge 5's ports to LANV into standby mode.

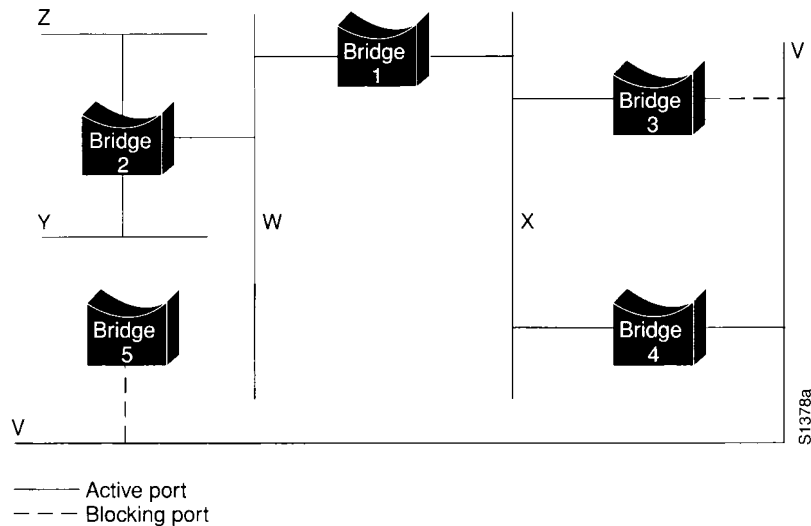


Figure 29-4 TB Network After Running STA

The spanning tree calculation occurs when the bridge is powered up and whenever a topology change is detected. The calculation requires communication between the spanning tree bridges, which is effected through configuration messages (sometimes called *bridge protocol data units*, or *BPDUs*). Configuration messages contain information identifying the bridge that is presumed to be the root (root identifier) and the distance from the sending bridge to the root bridge (root path cost). Configuration messages also contain the bridge and port identifier of the sending bridge and the age of information contained in the configuration message.

Bridges exchange configuration messages at regular intervals (typically 1 to 4 seconds). If a bridge fails (causing a topology change), neighboring bridges will soon detect the lack of configuration messages and initiate a spanning tree recalculation.

All TB topology decisions are made locally. Configuration messages are exchanged between neighbor bridges. There is no central authority on network topology or administration.

Frame Format

TB bridges exchange *configuration* messages and *topology change* messages. Configuration messages are sent between bridges to establish a network topology. Topology change messages are sent after a topology change has been detected, to indicate that the STA should be rerun.

The IEEE 802.1d configuration message format is shown in Figure 29-5.

Field length,
in bytes

	2	1	1	1	8	4	8	2	2	2	2	2
	Protocol identifier	Version	Message type	Flags	Root ID	Root path cost	Bridge ID	Port ID	Message age	Maximum age	Hello time	Forward delay

S1379a

Figure 29-5 TB Configuration Message Format

The first field in the TB configuration message is a *protocol identifier*, which contains the value zero.

The second field in the TB configuration frame is the *version*, which contains the value zero.

The third field in the TB configuration message is the *message type*, which contains the value zero.

The fourth field in the TB configuration message is a one-byte *flags* field. The TC bit signals a topology change. The TCA bit is set to acknowledge receipt of a configuration message with the TC bit set. The other six bits in this byte are unused.

The next field in the TB configuration message is the *root ID* field. This eight-byte field identifies the root bridge by listing its two-byte priority followed by its six-byte ID.

Following the root ID field is the *root path cost* field, which contains the cost of the path from the bridge sending the configuration message to the root bridge.

Next is the *bridge ID* field, which identifies the priority and ID of the bridge sending the message.

The *port ID* field identifies the port from which the configuration message was sent. This field allows loops created by multiply-attached bridges to be detected and dealt with.

The *message age* field specifies the amount of time since the root sent the configuration message upon which the current configuration message is based.

The *maximum age* field indicates when the current configuration message should be deleted.

The *hello time* field provides the time period between root bridge configuration messages.

Finally, the *forward delay* field provides the length of time that bridges should wait before transitioning to a new state after a topology change. If a bridge transitions too soon, not all network links may be ready to change their state, and loops can result.

Topological change messages consist of only 4 bytes. They include a *protocol identifier* field, which contains the value zero, a *version* field, which contains the value zero, and a *message type* field, which contains 128.

Chapter 30

Source-Route Bridging

30

Background

The Source-Route Bridging (SRB) algorithm was developed by IBM and proposed to the IEEE 802.1 committee as the means to bridge between all LANs. After the IEEE 802.1 committee decided that a competing standard (see Chapter 29, “Transparent Bridging,” for more information on the *Transparent Bridging (TB)* standard) would have this honor, SRB supporters proposed SRB to the IEEE 802.5 committee, which subsequently adopted SRB into the IEEE 802.5 Token Ring LAN specification.

Since its initial proposal, IBM has offered a new bridging standard to the IEEE 802 committee: the *Source-Route Transparent (SRT)* solution. See Chapter 31, “Mixed-Media Bridging,” for more information on SRT. SRT eliminates pure source-route bridges (SRBs) entirely, proposing that the two types of LAN bridges be TBs and SRTs. Although SRT has achieved support, SRBs are still widely deployed.

SRB Algorithm

SRBs are so named because they assume that the complete source-to-destination route is placed in all inter-LAN frames sent by the source. SRBs store and forward the frames as indicated by the route appearing in the appropriate frame field. Figure 30-1 illustrates a sample SRB network.

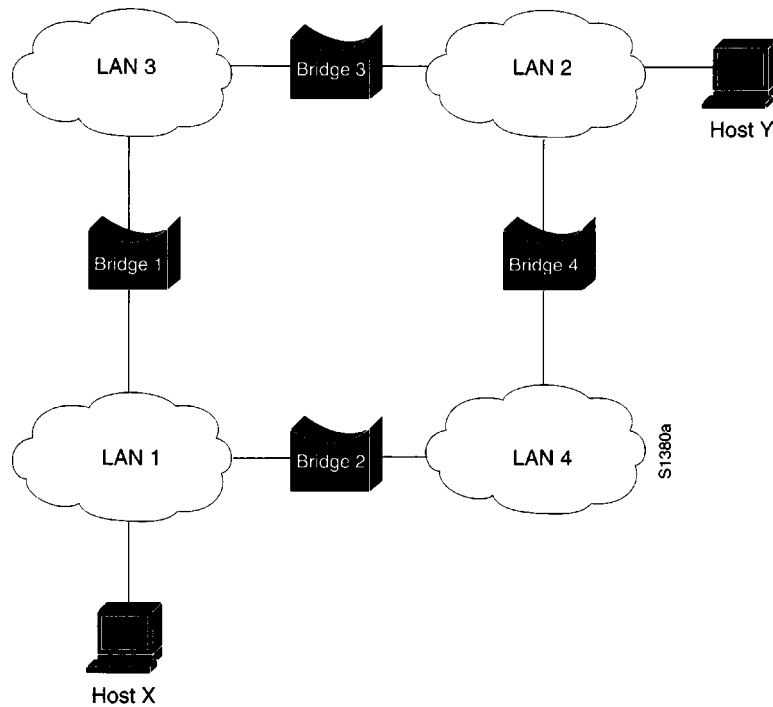


Figure 30-1 Sample SRB Network

Referring to Figure 30-1, assume that Host X wishes to send a frame to Host Y. Initially, Host X does not know whether Host Y resides on the same or on a different LAN. To determine this, Host X sends out a test frame. If that frame returns to Host X without a positive indication that Host Y has seen it, Host X must assume that Host Y is on a remote segment.

To determine the exact remote location of Host Y, Host X sends an *explorer* frame. Each bridge receiving the explorer frame (Bridges 1 and 2 in this example) copies the frame onto all outbound ports. Route information is added to the explorer frames as they travel through the internetwork. When Host X's explorer frames reach Host Y, Host Y replies to each individually using the accumulated route information. Upon receipt of all response frames, Host X may choose a path based on some predetermined criterion.

In the example in Figure 30-1, this process will yield two routes:

- LAN 1 - Bridge 1 - LAN 3 - Bridge 3 - LAN 2
- LAN 1 - Bridge 2 - LAN 4 - Bridge 4 - LAN 2

Host X must select one of these two routes. The IEEE 802.5 specification does not mandate the criteria Host X should use in choosing a route, but it does make several suggestions, including the following:

- First frame received
- Response with the minimum number of hops
- Response with the largest allowed frame size

- Various combinations of the above criteria

In most cases, the path contained in the first frame received will be used.

After a route is selected, it is inserted into frames destined for Host Y in the form of a *routing information field (RIF)*. A RIF is included only in those frames destined for other LANs. The presence of routing information within the frame is indicated by the setting of the most significant bit within the source address field, called the *routing information indicator (RII)* bit.

Frame Format

The IEEE 802.5 RIF is structured as shown in Figure 30-2.

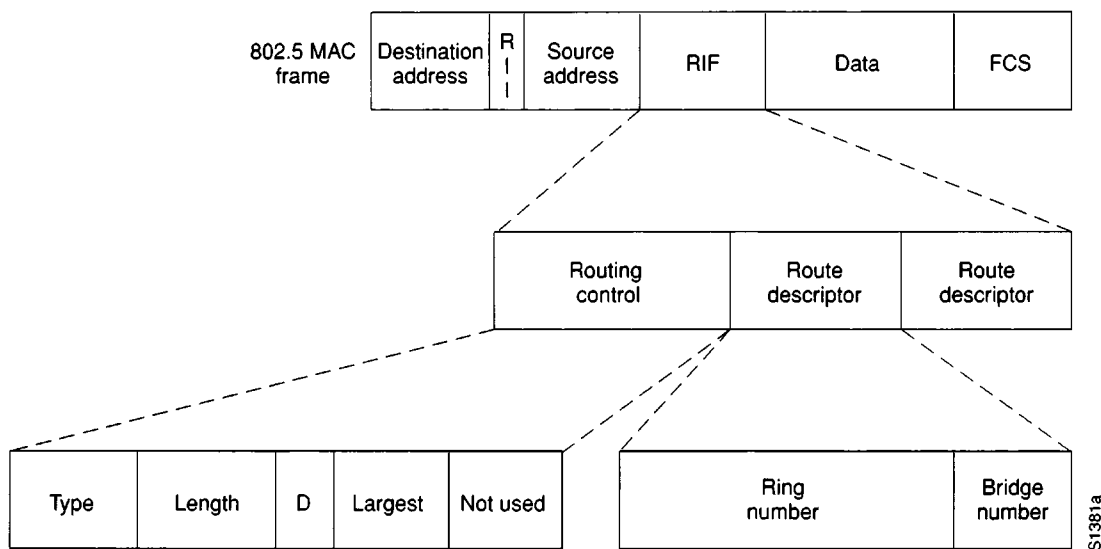


Figure 30-2 IEEE 802.5 RIF

The *type* subfield in the RIF indicates whether the frame should be routed to a single node, a group of nodes comprising a spanning tree of the internetwork, or all nodes. The first type is called a *specifically routed frame*, the second type is called a *spanning-tree explorer*, and the third type is called an *all-paths explorer*. The spanning tree explorer can be used as a transit mechanism for multicast frames. It can also be used as a replacement for the all-paths explorer in outbound route queries. In this case, the destination responds with an all-paths explorer.

The *length* subfield indicates the total length (in bytes) of the RIF.

The *D* bit indicates the direction of the frame (forward or reverse).

The *largest* field indicates the largest frame that can be handled along this route.

There can be multiple *route descriptor* fields. Each carries a ring number-bridge number pair that specifies a portion of a route. Routes, then, are simply alternating sequences of LAN and bridge numbers that start and end with LAN numbers.

Chapter 31

Mixed-Media Bridging

Background

Transparent bridges (TBs) are found predominantly in Ethernet networks, whereas source-route bridges (SRBs) are found almost exclusively in Token Ring networks. See Chapter 29, "Transparent Bridging," for more information on TB; see Chapter 30, "Source-Route Bridging," for more information on Source-Route Bridging. Both the TB and the SRB bridging methods are popular, so it is reasonable to ask whether a method exists to bridge between them. This basic question is illustrated in Figure 31-1.

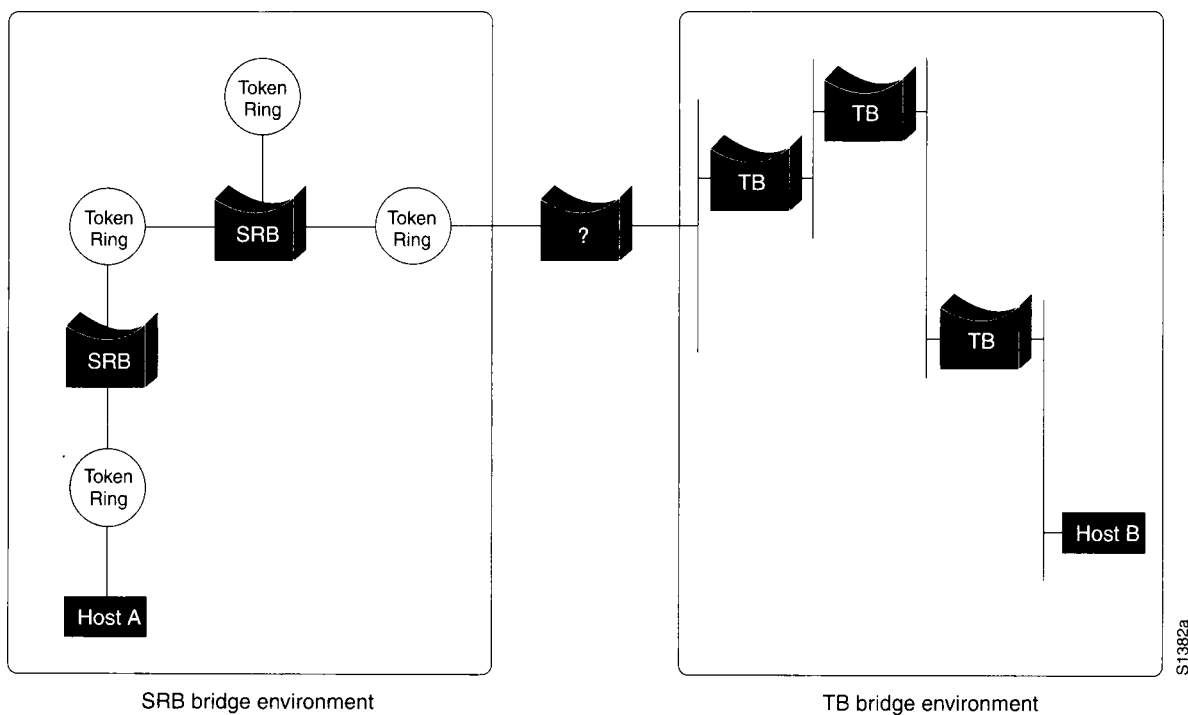


Figure 31-1 Bridging Between TB and SRB Domains

Technology Basics

Translational bridging (TLB) provides a relatively inexpensive solution to some of the many problems involved with bridging between TB and SRB domains. TLB first appeared in the mid- to late-1980s, but has not been championed by any standards organization. As a result, many aspects of TLB are left to the implementor.

In 1990, IBM addressed some of the weaknesses of TLB by introducing *Source-Route Transparent (SRT)* bridging. SRTs can forward traffic from both transparent and source route end nodes and form a common spanning tree with TBs, thereby allowing end stations of each type to communicate with end stations of the same type in a network of arbitrary topology.

Ultimately, the goal of connecting TB and SRB domains is to allow communication between TB and SRB end stations. This chapter describes the technical problems that must be addressed by algorithms attempting to do this and presents two possible solutions: TLB and SRT.

Translation Challenges

There are a number of challenges associated with allowing end stations from the Ethernet/TB domain to communicate with end stations from the SRB/Token Ring domain, including the following:

- **Incompatible bit ordering**—Although both Ethernet and Token Ring support 48-bit MAC addresses, the internal hardware representation of these addresses differs. In a serial bit stream representing an address, Token Ring considers the first bit encountered to be the high-order bit of a byte. Ethernet, on the other hand, considers the first bit encountered to be the low-order bit.
- **Embedded Media Access Control (MAC) addresses**—In some cases, MAC addresses are actually carried in the data portion of a frame. For example, the *Address Resolution Protocol (ARP)*, a popular protocol in *Transmission Control Protocol/Internet Protocol (TCP/IP)* networks, places hardware addresses in the data portion of a link-layer frame. Conversion of addresses that may or may not appear in the data portion of a frame is difficult because they must be handled on a case-by-case basis.
- **Incompatible maximum transfer unit (MTU) sizes**—Token Ring and Ethernet support different maximum frame sizes. Ethernet's MTU is approximately 1500 bytes, whereas Token Ring frames can be much larger. As bridges are not capable of frame fragmentation and reassembly, packets that exceed the MTU of a given network must be dropped.
- **Handling of frame status bit actions**—Token Ring frames include three frame status bits: A, C, and E. The purpose of these bits is to tell the frame's source whether the destination saw the frame (A bit set), copied the frame (C bit set) and/or found errors in the frame (E bit set). Since Ethernet does not support these bits, the question of how to deal with these bits is left to the Ethernet-Token Ring bridge manufacturer.

- Handling of exclusive Token Ring functions—Certain Token Ring bits have no corollary in Ethernet. For example, Ethernet has no priority mechanism, whereas Token Ring does. Other Token Ring bits that must be thrown out when a Token Ring frame is converted to an Ethernet frame include the token bit, the monitor bit, and the reservation bits.
- TB handling of explorer frames—TBs do not inherently understand what to do with SRB route discovery frames. TBs learn about the network's topology through analysis of the source address of incoming frames. They have no knowledge of the SRB route discovery process.
- TB handling of routing information field (RIF) information within Token Ring frames—The SRB algorithm places routing information in the RIF field. The TB algorithm has no RIF equivalent, and the idea of placing routing information in a frame is foreign to TB.
- Incompatible spanning-tree algorithms—TB and SRB both use the spanning-tree algorithm to try to avoid loops, but the particular algorithms employed by the two bridging methods are incompatible.
- SRB handling of frames without route information—SRB bridges expect all inter-LAN frames to contain route information. When a frame without a RIF field (including TB configuration and topology change messages and MAC frames sent from the TB domain) arrives at an SRB bridge, it is simply ignored.

Translational Bridging (TLB)

Because there has been no real standardization in how communication between two media types should occur, there is no single TLB implementation that can be called correct. The following describes several popular methods for implementing TLB.

TLBs reorder source and destination address bits when translating between Ethernet and Token Ring frame formats. The problem of embedded MAC addresses can be solved by programming the bridge to check for various types of MAC addresses, but this solution must be adapted with each new type of embedded MAC address. Some TLB solutions simply check for the most popular embedded addresses. If TLB software runs in a multiprotocol router, the router can successfully route these protocols and avoid the problem entirely.

The RIF field has a subfield that indicates the largest frame size that can be accepted by a particular SRB implementation. TLBs that send frames from the TB to the SRB domain will usually set this MTU size field to 1500, to limit the size of Token Ring frames entering the TB domain. Some hosts cannot correctly process this field, in which case TLBs are forced to simply drop those frames that exceed Ethernet's MTU size.

Bits representing Token Ring functions that have no Ethernet corollary are typically thrown out by TLBs. For example, Token Ring's priority, reservation, and monitor bits are discarded. As for Token Ring's frame status bits, these are treated differently depending on the TLB manufacturer. Some TLB manufacturers simply ignore the bits. Others have the bridge set the C bit but not the A bit. In the former case, there is no way for a Token Ring source node to determine whether the frame it sent has become lost. Proponents of this approach suggest that reliability mechanisms such as the tracking of lost frames are better left for implementation in layer 4 of the OSI model. Proponents of the "set the C bit approach" contend that this bit must be set to track lost frames, but that the A bit cannot be set because the bridge is not the final destination.

TLBs can create a software gateway between the two domains. To the SRB end stations, the TLB has a ring number and bridge number associated with it, and so looks like a standard SRB. The ring number, in this case, actually reflects the entire TB domain. To the TB domain, the TLB is simply another TB.

When bridging from the SRB domain to the TB domain, SRB information is removed. RIFs are usually cached for use by subsequent return traffic. When bridging from the TB to the SRB domain, the TLB can check the frame to see if it has a unicast destination. If the frame has a multicast or broadcast destination, it is sent into the SRB domain as a spanning tree explorer. If the frame has a unicast address, the TLB looks up the destination in the RIF cache. If a path is found, it is used and the RIF information is added to the frame; otherwise, the frame is sent as a spanning-tree explorer. Because the two spanning-tree implementations are not compatible, multiple paths between the SRB and the TB domains are typically not permitted.

Source-Route Transparent (SRT) Bridging

SRTs combine implementations of the TB and SRB algorithms. SRTs use the *routing information indicator (RII)* bit to distinguish between frames employing SRB and frames employing TB. If the RII bit is 1, a RIF is present in the frame, and the bridge uses the SRB algorithm. If the RII bit is 0, a RIF is not present, and the bridge uses TB.

Like TLBs, SRTs are not perfect solutions to the problems of mixed-media bridging. SRTs must still deal with the Ethernet/Token Ring incompatibilities described earlier. SRT is likely to require hardware upgrades to SRBs to allow them to handle the increased burden of analyzing every packet. Software upgrades to SRBs may also be required. Further, in environments of mixed SRTs, TBs, and SRBs, source routes chosen must traverse whatever SRTs and SRBs are available. The resulting paths can potentially be substantially inferior to spanning-tree paths created by TBs. Finally, mixed SRB/SRT networks lose the benefits of SRT, so users will feel compelled to execute a complete cutover to SRT at considerable expense. Still, SRT permits the coexistence of two incompatible environments and allows communication between SRB and TB end nodes.

Part 7

**Network
Management**



Background

Three development efforts impacted what was to become the Simple Network Management Protocol (SNMP):

- *High-level Entity Management System (HEMS)*—Specifies a management system with some interesting technical attributes. Unfortunately, HEMS was never used outside its development sites and this eventually led to its demise.
- *Simple Gateway Monitoring Protocol (SGMP)*—Initiated by a group of network engineers to address the problems associated with managing the rapidly-growing Internet (a large internetwork connecting government organizations, research institutions, universities, and commercial concerns), this effort produced a protocol designed to manage Internet gateways (routers). SGMP was implemented on many regional branches of the Internet.
- *CMIP over TCP (CMOT)*—Advocated *Open System Interconnection (OSI)*-based network management and specifically, the *Common Management Information Protocol (CMIP)* to help manage *Transmission Control Protocol (TCP)*-based internetworks.

Through the latter part of 1987, the merits and demerits of these three approaches (HEMS, SGMP, and CMOT) were frequently and hotly contested. In early 1988, an Internet Activities Board (IAB, a group responsible for technical development of the Internet protocols) committee was created to resolve the network management protocol debate. Eventually, the IAB committee agreed that an improved version of SGMP, to be called SNMP, would be the short-term solution, and some OSI-based technology (either CMOT or CMIP itself) would be analyzed for its long-term suitability. To ensure an easy upgrade path, a common framework for network management (now called the Internet-standard *Network Management Framework*) was created.

Today, SNMP is the most popular protocol for managing diverse commercial, university, and research internetworks. SNMP-related standardization activity continues even as vendors develop and release state-of-the-art SNMP-based management applications. SNMP is a relatively simple protocol, yet its feature set is sufficiently powerful to handle the difficult problems presented by management of heterogeneous networks.

Technology Basics

SNMP is an application-layer protocol designed to facilitate the exchange of management information between network devices. By using SNMP data (such as packets per second and network error rates), network administrators can more easily manage network performance and find and solve network problems.

Management Model

In SNMP, agents are software modules that run in managed devices. Agents compile information about the managed devices in which they run and make this information available to *network management systems (NMSs)* via a *network management protocol (SNMP)*. This model is graphically represented in Figure 32-1.

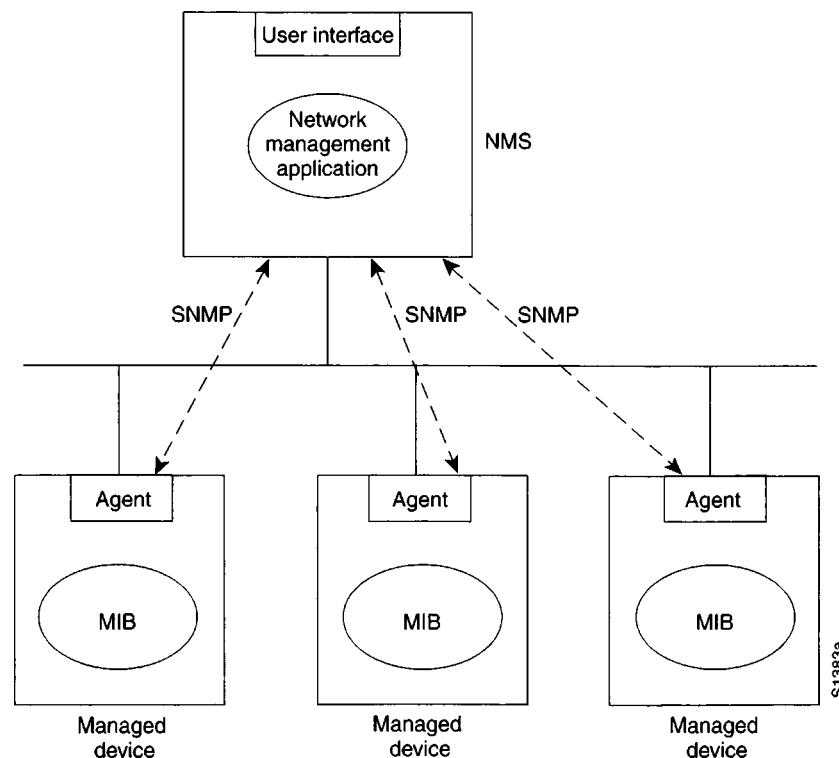


Figure 32-1 SNMP Management Model

A managed device can be any type of node residing on a network, including computer hosts, communication servers, printers, routers, bridges, and hubs. Because some of these systems may have limited ability to run management software (they may have relatively slow CPUs or limited memory, for example), management software must assume the lowest common denominator. In other words, management software must be built in such a way as to minimize its own performance impact on the managed device.

Because managed devices contain a lowest common denominator of management software, the management burden falls on the NMS. Therefore, NMSs are typically engineering workstation-calibre computers that have fast CPUs, megapixel color displays, substantial

memory, and lots of disk space. One or more NMSs may exist on any managed network. NMSs run the network management applications that present management information to users. The user interface is typically based on a standardized *graphical user interface (GUI)*.

Communication between managed devices and NMSs is governed by the network management protocol. The Internet-standard Network Management Framework assumes a remote-debugging paradigm, where managed devices maintain values for a number of variables and report those, on demand, to NMSs. For example, a managed device might keep track of the following:

- Number and state of its virtual circuits
- Number of certain kinds of error messages received
- Number of bytes and packets in and out of the device
- Maximum output queue length (for routers and other internetworking devices)
- Broadcast messages sent and received
- Network interfaces going down and coming up

Command Types

If a NMS wishes to control a managed device, it can do so by sending a message requiring the managed device to change the value of one of its variables. In total, managed devices respond to/initiate four different types of commands:

- *Reads*—To monitor managed devices, NMSs read variables maintained by the devices.
- *Writes*—To control managed devices, NMSs write variables stored within the managed devices.
- *Traversal operations*—NMSs use these to determine which variables a managed device supports and to sequentially gather information in variable tables (such as an IP routing table)
- *Traps*—Managed devices use traps to asynchronously report certain events to NMSs.

Data Representation Differences

Exchange of information in a managed network is potentially compromised by differences in the data representation techniques used by the managed devices. In other words, computers represent information differently; these incompatibilities must be rationalized to allow communication between diverse systems. An abstract syntax performs this function. SNMP uses a subset of *Abstract Syntax Notation One (ASN.1)*, an abstract syntax created for OSI, for this purpose. ASN.1 defines both packet formats and managed objects. A managed object is simply a characteristic of something that can be managed. A managed object differs from a variable, which is a particular object instance. Managed objects may be scalar (defining a single instance) or tabular (defining multiple, related instances).

Management Database

All managed objects are contained in the *Management Information Base (MIB)*, which is essentially a database of objects. Logically, a MIB is depicted as an abstract tree, with individual data items as leaves. Object identifiers uniquely identify MIB objects in the tree. Object identifiers are like phone numbers, in that they are organized hierarchically and that parts are assigned by different organizations. For example, international phone numbers consist of country codes (assigned by an international organization) and the telephone number, as defined by that country. United States phone numbers are further subdivided into an area code, a central office (CO) number, and a station number associated with that CO. Similarly, top-level MIB object identifiers are assigned by the International Organization for Standardization/International Electrotechnical Commission (ISO IEC). Lower-level object IDs are allocated by the associated organizations. The root and several prominent branches of the MIB tree appear in Figure 32-2.

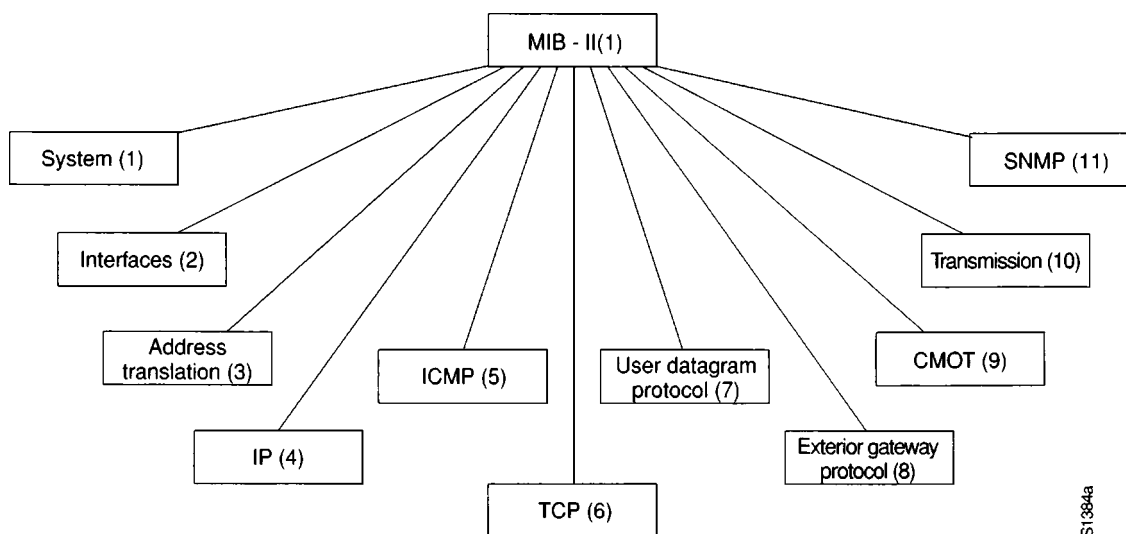


Figure 32-2 MIB Tree

The MIB tree is extensible by virtue of experimental and private branches. Vendors, for example, can define their own private branches to include instances of their own products. Currently, all standardization work is in the experimental branch.

A document called the *Structure of Management Information (SMI)* defines the structure of the MIB. The SMI defines the following data types:

- *Network addresses*—Represent an address from a particular protocol family. Currently, 32-bit IP addresses is the only instance of network addresses.
- *Counters*—Nonnegative integers that increase monotonically until they reach a maximum value, when they wrap back to zero. The total number of bytes received on an interface is an example of a counter.
- *Gauges*—Nonnegative integers that can increase or decrease, but latch at a maximum value. The length of an output packet queue (in packets) is an example of a gauge.

- *Ticks*—Hundredths of a second since some event. The time since an interface entered its current state is an example of a tick.
- *Opaque*—An arbitrary encoding. This is used to pass arbitrary information strings outside of the strict data typing used by the SMI.

Operations

SNMP itself is a simple request/response protocol. Nodes can send multiple requests without a response. Four SNMP operations are defined:

- *Get*—Retrieves an object instance from the agent.
- *Get-next*—A traversal operation that retrieves the next object instance from a table or list within an agent.
- *Set*—Sets object instances within an agent.
- *Trap*—Used by the agent to asynchronously inform the NMS of some event.

Message Format

SNMP messages contain two parts: a *community name* and *data*. The community name assigns an access environment for a set of NMSs using that community name. NMSs within the community can be said to exist within the same administrative domain. Because devices that do not know the proper community name are precluded from SNMP operations, network managers have also used the community name as a weak form of authentication.

The data portion of an SNMP message contains the specified SNMP operation (get, set, and so on) and associated operands. Operands indicate the object instances involved in the SNMP transaction.

SNMP messages are formally referred to as *protocol data units (PDUs)*. Figure 32-3 shows the SNMP packet format.

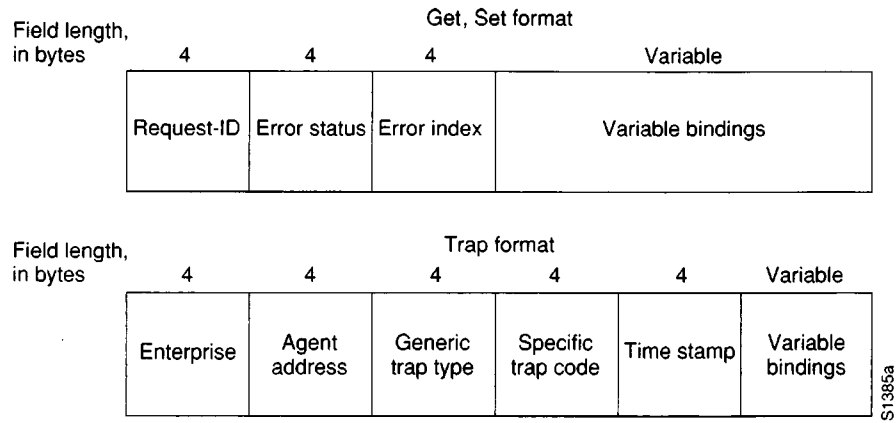


Figure 32-3 SNMP Message Format

SNMP get and set PDUs consist of the following parts:

- *Request-ID*—Associates requests with responses.
- *Error-status*—Indicates an error and an error type.
- *Error-index*—Associates the error with a particular object instance.
- *Variable bindings*—Comprise the data of an SNMP PDU. Variable bindings associate particular variables with their current values.

Trap PDUs are slightly different than PDUs for other operations. They consist of the following parts:

- *Enterprise*—Identifies the type of object generating the trap.
- *Agent address*—Provides the address of the object generating the trap.
- *Generic trap type*—Provides the generic trap type.
- *Specific trap code*—Provides the specific trap code.
- *Time stamp*—Provides the amount of time that has elapsed between the last network re-initialization and generation of this trap.
- *Variable bindings*—Provides a list of variables containing interesting information about the trap.

Chapter 33

IBM Network Management

33

Background

IBM was one of the first companies to recognize the importance of a complete, integrated network management strategy. In 1986, IBM proposed *Open Network Management (ONA)*, a framework describing a generalized network management architecture. *NetView*, the premier product for network management on an IBM mainframe, is actually a component of ONA. NetView provides a cohesive set of centralized network management services that allow users to monitor, control, and reconfigure their *Systems Network Architecture (SNA)* networks.

Since the introduction of ONA and NetView, IBM has almost continually enhanced, expanded, and otherwise altered its network management technology base. Today, IBM network management is comprehensive and extremely complex. The following sections describe the high-level basics of some of the components of IBM network management.

Functional Areas of Management

IBM divides network management into five user-based functions:

- *Configuration management*—Identifies physical and logical system resources and allows control of their relationships.
- *Performance and accounting management*—Allows quantification, measurement, reporting, and control of the responsiveness, availability, utilization, and usage of a network component.
- *Problem management*—Provides problem detection, diagnosis, resolution, and tracking and control capabilities.
- *Operations management*—Provides the means to query and control distributed network resources from a central site.
- *Change management*—Allows planning, control, and application of additions, deletions, and modifications to system hardware, microcode, and software.

These network management functions do not correlate perfectly with those proposed by the International Organization for Standardization (ISO) in its *Open Systems Interconnection (OSI)* model. The OSI and the IBM network management functions are compared in Figure 33-1.

OSI	IBM
Configuration management	Configuration management
Performance management	Performance and accounting management
Accounting management	
Fault management	Problem management
Security management	-
-	Operations management
-	Change management

S1386a

Figure 33-1 OSI and IBM Network Management Functions

Configuration Management

Configuration management controls information describing both physical and logical information systems resources and their relationships to each other. This information typically consists of resource names, addresses, locations, contacts, and phone numbers. IBM's configuration management function corresponds very closely to OSI's concept of configuration management.

Through configuration management facilities, users can maintain an inventory of network resources. Configuration management helps ensure that network configuration changes can be reflected expeditiously and accurately in the configuration management database. Configuration management data is used by problem management systems to compare version differences and to locate, identify, and check the characteristics of network resources. Change management systems can use configuration management data to analyze the effect of changes and to schedule changes at times of minimal network impact.

An SNA management service called *query product identification* retrieves software and hardware physical information from the configuration management database. The information retrieved is sometimes called *vital product data*.

Performance and Accounting Management

This SNA management function provides information about the performance of network resources. Through analysis of performance and accounting management data, users can determine whether network performance goals are being met.

Performance and accounting management includes accounting, monitoring of response times, availability, utilization, and component delay, as well as performance tuning, tracking, and control. Data from each of these functions can result in the initiation of problem determination procedures if performance levels are not being met.

Problem Management

SNA management services defines a *problem* as an error condition that causes a user to lose full functionality of a system resource. SNA divides problem management into several areas:

- Problem determination—Detects a problem and completes steps necessary for problem diagnosis to begin. Problem determination intends to isolate the problem to a particular subsystem, such as a hardware device, a software product, a microcode component, or a media segment.
- Problem diagnosis—Determines the precise cause of a problem and the action required to solve the problem. If problem diagnosis is done manually, it follows problem determination. If it is done automatically, it is usually done simultaneously with problem determination so that the results can be reported together.
- Problem bypass and recovery—Attempts to bypass a problem, either partially or completely. Normally, this operation is temporary, with the intent that complete problem resolution will follow, but problem bypass may be permanent when the problem is less-easily resolved.
- Problem resolution—Involves efforts required to eliminate the problem. Problem resolution usually begins after problem diagnosis is complete and often involves a corrective action that must be scheduled, such as replacement of a failed disk drive.
- Problem tracking and control—Tracks the problem until final resolution. Specifically, if external action is required to fix the problem, the vital information describing the problem (such as status monitoring data and problem status reports) are included in a problem management record which is entered into the problem database

Operations Management

Operations management involves management of distributed network resources from a central site. It entails two sets of functions: *common operations services* and *operations management services*.

Common operations services allow management of resources not explicitly addressed by SNA's other management categories by allowing specialized communication with these resources through new, more capable applications. Two very important services providing this capability are the **execute** command and the resource management service. The **execute** command provides a standardized means of executing a remote command. Resource management services provide a way to transport information in a context-independent manner.

Operations management services provide the ability to control remote resources through resource activation, resource deactivation, command cancellation, and the setting of network resource clocks. Operations management services can be initiated automatically as a result of system problem notification forwarding, thereby allowing automatic handling of remote problems.

Change Management

Change management helps users control network or system changes by allowing the sending, retrieving, installing, and removing of change files at remote nodes. Further, change management allows node activation. Changes occur because either user requirements have changed, or because a problem must be circumvented.

Although problems cause change, change can also cause problems. Change management attempts to minimize problems created by change through encouraging orderly change and by tracking changes.

Principal Management Architectures and Platforms

IBM offers several management architectures and many important management platforms. This section discusses these architectures and platforms.

The Open Network Management (ONA) Framework

The basic ONA framework is shown in Figure 33-2.

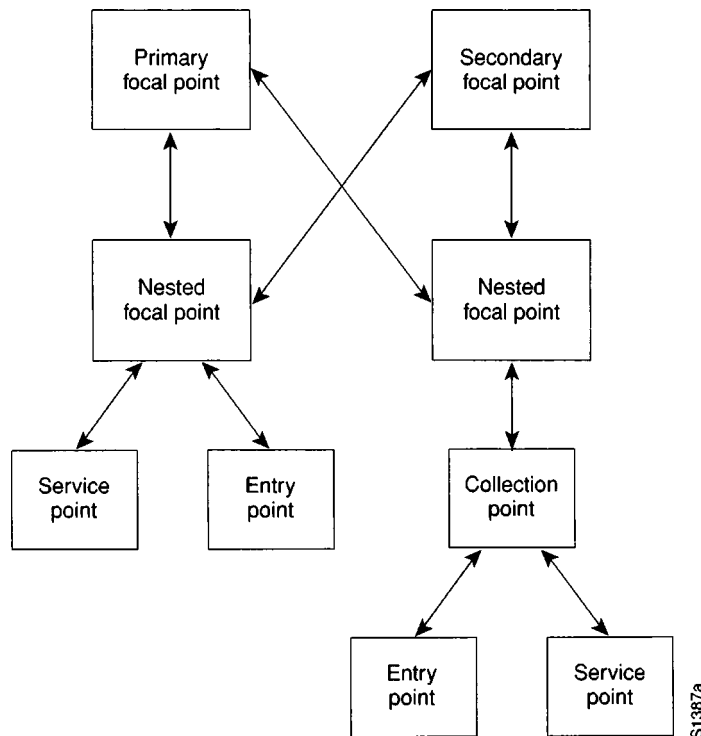


Figure 33-2 ONA Framework

Focal points provide support for centralized network management operations. They are the management entities referred to in the general model described previously. Focal points respond to end station alerts, maintain management databases, and provide the user interface to the network management operator.

There are two kinds of focal points: *primary* and *secondary*. Primary focal points are as described previously. Secondary focal points provide backup for primary focal points, and are used when primary focal points fail.

Nested focal points provide distributed management support for portions of large networks. They forward critical information to more global focal points.

Collection points relay information from self-contained SNA subnetworks to focal points. Collection points are commonly used to forward data from IBM peer-to-peer networks into the ONA hierarchy.

Entry points are SNA devices that can implement ONA for themselves and other devices. Most standard SNA devices are capable of being entry points.

Service points are systems that provide access into ONA for non-SNA devices. Service points are capable of sending network management information about non-SNA systems to focal points and are also capable of receiving commands from focal points, translating them into a format acceptable to non-SNA devices, and forwarding them to non-SNA devices for execution. Service points are essentially gateways into ONA.

SystemView

IBM announced SystemView in 1990. SystemView is a blueprint for the creation of management applications capable of managing multivendor information systems. Specifically, SystemView describes how applications that manage heterogeneous networks will look, feel, and cooperate with other management systems. Officially, SystemView is *Systems Application Architecture's* (SAA's) systems management strategy.

NetView

NetView is IBM's most comprehensive enterprise network management platform. It has the following major parts:

- *Command control facility*—Provides the ability to control the network through basic operator and file access commands to *Virtual Telecommunications Access Method (VTAM)* applications, controllers, operating systems, and *NetView/PC* (an interface between NetView and non-SNA devices).
- *Hardware monitor*—Monitors the network and automatically issues alerts to the network operator when a hardware error occurs.
- *Session monitor*—Acts as a VTAM performance monitor. The session monitor provides software problem determination and configuration management.

- *Help function*—Provides help for NetView management services users. The help function includes a browse facility, a help desk facility, and a library of commonly encountered network operation situations.
- *Status monitor*—Summarizes and presents network status information.
- *Performance monitor*—Monitors the performance of *communications controllers* (also called *front-end processors*, or *FEPs*), the *Network Control Program (NCP)*, and attached resources.
- *Distribution manager*—Plans, schedules, and tracks the distribution of data, software, and 3174 microcode in an SNA environment.

LAN Network Manager

IBM's *LAN Network Manager (LNM)* product is an OS/2 Extended Edition-based network management application that enables control of Token Ring local area networks (LANs) from a central support site. NetView can see LNM activity (for example, alarms). LNM communicates with *LAN Station Manager (LSM)* software, which implements management agents in individual LAN end stations. Communication between LNM and LSM is effected using OSI Common Management Information Services/Common Management Information Protocol (CMIS/CMIP) running over the connectionless Logical Link Control (LLC) protocol.

SNMP

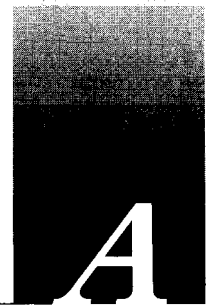
IBM has recently added support for the *Simple Network Management Protocol (SNMP)*. See Chapter 32, "SNMP," for information on this protocol.

Appendix



Appendix A

References and Recommended Reading



Banyan Systems, Inc. *VINES Protocol Definition*, DA254-00, Rev. 1.0. Westboro, MA, February 1990.

Black, U. *Data Networks: Concepts, Theory, and Practice*. Englewood Cliffs, NJ: Prentice-Hall, Inc.; 1989.

Case, J., et al. "A Simple Network Management Protocol." *RFC 1157*, DDN Network Information Center, Chantilly, VA; May 1990.

Cisco Systems, Inc. *Internetworking Terms and Acronyms*, DOC-TA13, Rev. 2.0. Menlo Park, CA, January 1993.

Cisco Systems, Inc. *Router Products Configuration and Reference*, Vols. 1-3, DOC-R9.1, Rev. 9.1. Menlo Park, CA, October 1992.

Clark, W. "SNA Internetworking." *ConneXions: The Interoperability Report*, Vol. 6, No. 3: March 1992.

Coltun, R. "OSPF: An Internet Routing Protocol." *ConneXions: The Interoperability Report*, Vol. 3, No. 8: August 1989.

Comer, D. *Internetworking with TCP/IP: Principles, Protocols, and Architecture*. Englewood Cliffs, NJ: Prentice Hall, Inc.; 1988.

Digital Equipment Corporation, Intel Corporation, Xerox Corporation. *The Ethernet, A Local-Area Network, Data Link Layer and Physical Layer Specifications*, Version 2.0, November, 1982.

Feinler, Elizabeth J., et al. *DDN Protocol Handbook*, Vols. 1-4, NIC 50004, 50005, 50006, 50007. Defense Communications Agency, Alexandria, VA, December 1985.

Green, J.H. *Telecommunications*, 2nd edition. Homewood, IL: BUSINESS ONE IRWIN; 1992.

Hagans, R. "Components of OSI: ES-IS Routing." *ConneXions: The Interoperability Report*, Vol. 3, No. 8: August 1989.

Hares, S. "Components of OSI: Inter-Domain Routing Protocol (IDRP)." *ConneXions: The Interoperability Report*, Vol. 6, No. 5: May 1992.

Hedrick, C.L. "An Introduction to IGRP." Unpublished paper, October 23, 1989.

Hedrick, C. "Routing Information Protocol." *RFC 1058*, DDN Network Information Center, Chantilly, VA; June 1988.

- Hewlett-Packard Company. *X.25: The PSN Connection*, PN 5958-3402, 2nd ed., Eybens, FRANCE, October 1985.
- IBM Corporation. *APPN Architecture and Product Implementations Tutorial*. Document Number GG24-3669, 1st ed., Raleigh, NC, May 1991.
- IBM Corporation. *Systems Network Architecture Management Services Reference*. Document Number SC30-3346-4, 5th ed., Research Triangle Park, NC, December 1991.
- Joyce, S.T. and Walker II, J.Q. "Advanced Peer-to-Peer Networking (APPN): An Overview." *ConneXions: The Interoperability Report*, Vol. 6, No. 10: October 1992.
- Kousky, K. "Bridging the Network Gap." *LAN Technology*, Vol. 6, No. 1: January 1990.
- Leinwand, A. and Fang, K. *Network Management: A Practical Perspective*. Reading, MA: Addison-Wesley Publishing Company; 1993.
- Lippis, N. "The Internetwork Decade." *Data Communications*, Vol. 20, No. 14: October, 1991.
- Lougheed, K. and Rekhter, Y. "A Border Gateway Protocol (BGP)." *RFC 1163*, DDN Network Information Center, Chantilly, VA; June 1990.
- Martin, J. *SNA: IBM's Networking Solution*. Englewood Cliffs, NJ: Prentice Hall, Inc.; 1987.
- Medin, M. "The Great IGP Debate—Part Two: The Open Shortest Path First (OSPF) Routing Protocol." *ConneXions: The Interoperability Report*, Vol. 5, No. 10: October 1991.
- Mills, D.L. "Exterior Gateway Protocol Formal Specification." *RFC 904*, DDN Network Information Center, Chantilly, VA; April 1984.
- Moy, J. "OSPF Specification." *RFC 1131*, DDN Network Information Center, Chantilly, VA; October 1989.
- Perlman, R. *Interconnections: Bridges and Routers*. Reading, MA: Addison-Wesley Publishing Company; 1992.
- Perlman, R. and Callon, R. "The Great IGP Debate—Part One: IS-IS and Integrated Routing." *ConneXions: The Interoperability Report*, Vol. 5, No. 10: October 1991.
- Rose, M. T. *The Open Book: A Practical Perspective on OSI*. Englewood Cliffs, NJ: Prentice Hall, Inc.; 1990.
- Rose, M. T. *The Simple Book: An Introduction to Management of TCP/IP-based Internets*. Englewood Cliffs, NJ: Prentice Hall, Inc.; 1991.
- Ross, F. E. "FDDI—A Tutorial." *IEEE Communications Magazine*, Vol. 24, No. 5: May 1986.
- Sherman, K. *Data Communications: A User's Guide*. Englewood Cliffs, NJ: Prentice Hall, Inc.; 1990.
- Spragins, J.D. et. al. *Telecommunications Protocols and Design*. Reading, MA: Addison-Wesley Publishing Company; 1991.
- Stallings, W. *Handbook of Computer-Communications Standards*, Vols. 1-3. Carmel, IN: Howard W. Sams, Inc.; 1990.

- Stallings, W. *Local Networks*, 3rd edition. New York, NY: Macmillan Publishing Company; 1990.
- Sunshine, C. A. (ed.). *Computer Network Architectures and Protocols*, 2nd edition. New York, NY: Plenum Press; 1989.
- Tanenbaum, A. S. *Computer Networks*, 2nd edition. Englewood Cliffs, NJ: Prentice Hall, Inc.; 1988.
- Terplan, K. *Communication Networks Management*. Englewood Cliffs, NJ: Prentice-Hall, Inc.; 1992.
- Tsuchiya, P. "Components of OSI: IS-IS Intra-Domain Routing." *ConneXions: The Interoperability Report*, Vol. 3, No. 8: August 1989.
- Tsuchiya, P. "Components of OSI: Routing (An Overview)." *ConneXions: The Interoperability Report*, Vol. 3, No. 8: August 1989.
- Ultra Network Technologies. *Product Overview*, PN 06-0017-001, Rev. B, San Jose, CA, 1992.
- Xerox Corporation. *Internet Transport Protocols*, XNSS 028112, Sunnyvale, CA, December, 1981.
- Zimmerman, H. "OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection." *IEEE Transactions on Communications* COM-28, No. 4: April 1980.

Index



Index

Numerics

- 10Base2, 5-2
- 10Base5, 5-2
- 10BaseT, 5-2
- 10Broad36, 5-2
- 1Base5, 5-2, 8-3
- 3Com
 - adopted XNS, 22-1
 - and RIP, 23-1

A

- AARP, 16-4
- ABM, 12-5
- Abstract Syntax Notation 1
 - See ASN.1
- access DQDB, 15-3
- ACSE, 20-6
- active link, 7-7
- active monitor, 6-4
- Address Resolution Protocol
 - IP, 18-6
 - VINES, 21-7
- Address Resolution Protocol (VINES)
 - See ARP (VINES)
- address spoofing, 15-3
- addressing
 - general description, 1-6
 - link-layer addresses, 1-6
 - network-layer addresses, 1-6
- adjacent routers, 25-4
- ADSP, 16-8
- AEP, 16-8
- AFI, 20-4
- all-paths explorer, 30-3
- American National Standards Institute
 - See ANSI
- ANSI
 - and Frame Relay, 14-1
 - and HSSI, 9-1
 - and IS-IS, 28-1

- general description, 1-8
- ANSI X3T9.5 3, 7-1
- Apple Computer, 16-1
- AppleTalk
 - and the OSI reference model, 16-2
 - attention packet, 16-8
 - extended network, 16-5
 - general operation, 16-1
 - media access protocols, 16-2
 - network-layer protocols, 16-3
 - nonextended network, 16-5
 - Phase I, 16-1
 - Phase II, 16-1
 - transport-layer protocols, 16-8
- AppleTalk Address Resolution Protocol
 - See AARP
- AppleTalk Echo Protocol
 - See AEP
- application service element
 - See ASE
- ARCnet
 - as a NetWare media, 19-2
- area
 - definition, 2-7
 - OSI, 20-1, 28-1
 - OSPF, 25-2
- area border router, 25-2
- ARM, 12-5
- ARP (IP), 18-6, 31-2
- ARP (VINES), 21-2, 21-7
- ARPANET, 25-1
- AS
 - definition, 2-7
 - in BGP, 27-1
 - in EGP, 26-1
 - in IP, 18-6
 - in OSPF, 25-2
- ASE, 20-6
- ASN.1, 20-6, 32-3
- Association Control Service Element
 - See ACSE

asynchronous balanced mode
 See ABM

asynchronous response mode
 See ARM

Asynchronous Transfer Mode
 See ATM

at-least-once transaction, 16-8

ATM, 15-9

attachment unit interface, 5-3

AUI, 5-3

authentication
 in OSI routing, 28-2

authority and format identifier
 See AFI

autonomous system
 See AS

autoreconfiguration, 6-4

B

backbone
 OSPF, 25-2

backoff algorithm, 5-1

Banyan
 and RIP, 23-1
 and VINES, 21-1

Basic Rate Interface
 see BRI

BBN
 and OSPF, 25-1
 and the Internet protocols, 18-1

beaconing, 6-4

Bellcore
 and SMDS, 15-1
 technical advisories, 15-6

Bellman-Ford routing algorithm, 2-8

Berkeley Standard Distribution
 See BSD

BGP
 as the basis for IDRP, 28-7
 frame formats, 27-2
 general operation, 27-1
 messages
 keepalive, 27-3
 notification, 27-4
 open, 27-3
 update, 27-3

BIS, 28-7

BISDN
 aligned with SIP, 15-3
 as an example of a WAN technology, 9-1

black hole, 14-2

Bolt, Beranek, and Newman
 See BBN

border intermediate system, 28-7

BPDU, 29-5

BRI, 11-3

bridge
 basic definition, 3-1

bridge protocol data unit
 See BPDU

bridging
 general operation, 3-1
 mixed media, 31-1
 Source-Route Bridging, 30-1
 translational bridging, 31-2
 Transparent Bridging, 29-1

Broadband Integrated Services Digital Network
 See BISDN

broadcast
 directed, 22-3
 global, 22-4

broadcast networks, 5-1

BSD, 8-3, 23-1

Btrieve, 19-5

C

carrier sense multiple access/collision detection
 See CSMA/CD

CCITT
 and HSSI, 9-1
 and OSI, 20-7
 and SDLC, 12-1
 and X.25, 13-1
 general description, 1-8

I.430, 11-3

I.431, 11-3

I.450, 11-5

I.451, 11-5

Q.920, 11-5

Q.921, 11-5

Q.930, 11-5

Q.931, 11-5

cell, 15-7
cell relay, 15-9
change management, 33-1, 33-4
CI bus, 17-2
circuits
 permanent virtual, 13-3
 switched virtual, 13-3
 virtual, 13-3, 14-1
Cisco Systems
 and UltraNet, 8-4
 customer support, xxiv
 defining HSSI, 9-1
 forming a Frame Relay consortium, 14-1
 inventing IGRP, 24-1
 maintenance agreements, xxiii
 warranty information, xxiii
Class A network, 18-4
Class B network, 18-4
Class C network, 18-4
Class D network, 18-4
Class E network, 18-4
Clearinghouse protocol, 22-5
client, 16-1, 19-1, 21-2
CLNP, 17-2, 28-1
CLNS, 17-2, 20-3
cluster controller, 12-3
CMIP
 and CMOT, 32-1
 and LAN Network Manager, 33-6
 definition, 20-7
CMIS, 33-6
collection point, 33-5
Common Management Information Protocol
 See CMIP
Common Management Information Services
 See CMIS
common operations services, 33-3
communications controller
 See also FEP
community name, 32-5
Computer-room Interconnect Bus
 See CI bus
concentrator, 7-3
confederation
 and IDR, 28-7
configuration management (IBM), 33-1
configuration message, 29-5

confirmation (OSI service primitive), 20-3
Connectionless Network Protocol
 See CLNP
Connectionless Network Service
 See CLNS
CONS
 and OSI, 20-2
Consultative Committee for International Telegraph
 and Telephone
 See CCITT
convergence, 2-5
core router, 26-1
cost
 VINES, 21-4
counter
 and SNMP, 32-4
country code, 15-6
count-to-infinity problem, 23-4
Courier protocol, 22-5
CPE, 15-1
CSMA/CD, 5-1
customer support (Cisco Systems), xxiv

D

DARPA, 18-1, 22-1
DAS, 7-3
data circuit-terminating equipment
 See DCE
data link connection identifier
 See DLCI
data network identification code
 See DNIC
data terminal equipment
 See DTE
Datagram Delivery Protocol
 See DDP
Datapoint Corporation
 and ARCnet, 19-2
DCE, 13-1, 14-1
DDCMP, 12-1, 17-2
DDP, 16-5
DECnet
 as the basis for IS-IS, 28-1
 general operation, 17-1
 media-access protocols, 17-2
 network-layer protocols, 17-2

Phase IV routing, 17-3
Phase IV routing layer frame format, 17-3
upper-layer protocols, 17-6
DECnet Phase V, 17-1
DECnet/OSI, 17-1
Defense Advanced Research Projects Agency
See DARPA
Department of Defense, 22-1
designated bridge, 29-4
designated port, 29-4
designated router, 25-4
destination service access point
See DSAP
Digital Data Communications Message Protocol
See DDCMP
Digital Equipment Corporation
and DDCMP, 12-1
and DECnet, 17-1
and Frame Relay, 14-1
and IS-IS, 28-1
Digital Network Architecture
See DNA
Digital Signal 1
See DS-1
Digital Signal 3
See DS-3
Dijkstra algorithm, 25-1
directed broadcasts, 22-4
directed multicasts, 22-3
Directory Services
See DS (OSI)
discovery protocol, 28-2
distance-vector routing algorithm, 2-8
distance-vector routing protocol, 24-1, 25-2
distributed architecture
advantages of, 16-1
Distributed Queue Dual Bus
See DQDB
DLCI, 14-3
DNA
and the OSI reference model, 17-1
general description, 17-1
DNIC, 13-4
domain
definition, 2-7
OSI, 20-1, 28-1
OSPF, 25-2

domain specific part
See DSP
DQDB, 15-1
DS (OSI), 20-7
DS-1, 15-1, 15-8
DS-3, 15-1, 15-8
DSAP, 12-6
DSP, 20-4
DTE, 13-1, 14-1
DTE/DCE interface, 13-5
and Frame Relay, 14-1
and HSSI, 9-1
and LAPB, 12-5
and PPP, 10-2
and X.25, 13-1
dual homing, 7-7
Dual IS-IS
See Integrated IS-IS
dual-attached station
See DAS
dynamic routing, 2-6, 18-7

E

early token release, 6-2
Echo Protocol, 22-5
EGP
frame formats, 26-2
general operation, 26-1
messages
error, 26-4
neighbor acquisition, 26-3
neighbor reachability, 26-3
poll, 26-3
routing update, 26-4
EIA
general description, 1-8
Electronic Industries Association
See EIA
end system
See ES
End System-to-Intermediate System
See ES-IS
entry point, 33-5
EP, 22-4
error message, 26-4

-
- Error Protocol
 - See EP
 - ES, 2-3, 20-1
 - ES hello frame format, 28-3
 - ES-IS
 - general operation, 28-2
 - subnetwork types, 28-2
 - Ethernet
 - and Token Ring MTU, 31-2
 - and Transparent Bridging, 29-1
 - as a DECnet media, 17-2
 - as a NetWare media, 19-2
 - as an AppleTalk media, 16-2
 - as an example of a broadcast subnetwork, 28-2
 - as an example of a LAN, 10-1
 - differences between IEEE 802.3 and, 5-2
 - frame format, 5-3
 - general operation, 5-1
 - EtherTalk, 16-2
 - exactly-once transaction, 16-8
 - extended network, 16-5
 - exterior gateway protocol, 25-4, 27-1
 - exterior router, 18-6
 - External Data Representation
 - See XDR
- F**
- fast packet switching, 15-9
 - FCC, 9-3, 13-1
 - FDDI
 - as a DECnet media, 17-2
 - as a NetWare media, 19-2
 - as an AppleTalk media, 16-2
 - as an OSI media, 20-2
 - concentrator, 7-3
 - fault tolerant features, 7-4
 - frame format, 7-8
 - general operation, 7-1
 - link types
 - active, 7-7
 - passive, 7-7
 - priority system, 7-4
 - similarities to Token Ring, 7-1
 - specifications
 - Media Access Control (MAC), 7-2
 - Physical Layer Medium Dependent (PMD), 7-2
 - Physical Layer Protocol (PHY), 7-2
 - Station Management (SMT), 7-2
 - traffic types, 7-4
 - FDDITalk, 16-2
 - Federal Communications Commission
 - See FCC
 - FEP, 12-3
 - fiber
 - multimode, 7-2
 - single mode, 7-2
 - Fiber Distributed Data Interface
 - See FDDI
 - File Transfer Protocol
 - See FTP
 - File Transfer, Access, and Management
 - See FTAM
 - Filing protocol, 22-5
 - focal point
 - nested, 33-5
 - primary, 33-5
 - secondary, 33-5
 - frame buffer, 8-4
 - frame format
 - BGP, 27-2
 - DECnet Phase IV routing layer, 17-3
 - EGP, 26-2
 - ES hello, 28-3
 - Ethernet, 5-3
 - FDDI, 7-8
 - Frame Relay, 14-3
 - IDP, 22-3
 - IEEE 802.3, 5-3
 - IGRP, 24-2
 - IP, 18-3
 - IPX, 19-3
 - IS hello, 28-3
 - IS-IS, 28-5
 - OSI address, 20-5
 - OSPF, 25-5
 - RIF, 30-3
 - RIP, 23-3
 - SDLC, 12-2
 - SIP Level 2, 15-7
 - SIP Level 3, 15-6
 - SNMP, 32-5

TCP, 18-9
Token Ring, 6-5
Transparent Bridging, 29-5
VIP, 21-5
X.25, 13-4
Frame Relay
 frame format, 14-3
 LMI extensions, 14-2
 LMI message format, 14-5
 network implementation, 14-5
frames, 1-7
FTAM, 20-7
FTP, 18-10

G

gateway
 basic definition, 3-1
 in IP community, 18-6
gauge
 and SNMP, 32-4
get operation (SNMP), 32-5
get-next operation (SNMP), 32-5
global broadcasts, 22-4
global multicasts, 22-3
graphical user interface, 32-3
GUI, 32-3

H

hardware addresses
 See link-layer addresses
HDLC
 and PPP, 10-1
 and VINES, 21-2
 differences between SDLC and, 12-4
 general operation, 12-4
 originated from SDLC, 12-1
header, 1-3
hello packet, 21-4
Hello protocol, 25-4
HEMS, 32-1
hierarchical routing, 2-6
High-level Data Link Control
 See HDLC
High-level Entity Management System
 See HEMS

High-Performance Parallel Interface
 See HIPPI
HIPPI, 9-2
holddown, 23-4, 24-4
hop count, 2-8
HSSI
 and the OSI reference model, 9-2
 general operation, 9-1
 loopback tests, 9-3

I

IAB
 and SNMP, 32-1
 general description, 1-8
IBM
 and LU 6.2, 19-4
 and NetBIOS, 19-5
 and network management, 33-1
 and SDLC, 12-1
 and SRB, 30-1
 and SRT, 31-2
IBM network management
 definition of, 33-1
 functional areas of, 33-1
 principle architectures and platforms, 33-4
IBM Token Ring Network
 differences between IEEE 802.5 and, 6-1
ICMP, 18-7
ICP (VINES), 21-7
IDI, 20-4
IDN, 13-4
IDP (OSI), 20-4
IDP (XNS), 22-3
IDRP, 28-1, 28-7
IEEE
 and transparent bridging, 29-1
 divided data-link layer into sublayers, 3-4
 general description, 1-8
 modified HDLC to create IEEE 802.2, 12-1
IEEE 802.1 committee, 30-1
IEEE 802.2
 and SDLC, 12-1
 as a DECnet protocol, 17-2
 as an OSI protocol, 20-2
 general operation, 12-5

-
- IEEE 802.3
 - and FDDI, 7-1
 - and Transparent Bridging, 29-1
 - as a NetWare media, 19-2
 - as an example of a broadcast subnetwork, 28-2
 - as an OSI media, 20-2
 - differences between Ethernet and, 5-2
 - frame format, 5-3
 - general operation, 5-1
 - IEEE 802.5
 - and FDDI, 7-1
 - as a NetWare media, 19-2
 - as an OSI media, 20-2
 - differences between IBM Token Ring Network and, 6-1
 - general operation, 6-1
 - IEEE 802.6
 - as the basis for the SIP, 15-1
 - MAC protocol, 15-8
 - IETF, 25-1
 - IGP
 - definition, 18-6
 - IGRP, 24-1
 - OSPF, 25-1
 - RIP, 19-4, 22-4
 - IGRP
 - frame format, 24-2
 - general operation, 24-1
 - stability features, 24-4
 - timers
 - flush, 24-5
 - hold time, 24-5
 - invalid, 24-5
 - update, 24-5
 - indication (OSI service primitive), 20-3
 - information (I) frames, 12-3
 - initial domain identifier
 - See IDI
 - initial domain part
 - See IDP (OSI)
 - Institute of Electrical and Electronic Engineers
 - See IEEE
 - Integrated IS-IS, 28-1
 - Integrated Services Digital Network
 - See ISDN
 - inter-AS routing protocol, 27-1
 - interdomain IS, 2-3
 - Inter-Domain Routing Protocol
 - See IDRP
 - interior gateway protocol
 - See IGP
 - Interior Gateway Routing Protocol
 - See IGRP
 - interior router, 18-6
 - intermediate system
 - See IS
 - Intermediate System-to-Intermediate System Intra-Domain Routing Exchange Protocol
 - See IS-IS
 - internal organization of the network layer
 - See IONL
 - International Data Number
 - See IDN
 - International Telecommunications Union
 - See ITU
 - Internet, 25-1
 - Internet Activities Board
 - See IAB
 - Internet community, 19-4, 22-4
 - Internet Control Message Protocol
 - See ICMP
 - Internet Control Protocol
 - See ICP (VINES)
 - Internet Datagram Protocol
 - See IDP (XNS)
 - Internet Engineering Task Force
 - See IETF
 - Internet Packet Exchange
 - See IPX
 - Internet Protocol
 - See IP
 - Internet protocols
 - and the OSI reference model, 18-2
 - definition of, 18-1
 - transport layer, 18-8
 - upper-layer protocols, 18-10
 - intradomain IS, 2-3
 - IONL, 20-2
 - IP
 - and IGRP, 24-1
 - and Integrated IS-IS, 28-1
 - and NetWare, 19-5
 - and OSPF, 25-1
 - and XNS, 22-1

- definition of, 18-1
- frame format, 18-3
- network classes, 18-4
- subnet masks, 25-6
- TOS bits, 25-6

IPX

- and XNS, 22-1
- frame format, 19-3
- general operation, 19-3

IS, 2-3, 20-1, 28-1

IS hello frame format, 28-3

ISDN

- and Frame Relay, 14-2
- Layer 1, 11-3
- Layer 2, 11-4
- Layer 3, 11-5
- reference points, 11-2

IS-IS

- and OSPF, 25-1
- frame formats, 28-5
- general operation, 28-1

ISO

- and FDDI, 7-1
- and HSSI, 9-1
- and IBM's network management areas, 33-1
- and PPP, 10-2
- and routing protocols, 28-1
- and SDLC, 12-1
- and the OSI reference model, 20-1
- general description, 1-8
- network device names, 2-3
- network management (functional areas)
 - accounting management, 4-2
 - configuration management, 4-2
 - fault management, 4-2
 - performance management, 4-2
 - security management, 4-2

ISO 10589, 28-1

ISO 10747, 28-1

ISO 3309-1979, 10-2

ISO 3309-1984/PDAD1, 10-2

ISO 8348, 20-2

ISO 8648, 20-2

ISO 9542, 28-1

ISO/IEC, 32-4

ITP, 23-1

ITU, 13-1

L

LAN Manager, 19-1

LAN Network Manager

- See LNM

LAN Station Manager

- See LSM

LAP, 12-1

LAPB

- and SDLC, 12-1
- compared to HDLC and SDLC, 12-5
- frame types
 - information frames, 13-5
 - supervisory frames, 13-6
 - unnumbered frames, 13-6
- general operation, 13-5

LCP, 10-1, 10-3

LCP packet types

- link establishment, 10-4
- link maintenance, 10-4
- link termination, 10-4

LEDs, 7-2

Level 1 router, 17-5

Level 1 routing, 28-1

Level 2 router, 17-5

Level 2 routing, 28-1

light-emitting diodes

- See LEDs

Link Access Procedure

- See LAP

Link Access Procedure, Balanced

- See LAPB

link adapter, 8-4

Link Control Protocol

- See LCP

link-layer addresses, 1-6

link-state advertisement

- See LSA

link-state packet

- See LSP

link-state routing algorithm

- definition, 2-8
- vs. distance-vector algorithm, 24-1

LLC

- and LAN Network Manager, 33-6
- classes
 - Class I, 12-6

- Class II, 12-6
- Class III, 12-6
- Class IV, 12-6
- general operation, 12-5
- sublayer, 3-4
- types
 - Type 1, 12-5
 - Type 2, 12-5
 - Type 3, 12-5
- LLC2, 12-5
- LMI
 - extensions, 14-2
 - message format, 14-5
- LNM, 33-6
- local management interface
 - See LMI
- LocalTalk, 16-2
- Logical Link Control
 - See LLC
- LSA, 25-2
- LSM, 33-6
- LSP, 28-5

M

- MAC
 - address, 28-3
 - sublayer, 3-4
- Macintosh, 16-1
- MAC-layer bridge, 3-4
- maintenance agreements (Cisco Systems), xxiii
- management information base
 - See MIB
- MAU, 5-3
- Media Access Control sublayer
 - See MAC
- medium attachment unit
 - See MAU
- Message Handling System
 - See MHS
- messages, 1-7
- metrics
 - and IGRP, 24-2
 - and RIP, 23-1
 - BGP, 27-2
 - definition, 2-1, 2-8
 - EGP, 26-4

- general description, 2-8
- IS-IS, 28-5
- OSPF, 25-6
- MHS, 20-7
- MIB, 32-4
- mixed-media bridging, 31-1
- modal dispersion, 7-2
- monomode, 7-2
- MSAU, 6-1
- multi-access network, 25-4
- multicast
 - directed, 22-3
 - global, 22-4
- multi-CPE configuration, 15-4
- multimode, 7-2
- multipath routing
 - in IGRP, 24-2
 - in OSPF, 25-6
- multistation access unit
 - See MSAU

N

- Name Binding Protocol
 - See NBP
- NANP, 15-6
- national terminal number
 - See NTN
- NAU, 19-4
- NBP, 16-7
- NCP (IBM), 33-6
- NCP (NetWare), 19-3, 19-5
- NCP (PPP), 10-1
- neighbor, 25-4, 26-2
- neighbor acquisition message, 26-3
- neighbor reachability message, 26-3
- NET, 20-4, 28-3
- NetBIOS, 19-5
- NetView, 33-1, 33-5
- NetWare
 - and the OSI reference model, 19-2
 - general operation, 19-1
 - media-access protocols, 19-2
 - network-layer protocols, 19-3
 - shell, 19-5
 - transport-layer protocols, 19-5
 - upper-layer protocols, 19-5

NetWare Core Protocol
 See NCP (NetWare)

NetWare Loadable Module
 See NLM

NetWare Message Handling System
 See NetWare MHS

NetWare MHS, 19-5, 20-7

NetWare Remote Procedure Call
 See NetWare RPC

NetWare RPC, 19-5

network (AppleTalk), 16-4

network addressable unit
 See NAU

Network Basic Input/Output System
 See NetBIOS

Network Control Program
 See NCP (IBM)

Network Control Protocol
 See NCP (PPP)

network entry title
 See NET

Network File System
 See NFS

Network Information Center
 See NIC

network management
 ISO functional areas
 accounting management, 4-2
 configuration management, 4-2
 fault management, 4-2
 performance management, 4-2
 security management, 4-2

network management bus
 See NMB

network management framework, 32-1, 32-3

network management station, 4-2, 32-2

network operating system
 See NOS

network service access point
 See NSAP

Network Services Protocol
 See NSP

network-visible entity
 See NVE

NFS, 18-10, 19-1, 20-7

NIC, 24-1

NLM, 19-5

NMB, 8-3

node (AppleTalk), 16-4

normal response mode
 See NRM

North American numbering plan
 See NANP

Northern Telecom
 and Frame Relay, 14-1

NOS, 19-1

Novell
 adopted XNS, 22-1
 and NetWare, 19-1
 and RIP, 23-1

NRM, 12-5

NSAP, 20-4, 28-3

NSAP address, 20-4

n-selector, 20-4

NSP, 17-6

NT1, 11-2

NT1/2, 11-2

NT2, 11-2

NTN, 13-4

NVE, 16-7

O

OC-1, 9-2

Office Channel 1
 See OC-1

ONA, 33-1, 33-4

opaque
 and SNMP, 32-5

open network management
 See ONA

Open Shortest Path First
 See OSPF

Open Systems Interconnection
 See OSI

operations management, 33-1, 33-3

operations management services, 33-3

OSI
 address format, 20-5
 and IBM network management, 33-1
 media-access protocols, 20-2
 network service definition, 20-2
 network-layer protocols, 17-2, 20-2
 routing protocols, 28-1

- service primitives
 - confirmation, 20-3
 - indication, 20-3
 - request, 20-3
 - response, 20-3
- transport-layer protocols, 20-5
- upper-layer protocols, 20-6
- OSI reference model, 13-2
 - compatibility issues, 1-4
 - information formats, 1-3
 - layers
 - application layer, 1-5
 - link layer, 1-6
 - network layer, 1-6
 - physical layer, 1-6
 - presentation layer, 1-5
 - session layer, 1-5
 - transport layer, 1-5
 - service interface, 1-2
- OSPF
 - and IS-IS, 28-2
 - database description packet, 25-5
 - frame formats, 25-5
 - general operation, 25-1
 - link-state acknowledgement packet, 25-5
 - link-state request packet, 25-5
 - link-state update packet, 25-5
 - routing hierarchy, 25-2

P

- packet assembler/disassembler
 - See PAD
- Packet Exchange Protocol
 - See PEP
- packets, 1-7
- packet-switched network
 - See PSN
- packet-switching exchange
 - See PSE
- PAD, 13-2
- paddlecard, 8-4
- PARC
 - See Xerox PARC
- passive link, 7-7
- path splitting, 28-2
- PDN, 13-1

- PDU, 1-7
- PEP, 22-4
- performance and accounting management, 33-1, 33-2
- permanent virtual circuit
 - See PVC
- personality module, 8-4
- Phase IV routing, 17-3
- physical addresses
 - See link-layer addresses
- Physical Layer Convergence Protocol
 - See PLCP
- PLCP, 15-8
- Point-to-Point Protocol
 - See PPP
- poison reverse update, 23-4, 24-4
- poll message, 26-3
- PPP
 - and NetWare, 19-2
 - data link layer, 10-2
 - frame format, 10-2
 - general operation, 10-1
 - Link Control Protocol, 10-3
 - Network Control Protocols, 10-1
 - physical layer requirements, 10-2
- PRI, 11-3
- Primary Rate Interface
 - See PRI
- Printing protocol, 22-5
- problem management, 33-1, 33-3
- protocol data unit
 - See PDU
- protocol processor board, 8-4
- PSE, 13-1
- PSN, 13-1
- public data network
 - See PDN
- PVC, 13-3, 14-2

Q

- quality of service (QOS), 28-5

R

- RARP, 18-6
- RD, 28-7

RDI, 28-7
redirect message, 28-4
Reliable Transfer Service Element
 See RTSE
Remote Operations Service Element
 See ROSE
remote procedure call, 19-1, 20-6
Remote Procedure Call (NFS)
 See RPC
repeater
 basic definition, 3-1
request (OSI service primitive), 20-3
Requests For Comments
 See RFC
response (OSI service primitive), 20-3
Reverse Address Resolution Protocol
 See RARP
RFC, 18-2, 23-1, 27-1
RFC (assigned numbers), 10-3
RFC 1058, 23-2
RFC 1131, 25-1
RFC 904, 26-1
RIB, 28-7
RIF
 general operation, 30-3
RII, 30-3
RIP
 and IGRP, 24-1
 and IS-IS, 28-2
 and NetWare, 19-4
 and OSPF, 25-1
 and UltraNet, 8-1
 and VINES, 21-2
 and XNS, 22-1, 22-4
 frame format, 23-2
 general operation, 23-1
 stability features, 23-4
root bridge, 29-4
root path cost, 29-4
root port, 29-4
ROSE, 20-6
route flush timer, 23-3
route invalid timer, 23-3
routed protocols
 AppleTalk, 16-1
 DECnet, 17-1
 IP, 18-2
 IPX, 19-3
 OSI, 20-2
 VINES, 21-6
 vs. routing protocols, 2-9
 XNS, 22-1
router
 basic definition, 3-1
 exterior, 18-6
 interior, 18-6
router of last resort, 2-6
routing
 algorithm design goals, 2-4
 definition of, 2-1
 metrics, 2-8
 multipath, 24-2, 25-6
 update message, 2-2, 26-4
 update timer, 23-3
routing algorithm types, 2-6
routing algorithms
 Bellman-Ford, 2-8
 distance-vector, 2-7
 link-state, 2-7
 shortest path first, 2-7
routing area, 2-4
routing domain, 2-3, 28-7
routing domain identifier
 See RDI
routing hierarchy, 28-2
routing information base
 See RIB
routing information field
 See RIF
routing information indicator
 See RII
Routing Information Protocol
 See RIP
routing loop, 23-4, 24-5
routing metrics
 See metrics
routing protocols
 BGP, 27-1
 EGP, 26-1
 IGRP, 24-1
 OSPF, 25-2
 RIP, 23-1
 RTMP, 16-6
 vs. routed protocols, 2-9

routing table, 2-1
Routing Table Maintenance Protocol
 See RTMP
Routing Update Protocol
 See RTP
RPC, 18-10
RS-232, 9-1, 10-2, 13-6
RS-422, 10-2
RS-423, 10-2
RTMP, 16-6, 23-1
RTP
 general operation, 21-2
 packet types, 21-6
RTSE, 20-6

S

SAP (NetWare), 19-4
SAP (OSI), 1-3, 12-6
SAP (XNS), 22-1
SAS, 7-3
SCSI-2
 and HSSI, 9-3
SDLC
 configurations, 12-2
 frame format, 12-2
 frame types
 information, 12-3
 supervisory, 12-3
 unnumbered, 12-3
 general operation, 12-1
 node types, 12-1
SDU, 15-5
sequence number packet
 See SNP
Sequenced Packet Exchange
 See SPX
server, 16-1, 19-1, 21-2
service access point, 1-3, 12-6
Service Access Protocol
 See SAP (XNS)
Service Advertisement Protocol
 See SAP (NetWare)
service data unit
 See SDU
service node, 21-2
service point, 33-5
set operation (SNMP), 32-5
SGMP, 32-1
shortest-path-first routing algorithm, 2-7, 25-1
Simple Gateway Monitoring Protocol
 See SGMP
Simple Mail Transfer Protocol
 See SMTP
Simple Network Management Protocol
 See SNMP
single mode, 7-2
 See also monomode
single-attached station
 See SAS
single-CPE configuration, 15-4
SIP
 detailed description, 15-5
 Level 1, 15-8
 Level 2, 15-7
 Level 3, 15-6
 transmission system sublayer, 15-8
slow convergence
 See convergence
Small Computer Systems Interface 2
 See SCSI-2
SMDS
 and relationship to IEEE 802.6, 15-3
 general operation, 15-1
 network implementation, 15-9
 SIP, 15-3
SMDS Interface Protocol
 See SIP
SMI, 32-4
SMTP, 18-10
SNA
 and IBM network management, 33-1
 and SDLC, 12-1
SNI, 15-1
SNMP
 and CMIP, 20-7
 and IBM network management, 33-6
 and the Internet protocols, 18-10
 and UltraNet, 8-1
 general operation, 32-1
SNP, 28-5
SNPA, 28-3
source service access point
 See SSAP

source-route bridge, 31-1
Source-Route Bridging
 See SRB
Source-Route Transparent
 See SRT
spanning-tree algorithm
 See STA
spanning-tree explorer, 30-3
specifically routed frame, 30-3
SPF algorithm, 25-4
split horizons, 23-4, 24-4
SPP, 19-5, 22-4
SPX, 19-5
SRB
 bridging algorithm, 30-2
 general operation, 30-1
SRT, 30-1, 31-2, 31-4
SSAP, 12-6
STA, 29-3
standards organizations, 1-8
Stanford University, 18-1
StarLAN, 8-3
static routing, 2-6, 18-7
Stratacom, 14-1
Structure of Management Information
 See SMI
subnet, 18-5
subnet masks, 25-6
subnetwork point of attachment
 See SNPA
subscriber network interface
 See SNI
Sun Microsystems, 19-1, 20-7
supervisory (S) frames, 12-3
SVC, 13-3, 14-2
Switched Multimegabit Data Service
 See SMDS
switched virtual circuit
 See SVC, 14-2
switching
 definition, 2-1
 fast packet, 15-9
switching algorithms, 2-2
Systems Network Architecture
 See SNA
SystemView, 33-5

T

T1S1 committee, 14-1
T3, 9-1
T3Plus Networking, 9-1
TA, 11-1
table of all known networks, 21-6
table of neighbors, 21-6
TCP
 and SNMP, 32-1
 and SPP, 22-4
 and TP4, 20-5
 and XNS, 22-1
 definition of, 18-1
 frame format, 18-9
TCP/IP
 and RIP, 23-1
 and UltraNet, 8-3
 background, 18-1
TDM, 14-1
TE1, 11-1
TE2, 11-1
TELENET, 13-1
Telnet, 18-10
terminal adapter
 See TA
TIA TR30.2 committee, 9-1
tick
 and SNMP, 32-5
time-division multiplexing
 See TDM
TLB, 31-2, 31-3
token, 6-2, 20-6
Token Ring
 and IBM network management, 33-6
 as a NetWare media, 19-2
 as a VINES media, 21-2
 as an AppleTalk media, 16-2
 fault management mechanisms, 6-4
 frame format, 6-5
 general operation, 6-1
 MTU different from that of Ethernet, 31-2
 priority system, 6-4
TokenTalk, 16-2
topological database, 25-2
topology change message, 29-5

TOS
 and IS-IS, 28-2
TP0, 20-5
TP1, 20-5
TP2, 20-5
TP3, 20-5
TP4, 20-5, 22-4
transaction ID (AppleTalk), 16-8
translation challenges, 31-2
Translational Bridging
 See TLB
Transmission Control Protocol
 See TCP
Transmission Control Protocol/Internet Protocol
 See TCP/IP
transparent bridge, 31-1
Transparent Bridging
 general operation, 29-1
trap operation (SNMP), 32-5
TYMNET, 13-1
type of service
 See TOS

U

UDP, 18-8, 19-5, 22-4
Ultra Network Technologies, 8-1
UltraNet
 and the OSI reference model, 8-1
 general operation, 8-1
 host software components, 8-3
 network management bus, 8-3
 topology, 8-2
UltraNet frame buffer, 8-4
UltraNet hub, 8-4
UltraNet Manager, 8-3
UltraNet network processors, 8-4
Ungermann-Bass
 adopted XNS, 22-1
 and RIP, 23-1
UNIX, 8-3, 23-1
unnumbered (U) frames, 12-3
User Datagram Protocol
 See UDP

V

V.24, 13-6
V.28, 13-6
V.35, 9-1, 10-2, 20-2
variable-length subnet masks, 25-6, 28-2
VINES
 general operation, 21-1
 network-layer protocols, 21-2
 routing algorithm, 21-5
 transport-layer protocols, 21-8
 upper-layer protocols, 21-8
VINES Internetwork Protocol
 See VIP
VIP
 frame format, 21-5
 general operation, 21-2
virtual circuit, 13-3, 14-1
virtual links, 25-4
Virtual Telecommunications Access Method
 See VTAM
Virtual Terminal Protocol
 See VTP
VTAM, 33-5
VTP, 20-7

W

warranty information (Cisco Systems), xxiii
workstation
 See client

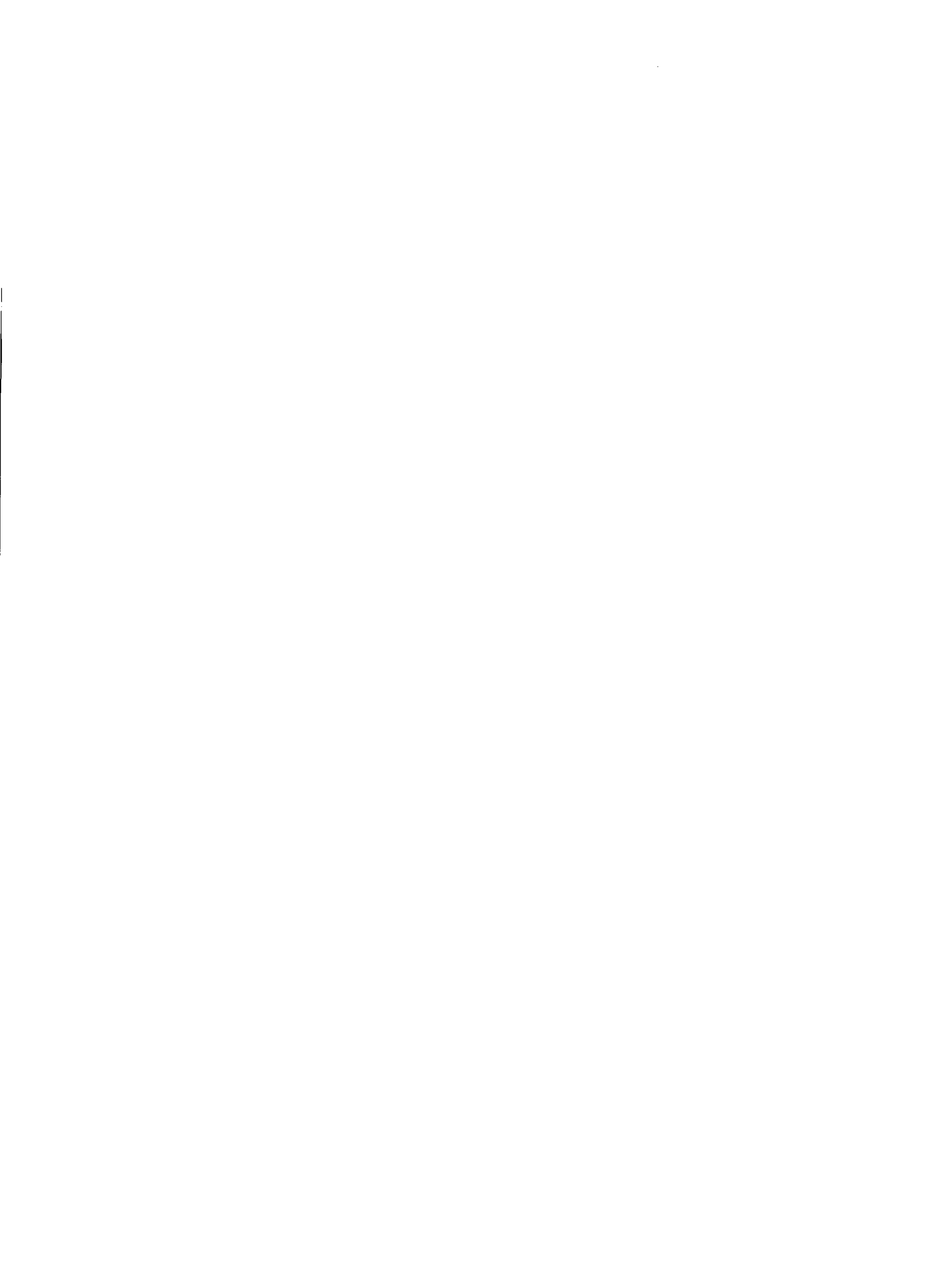
X

X Windows, 18-10
X.21, 20-2
X.21bis, 13-6
X.25
 and DECnet, 17-2
 and Frame Relay, 14-1
 and OSI, 20-2
 and OSI routing, 28-2
 and PPP, 10-1
 and SDLC, 12-5
 and VINES, 21-2
 frame format, 13-4
 general operation, 13-1

- Layer 1, 13-6
- Layer 2, 13-5
- Layer 3, 13-4
- X.28, 13-2
- X.29, 13-2
- X.3, 13-2
- X.500, 20-7
- X3S3.3 committee, 28-1
- XDR, 18-10
- Xerox Corporation
 - and Ethernet, 5-1
 - and NetWare, 19-1
 - and VINES, 21-1
 - and XNS, 22-1
- Xerox PARC, 5-1
- XNS
 - and NetWare, 19-1
 - and RIP, 23-1
 - and the OSI reference model, 22-1
 - and VINES, 21-1
 - general operation, 22-1
 - media-access protocols, 22-2
 - network-layer protocols, 22-3
 - transport-layer protocols, 22-4
 - upper-layer protocols, 22-5
- XON/XOFF, 14-3

Z

- ZIP, 16-7
- ZIT, 16-7
- zone, 16-4
- Zone Information Protocol
 - See ZIP
- zone information table
 - See ZIT



Corporate Headquarters

Cisco Systems, Inc.
P.O. Box 3075
1525 O'Brien Drive
Menlo Park, CA 94026
USA
Tel: 415 326-1941
800 553-NETS (6387)
Fax: 415 326-1989

European Headquarters

Cisco Systems Europe, s.a.r.l.
BP 706 Evolic
16 avenue du Quebec
91961 Les Ulis Cedex
France
Tel: 33 1 6092 2000
Fax: 33 1 6928 8326

European Offices**Belgium**

Tel: 32 2 643 2626
Fax: 32 2 643 2627

Germany

Tel: 49 89 3215 070
Fax: 49 89 3215 0710

Italy

Tel: 39 2 62 726 43
Fax: 39 2 62 729 13

Spain

Tel: 34 1 57 203 60
Fax: 34 1 57 071 99

Sweden

Tel: 46 8 19 62 05
Fax: 46 8 19 04 24

Switzerland

Tel: 41 55 95 60 44
Fax: 41 55 95 64 14

United Kingdom

Tel: 44 494 464944
Fax: 44 494 465300

Intercontinental Headquarters

(Latin America and Asia-Pacific)

Cisco Systems, Inc.
1525 O'Brien Drive
P.O. Box 3075
Menlo Park, CA 94026
USA
Tel: 415 326-1941
Fax: 415 688-4646

Regional Offices

Cisco Systems Australia Pty., Ltd.

Tel: 61 2 957 4944
Fax: 61 2 957 4077

Cisco Systems Canada Limited

Tel: 416 506-1500
Fax: 416 506-1506

Cisco Systems Hong Kong, Ltd.

Tel: 852 529 3534
Fax: 852 520 2676

Cisco Systems de México, S.A. de C.V.

Tel: 525 254 0880
Fax: 525 531 9659

Cisco Systems New Zealand

Tel: 9 649 358 3776
Fax: 9 649 358 4442

Japanese Headquarters

Nihon Cisco Systems K.K.
Shiba Excellent Building, 5F
2-1-13 Hamamatsucho,
MinatoKuTokyo 105, Japan
Tel: 81 3 5472 3571
Fax: 81 3 5472 3577

Cisco Systems has over 50 sales offices worldwide. Call 415 326-1941 to contact your local account representative or, in North America, call 800 553-NETS (6387)

