



www.chinatag.com

CHINATAG

640-801

ICND 2.1 and INTRO

Study Guide
DEMO Version

Copyright (c) 2003 Chinatag LLC. All rights reserved.

Important Note Please Read Carefully

For demonstration purpose only, this free version Chinatag study guide contains **10** full length questions selected from our full version products which have more than **200** questions each.

This Study guide has been carefully written and compiled by Chinatag certification experts. It is designed to help you learn the concepts behind the questions rather than be a strict memorization tool. Repeated readings will increase your comprehension.

For promotion purposes, all PDF files are **not** encrypted. Feel free to distribute copies among your friends and let them know Chinatag website. Our IT certification products start at only **\$7.99**.

Study Tips

This product will provide you questions and answers along with detailed explanations carefully compiled and written by our experts. Try to understand the concepts behind the questions instead of cramming the questions. Go through the entire document at least twice so that you make sure that you are not missing anything.

Latest Version

We are constantly reviewing our products. New material is added and old material is revised. Free updates are available for 180 days after the purchase. You should check the products page on the <http://www.chinatag.com> website for an update 3-4 days before the scheduled exam date.

Please tell us what you think of our products. We appreciate both positive and critical comments as your feedback helps us improve future versions. Feedback on specific questions should be send to feedback@chinatag.com.

Thanks for purchasing our products and look forward to supplying you with all your Certification training needs.

Good studying!

Technical and Support Team
Chinatag LLC.

TABLE OF CONTENTS

List of Tables

Introduction

1. Networking Fundamentals

1.1 The OSI Reference Model

1.1.1 Interaction Between OSI Layers

1.2 TCP/IP and the OSI Reference Model

1.2.1 The TCP/IP Protocol Architecture

1.2.2 TCP/IP Data Encapsulation

1.3 Networks

1.3.1 Network Definitions

1.3.2 Types of Networks

1.3.3 Network Topologies

1.3.4 Network Technologies

1.3.4.1 Ethernet

1.3.4.2 Fast Ethernet

1.3.4.3 Gigabit Ethernet

1.3.4.4 Token Ring

1.3.5 Network Addressing

1.3.6 Bridging

1.3.7 LAN Switching

1.3.8 Wireless Networks

1.3.8.1 Wireless Network Standards

1.3.8.2 Wireless Network Modes

1.3.8.3 Security Features

1.4 The Cisco IOS Software

1.4.1 The Cisco IOS Software Command-Line Interface

1.4.1.1 The CLI Help Features

1.4.1.2 Syslog Messages and the debug Command

1.4.2 Configuring Cisco IOS Software

1.4.2.1 Managing Configuration Files

1.4.2.2 Upgrading Cisco IOS Software

1.4.2.3 The Cisco IOS Software Boot Sequence

1.5 Spanning-Tree Protocol (STP)

1.5.1 Root Bridge Election

1.5.2 Root Ports Election

1.5.3 Designated Ports Election

1.5.4 STP States

1.5.5 STP Timers

640-801 ICND 2.1 & INTRO

- 1.5.6 Optional STP Features
 - 1.5.6.1 EtherChannel
 - 1.5.6.2 PortFast
 - 1.5.6.3 Rapid Spanning Tree (IEEE 802.1w)

2. Virtual LANs and Trunking

- 2.1 VLAN Membership
- 2.2 Extent of VLANs
- 2.3 VLAN Trunking
 - 2.3.1 Inter-Switch Link (ISL)
 - 2.3.2 802.1Q
- 2.4 VLAN Trunking Protocol (VTP)
 - 2.4.1 VTP Modes
 - 2.4.1.1 Server Mode
 - 2.4.1.2 Client Mode
 - 2.4.1.3 Transparent Mode
 - 2.4.2 VTP Pruning
 - 2.4.3 VTP Configuration
 - 2.4.3.1 Configuring a VTP Management Domain
 - 2.4.3.2 Configuring the VTP Mode
 - 2.4.3.3 Configuring the VTP Version

3. IP Addressing and Subnetting

- 3.1 IP Addressing
 - 3.1.1 Binary Format
 - 3.1.2 Dotted Decimal Format
 - 3.1.3 IP Address Classes
 - 3.1.4 Classless Interdomain Routing (CIDR) Notation
 - 3.1.5 Variable-Length Subnet Masks
- 3.2 Subnetting
- 3.3 Summarization
 - 3.3.1 Automatic Summarization
 - 3.3.2 Manual Summarization
- 3.4 Determining the Network ID using the Logical AND Operation

4. Routing

- 4.1 Routing Tables
 - 4.1.1 Static Routing

640-801 ICND 2.1 & INTRO

- 4.1.2 Dynamic Routing
- 4.1.3 Routing Updates
- 4.1.4 Verifying Routing Tables

4.2 Routing Protocols

- 4.2.1 Distance-Vector Routing
 - 4.2.1.1 Route Poisoning
 - 4.2.1.2 Split Horizon
 - 4.2.1.3 Split Horizon with Poison Reverse
 - 4.2.1.4 Hold-Down Timer
 - 4.2.1.5 Triggered Updates
- 4.2.2 Link-State Routing
- 4.2.3 Classful Routing
- 4.2.4 Classless Routing

4.3 Basic Switching Functions

4.4 Convergence

- 4.4.1 Distance-Vector Routing Convergence
 - 4.4.1.1 RIP and IGRP Convergence
 - 4.4.1.2 EIGRP Convergence
- 4.4.2 Link-State Convergence

4.5 Testing and Troubleshooting Routes

- 4.5.1 The ping Command
- 4.5.2 The traceroute Command

5. Link-State Protocols

5.1 Building Routing Table on New OSPF-Configured Routers

5.2 Steady-State Operation

5.3 OSPF Areas

- 5.3.1 OSPF Area Types
- 5.3.2 Router Responsibilities

5.4 Balanced Hybrid Routing Protocol and EIGRP

- 5.4.1 EIGRP Loop Avoidance

5.5 Router Configuration

- 5.5.1 Configuring OSPF
- 5.5.2 Verifying the OSPF Configuration
- 5.5.3 Configuring EIGRP
- 5.5.4 Verifying the EIGRP Configuration

6. Advanced TCP/IP



640-801 ICND 2.1 & INTRO

6.1 Private IP Addressing

6.2 Network Address Translation (NAT)

6.2.1 Variations of NAT

6.2.1.1 Static NAT

6.2.1.2 Dynamic NAT

6.2.1.3 Overloading NAT with Port Address Translation (PAT)

6.2.1.4 Translating Overlapping Addresses

6.2.2 Configuring NAT

6.2.2.1 Configuring Static NAT

6.2.2.2 Configuring Dynamic NAT

6.2.2.3 Configuring NAT Overload and PAT

6.3 Internet Control Message Protocol (ICMP)

6.4 FTP and TFTP

6.5 MTU and Fragmentation

7. Wide Area Networks (WANs)

7.1 Point-to-Point Leased Lines

7.1.1 Overview

7.1.2 Data-Link Protocols

7.1.3 Configuring HDLC and PPP Configuration

7.2 Integrated Services Digital Network (ISDN)

7.2.1 ISDN Channels

7.2.2 ISDN Protocols

7.2.3 ISDN Layers

7.2.3.1 ISDN Layer 1

7.2.3.2 ISDN Layer 2

7.2.3.3 ISDN Layer 3

7.2.4 BRI Function Groups and Reference Points

7.2.5 Encoding and Framing

7.2.6 Dial-on-Demand Routing (DDR)

7.2.7 ISDN Configuration for DDR

7.2.7.1 Configuring Legacy DDR

7.2.7.2 Configuring DDR with Dialer Profiles

7.2.8 Multilink PPP

7.3 Frame Relay

7.3.1 Virtual Circuits

7.3.2 LMI and Encapsulation Types

7.3.3 DLCI Addressing

7.3.4 Frame Relay Configuration

7.3.4.1 Determining the Interface

- 7.3.4.2 Configuring Frame Relay Encapsulation
- 7.3.4.3 Configuring Protocol-Specific Parameters
- 7.3.4.4 Configuring Frame Relay Characteristics
- 7.3.4.5 Verifying Frame Relay Configuration

8. IP Access Control List Security

- 8.1 Standard IP Access Control Lists
 - 8.1.1 Wildcard Masks
 - 8.1.2 Standard IP Access List Configuration
- 8.2 Extended IP Access Control Lists
- 8.3 Named IP Access Lists
- 8.4 Controlling Telnet Access with ACLs

Appendix A: Decimal to Binary Conversion Table

LIST OF TABLES

TABLE 1.1:	The TCP/IP Architectural Model and Protocols
TABLE 1.2:	Network Definitions
TABLE 1.3:	Coaxial Cable for Ethernet
TABLE 1.4:	Twisted-Pair and Fiber Optic Cable for Ethernet
TABLE 1.5:	Fast Ethernet Cabling and Distance Limitations
TABLE 1.6:	Gigabit Ethernet Cabling and Distance Limitations
TABLE 1.7:	The boot system Commands
TABLE 4.1:	Parameters for the ping Command
TABLE 4.2:	Parameters for the traceroute Command
TABLE 5.1:	EIGRP, IGRP and OSPF Compared
TABLE 6.1:	The Private IP Address Space defined by RFC 1918
TABLE 6.2:	ICMP Messages

LIST OF ACRONYMS

AAA	Authentication, Authorization, and Accounting
ABR	Area Border Router
ACF	Advanced Communications Function
ACK	Acknowledgment bit (in a TCP segment)
ACL	Access Control List
ACS	Access Control Server
AD	Advertised Distance
ADSL	Asymmetric Digital Subscriber Line
ANSI	American National Standards Institute
API	Application Programming Interface
APPC	Advanced Program-to-Program Communications
ARAP	AppleTalk Remote Access Protocol
ARE	All Routes Explorer
ARP	Address Resolution Protocol
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
AS	Autonomous System
ASA	Adaptive Security Algorithm
ASBR	Autonomous System Boundary Router
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuits
ATM	Asynchronous Transfer Mode
AUI	Attachment Unit Interface
Bc	Committed burst (Frame Relay)
B channel	Bearer channel (ISDN)
BDR	Backup Designated Router
Be	Excess burst (Frame Relay)
BECN	Backward Explicit Congestion Notification (Frame Relay)
BGP	Border Gateway Protocol
BGP-4	Border Gateway Protocol version 4
BIA	Burned-in Address (another name for a MAC address)

640-801 ICND 2.1 & INTRO

BOD	Bandwidth on Demand.
BPDU	Bridge Protocol Data Unit
BRF	Bridge Relay Function
BRI	Basic Rate Interface (ISDN)
BSD	Berkeley Standard Distribution (UNIX)
CBT	Core Based Trees
CBWFQ	Class-Based Weighted Fair Queuing
CCITT	Consultative Committee for International Telegraph and Telephone
CCO	Cisco Connection Online
CDDI	Copper Distribution Data Interface
CEF	Cisco Express Forwarding
CHAP	Challenge Handshake Authentication Protocol
CIDR	Classless Interdomain Routing
CIR	Committed Information Rate. (Frame Relay)
CGMP	Cisco Group Management Protocol
CLI	Command-Line Interface
CLSC	Cisco LAN Switching Configuration
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CR	Carriage Return.
CRC	Cyclic Redundancy Check (error)
CRF	Concentrator Relay Function
CST	Common Spanning Tree
CSU	Channel Service Unit
DB	Data Bus (connector)
DCE	Data Circuit-Terminating Equipment
dCEF	Distributed Cisco Express Forwarding
DDR	Dial-on-Demand Routing
DE	Discard Eligible Indicator
DECnet	Digital Equipment Corporation Protocols
DES	Data Encryption Standard
DHCP	Dynamic Host Control Protocol
DLCI	Data-Link Connection Identifier
DNIC	Data Network Identification Code. (X.121addressing)

640-801 ICND 2.1 & INTRO

DNS	Domain Name System
DoD	Department of Defense (US)
DR	Designated Router
DRiP	Duplicate Ring Protocol
DS	Digital Signal
DS0	Digital Signal level 0
DS1	Digital Signal level 1
DS3	Digital Signal level 3
DSL	Digital Subscriber Line
DSU	Data Service Unit
DTE	Data Terminal Equipment
DTP	Dynamic Trunking Protocol
DUAL	Diffusing Update Algorithm
DVMRP	Distance Vector Multicast Routing Protocol
EBC	Ethernet Bundling Controller
EGP	Exterior Gateway Protocol
EIA/TIA	Electronic Industries Association/Telecommunications Industry Association
EIGRP	Enhanced Interior Gateway Routing Protocol
ESI	End-System Identifier
FCC	Federal Communications Commission
FCS	Frame Check Sequence
FC	Feasible Condition (Routing)
FD	Feasible Distance (Routing)
FDDI	Fiber Distributed Data Interface
FEC	Fast EtherChannel
FECN	Forward Explicit Congestion Notification
FIB	Forwarding Information Base
FIFO	First-In, First-Out (Queuing)
FR	Frame Relay
FS	Feasible Successor (Routing)
FSSRP	Fast Simple Server Redundancy Protocol
FTP	File Transfer Protocol
GBIC	Gigabit Interface Converters
GEC	Gigabit EtherChannel

640-801 ICND 2.1 & INTRO

GSR	Gigabit Switch Router
HDLC	High-Level Data Link Control
HDSL	High data-rate digital subscriber line
HSRP	Hot Standby Router Protocol
HSSI	High-Speed Serial Interface
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IDN	International Data Number
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
ILMI	Integrated Local Management Interface
IOS	Internetwork Operating System
IP	Internet Protocol
IPSec	IP Security
IPv6	IP version 6
IPX	Internetwork Packet Exchange (Novell)
IRDP	ICMP Router Discovery Protocol
IS	Information Systems
IS-IS	Intermediate System-to-Intermediate System
ISDN	Integrated Services Digital Network
ISL	Inter-Switch Link
ISO	International Organization for Standardization
ISOC	Internet Society
ISP	Internet Service Provider
ITU-T	International Telecommunication Union–Telecommunication Standardization Sector
kbps	kilobits per second (bandwidth)
LAN	Local Area Network
LANE	LAN Emulation
LAPB	Link Access Procedure, Balanced
LAPD	Link Access Procedure on the D channel

640-801 ICND 2.1 & INTRO

LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LED	Light Emitting Diode
LES	LAN Emulation Server
LLC	Logic Link Control (OSI Layer 2 sublayer)
LLQ	Low-Latency Queuing
LMI	Local Management Interface
LSA	Link-State Advertisement
MAC	Media Access Control (OSI Layer 2 sublayer)
MAN	Metropolitan-Area Network
MD5	Message Digest Algorithm 5
MLS	Multilayer Switching
MLS-RP	Multilayer Switching Route Processor
MLS-SE	Multilayer Switching Switch Engine
MLSP	Multilayer Switching Protocol
MOSPF	Multicast Open Shortest Path First
MSAU	Multistation Access Unit
MSFC	Multilayer Switch Feature Card
MTU	Maximum Transmission Unit
NAK	Negative Acknowledgment
NAS	Network Access Server
NAT	Network Address Translation
NBMA	Nonbroadcast Multiaccess
NetBEUI	NetBIOS Extended User Interface
NetBIOS	Network Basic Input/Output System
NFFC	NetFlow Feature Card
NMS	Network Management System
NNI	Network-to-Network Interface
NSAP	Network Service Access Point
NVRAM	Nonvolatile Random Access Memory
OC	Optical Carrier
ODBC	Open Database Connectivity
OLE	Object Linking and Embedding
OSI	Open Systems Interconnection (Model)

640-801 ICND 2.1 & INTRO

OSPF	Open Shortest Path First
OTDR	Optical Time Domain Reflectometer
OUI	Organizationally Unique Identifier
PAGP	Port Aggregation Protocol
PAP	Password Authentication Protocol
PAT	Port Address Translation
PDN	Public Data Network
PDU	Protocol Data Unit (i.e., a data packet)
PIM	Protocol Independent Multicast
PIM	SM Protocol Independent Multicast Sparse Mode
PIMDM	Protocol Independent Multicast Mode
PIX	Private Internet Exchange (Cisco Firewall)
PNNI	Private Network-to-Network Interface
POP	Point of Presence
POTS	Plain Old Telephone Service
PPP	Point-to-Point Protocol
PQ	Priority Queuing
PRI	Primary Rate Interface (ISDN)
PSTN	Public Switched Telephone Network
PTT	Poste, Telephone, Telegramme
PVC	Permanent Virtual Circuit (ATM)
PVST	Per-VLAN Spanning Tree
PVST+	Per-VLAN Spanning Tree Plus
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RAS	Remote Access Service
RIF	Routing Information Field
RIP	Routing Information Protocol
RJ	Registered Jack (connector)
RMON	Embedded Remote Monitoring
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSFC	Route Switch Feature Card
RSM	Route Switch Module

640-801 ICND 2.1 & INTRO

RSP	Route Switch Processor
RSTP	Rapid Spanning Tree Protocol
RTP	Reliable Transport Protocol
RTO	Retransmission Timeout
SA	Source Address
SAID	Security Association Identifier
SAP	Service Access Point; also Service Advertising Protocol (Novell)
SAPI	Service Access Point Identifier
SAR	Segmentation and Reassembly
SDLC	Synchronous Data Link Control (SNA)
SIA	Stuck in Active (EIGRP)
SIN	Ships-in-the-Night (Routing)
SLIP	Serial Line Internet Protocol
SMDS	Switched Multimegabit Data Service
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture (IBM)
SNAP	SubNetwork Access Protocol
SNMP	Simple Network Management Protocol
SOF	Start of Frame
SOHO	Small Office, Home Office
SONET	Synchronous Optical Network
SONET/SDH	Synchronous Optical Network/Synchronous Digital Hierarchy
SPAN	Switched Port Analyzer
SPF	Shortest Path First
SPID	Service Profile Identifier
SPP	Sequenced Packet Protocol (Vines)
SPX	Sequenced Packet Exchange (Novell)
SQL	Structured Query Language
SRAM	Static Random Access Memory
SRB	Source-Route Bridge
SRT	Source-Route Transparent (Bridging)
SRTT	Smooth Round-Trip Timer (EIGRP)
SS7	Signaling System 7
SSAP	Source service access point (LLC)

640-801 ICND 2.1 & INTRO

SSE	Silicon Switching Engine.
SSP	Silicon Switch Processor
SSRP	Simple Server Redundancy Protocol
STA	Spanning-Tree Algorithm
STP	Spanning-Tree Protocol; also Shielded Twisted-Pair (cable)
SVC	Switched Virtual Circuit (ATM)
SYN	Synchronize (TCP segment)
TA	Terminal Adapter (ISDN)
TAC	Technical Assistance Center (Cisco)
TACACS	Terminal Access Controller Access Control System
TCI	Tag Control Information
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TCN	Topology Change Notification
TDM	Time-Division Multiplexing
TDR	Time Domain Reflectometers
TFTP	Trivial File Transfer Protocol
TIA	Telecommunications Industry Association
TLV	Type-Length-Value
ToS	Type of Service
TPID	Tag Protocol Identifier
TrBRF	Token Ring Bridge Relay Function
TrCRF	Token Ring Concentrator Relay Function
TTL	Time-To-Live
UDP	User Datagram Protocol
UNC	Universal Naming Convention or Uniform Naming Convention
UNI	User-Network Interface
URL	Uniform Resource Locator
UTC	Coordinated Universal Time (same as Greenwich Mean Time)
UTL	Utilization
UTP	Unshielded Twisted-Pair (cable)
VBR	Variable Bit Rate
VC	Virtual Circuit (ATM)
VID	VLAN Identifier

640-801 ICND 2.1 & INTRO

VIP	Versatile Interface Processor
VLAN	Virtual Local Area Network
VLSM	Variable-Length Subnet Mask
VMPS	VLAN Membership Policy Server
VPN	Virtual Private Network
VTP	VLAN Trunking Protocol
vt	Virtual terminal line
WAIS	Wide Area Information Server
WAN	Wide Area Network
WFQ	Weighted Fair Queuing
WLAN	Wireless Local Area Network
WWW	World Wide Web
XNS	Xerox Network Systems
XOR	Exclusive-OR
XOT	X.25 over TCP
ZIP	Zone Information Protocol (AppleTalk)

ICND 2.1 and INTRO

Exam Code: 640-801

Certifications:

Cisco Certified Network Associate (CCNA)

Core

Prerequisites:

None

About This Study Guide

This Study Guide is based on the current pool of exam questions for the Cisco CCNA 640-801 composite exam. As such it provides all the information required to pass the 640-801 exam and is organized around the specific skills that are tested in that exam. Thus, the information contained in this Study Guide is specific to the 640-801 exam and does not represent a complete reference work on the subject of Interconnecting Cisco Networking Devices. Topics covered in this Study Guide includes: Designing or Modifying a simple Local Area Network (LAN) using Cisco Products; Designing an IP Addressing Scheme; Selecting Appropriate Routing Protocols; Designing a simple Internetwork using Cisco products; Developing an Access List to Meet User Specifications; Choosing Wide Area Network (WAN) Services; Managing System Image and Device Configuration Files Performing an Initial Configuration on a Switch; Configuring Routing Protocols; Configuring IP Addresses, Subnet Masks, and Gateway Addresses on Routers and Hosts; Configuring a Router for Additional Administrative Functionality; Configuring a Switch with Virtual LANs (VLANs) and Inter-switch Communication; Implementing a LAN; Customizing a Switch Configuration; Implementing Access Lists; Implementing Simple WAN Protocols; Utilizing the OSI Reference Model as a Guide for Systematic Network Troubleshooting; Performing LAN and VLAN Troubleshooting; Troubleshooting Routing Protocols; Troubleshooting IP Addressing and Host Configuration; Troubleshooting a Device as Part of a Working Network; Troubleshooting an Access List; Performing Simple WAN Troubleshooting; Understanding Network Communications based on Layered Models; Understanding the Components of Network Devices; Understanding the Spanning Tree Process; Evaluating the Characteristics of LAN Environments; Evaluating the TCP/IP Communication Process and its Associated Protocols; Evaluating the Characteristics of Routing Protocols; Evaluating Rules for Packet Control; and Evaluating Key Characteristics of WANs.

Intended Audience

This Study Guide is targeted specifically at people who wish to take the Cisco CCNA 640-801 Composite exam. This information in this Study Guide is specific to the exam. It is not a complete reference work. Although our Study Guides are aimed at new comers to the world of IT, the concepts dealt with in this Study Guide are complex. Knowledge of CompTIA's A+ and Network+ courses would be advantageous.

Note: Because the 640-801 exam is a composite of the 640-811 and 640-821 exams, there is a fair amount of overlap between this Study Guide and the

640-801 ICND 2.1 & INTRO

640-811 and 640-821 Study Guides. However, this Study Guide does not combine the 640-811 and 640-821 Study Guides but addresses the 640-801 exam specifically. As such, we would not advise using this Study Guide for the 640-811 exam and/or the 640-821 exam.

How To Use This Study Guide

To benefit from this Study Guide we recommend that you:

- Study each chapter carefully until you fully understand the information. This will require regular and disciplined work. Where possible, attempt to implement the information in a lab setup.
- Be sure that you have studied and understand the entire Study Guide before you take the exam.

Note: Remember to pay special attention to these note boxes as they contain important additional information that is specific to the exam.

Good luck!

1. Networking Fundamentals

1.1 The OSI Reference Model

The OSI is the Open System Interconnection reference model for communications. As illustrated in Figure 1.1, the OSI reference model consists of seven layers, each of which can have several sublayers. The upper layers of the OSI reference model define functions focused on the application, while the lower three layers define functions focused on end-to-end delivery of the data.

- The **Application Layer (Layer 7)** refers to communications services to applications and is the interface between the network and the application. Examples include: Telnet, HTTP, FTP, Internet browsers, NFS, SMTP gateways, SNMP, X.400 mail, and FTAM.
- The **Presentation Layer (Layer 6)** defining data formats, such as ASCII text, EBCDIC text, binary, BCD, and JPEG. Encryption also is defined as a presentation layer service. Examples include: JPEG, ASCII, EBCDIC, TIFF, GIF, PICT, encryption, MPEG, and MIDI.
- The **Session Layer (Layer 5)** defines how to start, control, and end communication sessions. This includes the control and management of multiple bidirectional messages so that the application can be notified if only some of a series of messages are completed. This allows the presentation layer to have a seamless view of an incoming stream of data. The presentation layer can be presented with data if all flows occur in some cases. Examples include: RPC, SQL, NFS, NetBios names, AppleTalk ASP, and DECnet SCP
- The **Transport Layer (Layer 4)** defines several functions, including the choice of protocols. The most important Layer 4 functions are error recovery and flow control. The transport layer may provide for retransmission, i.e., error recovery, and may use flow control to prevent unnecessary congestion by attempting to send data at a rate that the network can accommodate, or it might not, depending on the choice of protocols. Multiplexing of incoming data for different flows to applications on the same host is also performed. Reordering of the incoming data stream when packets arrive out of order is included. Examples include: TCP, UDP, and SPX.
- The **Network Layer (Layer 3)** defines end-to-end delivery of packets and defines logical addressing to accomplish this. It also defines how routing works and how routes are learned; and how to fragment a packet into smaller packets to accommodate media with smaller maximum transmission unit sizes. Examples include: IP, IPX, AppleTalk DDP, and ICMP. Both IP and IPX define logical addressing, routing, the learning of routing information, and end-to-end delivery rules. The IP and IPX protocols most closely match the OSI network layer (Layer 3) and are called Layer 3 protocols because their functions most closely match OSI's Layer 3.
- The **Data Link Layer (Layer 2)** is concerned with getting data across one particular link or medium. The data link protocols define delivery across an individual link. These protocols are necessarily concerned with the type of media in use. Examples include: IEEE 802.3/802.2, HDLC, Frame Relay, PPP, FDDI, ATM, and IEEE 802.5/802.2.
- The **Physical Layer (Layer 1)** deals with the physical characteristics of the transmission medium. Connectors, pins, use of pins, electrical currents, encoding, and light modulation are all part of different

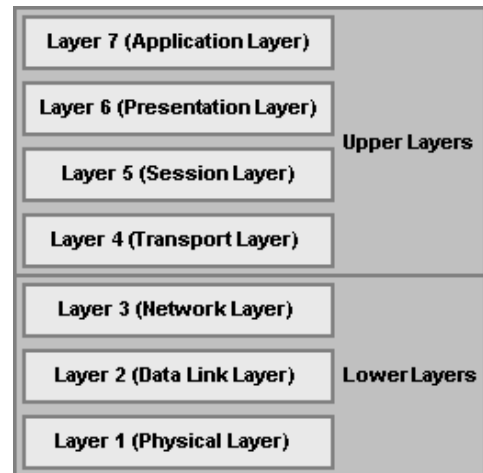


FIGURE 1.1: *The OSI Reference Model*

640-801 ICND 2.1 & INTRO

physical layer specifications. Examples includes: EIA/TIA-232, V.35, EIA/TIA-449, V.24, RJ-45, Ethernet, 802.3, 802.5, FDDI, NRZI, NRZ, and B8ZS.

The upper layers of the OSI reference model, i.e., the Application Layer (Layer 7), the Presentation Layer (Layer 6), and the Session Layer (Layer 5), define functions focused on the application. The lower four layers, i.e., the Transport Layer (Layer 4), the Network Layer (Layer 3), the Data Link Layer (Layer 2), and the Physical Layer (Layer 1), define functions focused on end-to-end delivery of the data. As a Cisco Certified Network Associate, you will deal mainly with the lower layers, particularly the data link layer (Layer 2) upon which switching is based, and the network layer (Layer 3) upon which routing is based.

1.1.1 Interaction Between OSI Layers

When a host receives a data transmission from another host on the network, that data is processed at each of the OSI layers to the next higher layer, in order to render the data transmission useful to the end-user. To facilitate this processing, headers and trailers are created by the sending host's software or hardware, that are placed before or after the data given to the next higher layer. Thus, each layer has a header and trailer, typically in each data packet that comprises the data flow. The sequence of processing at each OSI layer, i.e., the processing between adjacent OSUI layers, is as follows:

- The **Physical Layer** (Layer 1) ensures **bit synchronization** and places the received binary pattern into a buffer. It notifies the Data Link Layer (Layer 2) that a frame has been received after decoding the incoming signal into a bit stream. Thus, Layer 1 provides delivery of a stream of bits across the medium.
- The **Data Link Layer** (Layer 2) examines the **frame check sequence (FCS)** in the trailer to determine whether errors occurred in transmission, providing **error detection**. If an error has occurred, the frame is discarded. The current host examines data link address is examined to determine if the data is addressed to it or whether to process the data further. If the data is addressed to the host, the data between the Layer 2 header and trailer is handed over to the Network Layer (Layer 3) software. Thus, the data link layer delivers data across the link.
- The **Network Layer** (Layer 3) examines the destination address. If the address is the current host's address, processing continues and the data after the Layer 3 header is handed over to the Transport Layer (Layer 4) software. Thus, Layer 3 provides end-to-end delivery.
- If error recovery was an option chosen for the **Transport Layer** (Layer 4), the counters identifying this piece of data are encoded in the Layer 4 header along with acknowledgment information, which is called **error recovery**. After error recovery and reordering of the incoming data, the data is given to the Session Layer (Layer 5).
- The **Session Layer** (Layer 5) ensures that a series of messages is completed. The Layer 5 header includes fields signifying sequence of the packet in the data stream, indicating the position of the data packet in the flow. After the session layer ensures that all flows are completed, it passes the data after the Layer 5 header to the Presentation Layer (Layer 6) software.
- The **Presentation Layer** (Layer 6) defines and manipulates the data format of the data transmission. It converts the data to the proper format specified in the Layer 6 header. Typically, this header is included only for initialization flows, not with every data packet being transmitted. After the data formats have been converted, the data after the Layer 6 header is passed to the Application Layer (Layer 7) software.
- The **Application Layer** (Layer 7) processes the final header and examines the end-user data. This header signifies agreement to operating parameters by the applications on the two hosts. The headers are used to

signal the values for all parameters; therefore, the header typically is sent and received at application initialization time only.

In addition to processing between adjacent OSI layers, the various layers must also interact with the same layer on another computer to successfully implement its functions. To interact with the same layer on another computer, each layer defines additional data bits in the header and, in some cases, trailer that is created by the sending host's software or hardware. The layer on the receiving host interprets the headers and trailers created by the corresponding layer on the sending host to determine how that layer's processing is being defined, and how to interact within that framework.

1.2 TCP/IP and the OSI Reference Model

As illustrated in Figure 1.2, the TCP/IP model consists of four layers, each of which can have several sublayers. These layers correlate roughly to layers in the OSI reference model and define similar functions. Some of the TCP/IP layers correspond directly with layers in the OSI reference model while other span several OSI layers. The four TCP/IP layers are:

- The **TCP/IP Application Layer** refers to communications services to applications and is the interface between the network and the application. It is also responsible for presentation and controlling communication sessions. It spans the Application Layer, Presentation Layer and Session Layer of the OSI reference model. Examples include: HTTP, POP3, and SNMP.
- The **TCP/IP Transport Layer** defines several functions, including the choice of protocols, error recovery and flow control. The transport layer may provide for retransmission, i.e., error recovery, and may use flow control to prevent unnecessary congestion by attempting to send data at a rate that the network can accommodate, or it might not, depending on the choice of protocols. Multiplexing of incoming data for different flows to applications on the same host is also performed. Reordering of the incoming data stream when packets arrive out of order is included. It correlates with the Transport Layer of the OSI reference model. Examples include: TCP and UDP, which are called **Transport Layer**, or **Layer 4**, protocols.
- The **TCP/IP Internetwork Layer** defines end-to-end delivery of packets and defines logical addressing to accomplish this. It also defines how routing works and how routes are learned; and how to fragment a packet into smaller packets to accommodate media with smaller maximum transmission unit sizes. It correlates with the Network Layer of the OSI reference model. Examples include: IP and ICMP.
- The **TCP/IP Network Interface Layer** is concerned with the physical characteristics of the transmission medium as well as getting data across one particular link or medium. This layer defines delivery across an individual link as well as the physical layer specifications. It spans the Data Link Layer and Physical Layer of the OSI reference model. Examples include: Ethernet and Frame Relay.

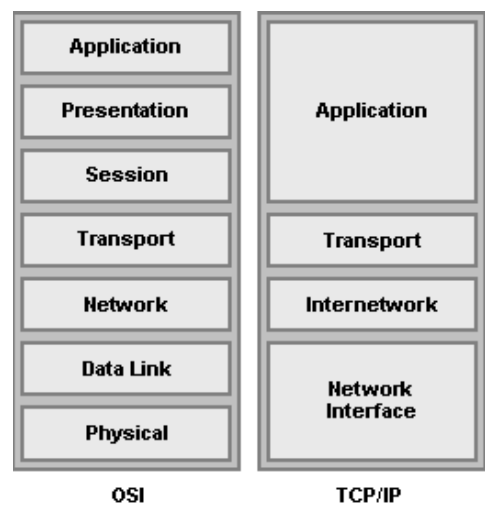


FIGURE 1.2: OSI, TCP/IP and NetWare

1.2.1 The TCP/IP Protocol Architecture

TCP/IP defines a large collection of protocols that allow computers to communicate. Table 1.1 outlines the protocols and the TCP/IP architectural layer to which they belong. TCP/IP defines the details of each of these protocols in **Requests For Comments (RFC)** documents. By implementing the required protocols defined in TCP/IP RFCs, a computer that implements the standard networking protocols defined by TCP/IP can communicate with other computers that also use the TCP/IP standards.

TABLE 1.1: *The TCP/IP Architectural Model and Protocols*

TCP/IP Architecture Layer	Protocols
Application	HTTP, POP3, SMTP
Transport	TCP, UDP
Internetwork	IP
Network interface	Ethernet, Frame Relay

1.2.2 TCP/IP Data Encapsulation

The term **encapsulation** describes the process of putting headers and trailers around some data. A computer that needs to send data encapsulates the data in headers of the correct format so that the receiving computer will know how to interpret the received data. Data encapsulation with TCP/IP consists of five-steps:

Step 1: Create the application data and headers.

Step 2: Package the data for transport, which is performed by the transport layer (TCP or UDP). The Transport Layer creates the transport header and places the data behind it.

Step 3: Add the destination and source network layer addresses to the data, which is performed by the Internetwork Layer. The Internetwork Layer creates the network header, which includes the network layer addresses, and places the data behind it.

Step 4: Add the destination and source data link layer addresses to the data, which is performed by the Network Interface Layer. The Network Interface Layer creates the data link header, places the data behind it, and places the data link trailer at the end.

Step 5: Transmit the bits, which is performed by the Network Interface Layer. The Network Interface Layer encodes a signal onto the medium to transmit the frame.

1.3 Networks

A network is defined as a group of two or more computers linked together for the purpose of communicating and sharing information and other resources, such as printers and applications. Most networks are constructed around a cable connection that links the computers, however, modern wireless networks that use radio wave or infrared connections are also becoming quite prevalent. These connections permit the computers to communicate via the wires in the cable, radio wave or infrared signal. For a network to function it must provide connections, communications, and services.

- **Connections** are defined by the hardware or physical components that are required to connect a computer to the network. This includes the **network medium**, which refers to the hardware that physically connects one computer to another, i.e., the network cable or a wireless connection; and the **network interface**, which refers to the hardware that attaches a computer to the network medium and is usually a network interface card (NIC).

640-801 ICND 2.1 & INTRO

- **Communications** refers to the network protocols that are used to establish the rules governing network communication between the networked computers. Network protocols allow computers running different operating systems and software to communicate with each.
- **Services** define the resources, such as files or printers, that a computer shares with the rest of the networked computers.

1.3.1 Network Definitions

Computer networks can be classified and defined according to geographical area that the network covers. There are four network definitions: a Local Area Network (LAN), a Campus Area Network (CAN), a Metropolitan Area Network (MAN), and a Wide Area Network (WAN). There are three additional network definitions, namely the Internet, an intranet and an Internetwork. These network definitions are discussed in Table 1.2.

TABLE 1.2: *Network Definitions*

Definition	Description
Local Area Network (LAN)	A LAN is defined as a network that is contained within a closed environment and does not exceed a distance of 1.25 mile (2 km). Computers and peripherals on a LAN are typically joined by a network cable or by a wireless network connection. A LAN that consists of wireless connections is referred to as a Wireless LAN (WLAN) .
Campus Area Network (CAN)	A CAN is limited to a single geographical area but may exceed the size of a LAN
Metropolitan Area Network (MAN)	A MAN is defined as a network that covers the geographical area of a city that is less than 100 miles.
Wide Area Network (WAN)	A WAN is defined as a network that exceeds 1.25 miles. A WAN often consists of a number of LANs that have been joined together. A CAN and a MAN is also a WAN. WANs typically connected numerous LANs through the internet via telephone lines, T1 lines, Integrated Services Digital Network (ISDN) lines, radio waves, cable or satellite links.
Internet	The Internet is a world wide web of networks that are based on the TCP/IP protocol and is not own by a single company or organization.
Intranet	An intranet uses that same technology as the Internet but is owned and managed by a company or organization. A LAN or a WAN s usually an intranet.
Internetwork	An internetwork consists of a number of networks that are joined by routers. The Internet is the largest example of an internetwork.

Of these network definitions, the most common are the Internet, the LAN and the WAN.

1.3.2 Types of Networks

These network definitions can be divided into two types of networks, based on how information is stored on the network, how network security is handled, and how the computers on the network interact. These two types are: **Peer-To-Peer (P2P) Networks** and **Server/Client Networks**. The latter is often also called Server networks.

- On a **Peer-To-Peer (P2P) Network**, there is no hierarchy of computers; instead each computer acts as either a server which shares its data or services with other computers, or as a client which uses data or services on another computer. Furthermore, each user establishes the security on their own computers and determines which of their resources are made available to other users. These networks are typically limited to between 15 and 20 computers. Microsoft Windows for Workgroups, Windows 95, Windows 98, Windows ME, Windows NT Workstation, Windows 2000, Novell's NetWare, UNIX, and Linux are some operating systems that support peer-to-peer networking.
- A **Server/Client Network** consists of one or more dedicated computers configured as servers. This server manages access to all shared files and peripherals. The server runs the network operating system (NOS) manages security and administers access to resources. The client computers or workstations connect to the network and use the available resources. Among the most common network operating systems are Microsoft's Windows NT Server 4, Windows 2000 Server, and Novell's NetWare. Before the release of Windows NT, most dedicated servers worked only as hosts. Windows NT allows these servers to operate as an individual workstation as well.

1.3.3 Network Topologies

The layout of a LAN design is called its topology. There are three basic types of topologies: the star topology, the bus topology, and the ring topology. Hybrid combinations of these topologies also exist.

- In a network based on the **star topology**, all computers and devices are connected to a centrally located hub or switch. The hub or switch collects and distributes the flow of data within the network. When a hub is used, data from the sending host are sent to the hub and are then transmitted to all hosts on the network except the sending host. Switches can be thought of as intelligent hubs. When switches are used rather than hubs, data from the sending host are sent to the switch which transmits the data to the intended recipient rather than to all hosts on the network.
- In a network based on the **bus topology**, all computers and devices are connected in series to a single linear cable called a trunk. The trunk is also known as a backbone or a segment. Both ends of the trunk must be terminated to stop the signal from bouncing back up the cable. Because a bus network does not have a central point, it is more difficult to troubleshoot than a star network. Furthermore, a break or problem at any point along the bus can cause the entire network to go down.

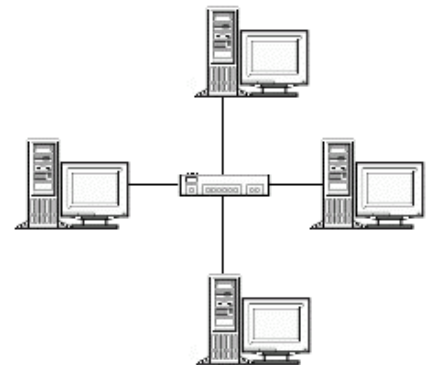


FIGURE 1.3: *The Star Topology*

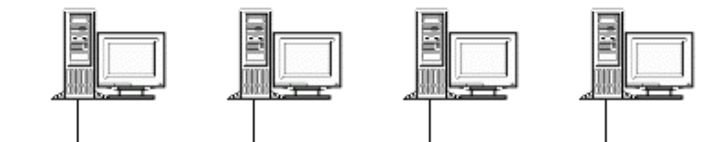


FIGURE 1.4: *The Bus Topology*

640-801 ICND 2.1 & INTRO

- In a network based on a **ring topology**, all computers and devices are connected to cable that forms a closed loop. On such networks there are no terminating ends; therefore, if one computer fails, the entire network will go down. Each computer on such a network acts like a repeater and boosts the signal before sending it to the next station. This type of network transmits data by passing a “token” around the network. If the token is free of data, a computer waiting to send data grabs it, attaches the data and the electronic address to the token, and sends it on its way. When the token reaches its destination computer, the data is removed and the token is sent on. Hence this type of network is commonly called a token ring network.

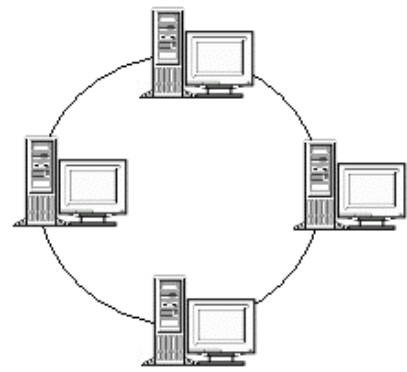


FIG 1.5: *The Ring Topology*

Of these three network topologies, the star topology is the most predominant network type and is based on the Ethernet standard.

1.3.4 Network Technologies

Various network technologies can be used to establish network connections, including Ethernet, Fiber Distribution Data Interface (FDDI), Copper Distribution Data Interface (CDDI), Token Ring, and Asynchronous Transfer Mode (ATM). Of these, Ethernet is the most popular choice in installed networks because of its low cost, availability, and scalability to higher bandwidths.

1.3.4.1 Ethernet

Ethernet is based on the Institute of Electrical and Electronics Engineers (IEEE) 802.3 standard and offers a bandwidth of 10 Mbps between end users. Ethernet is based on the carrier sense multiple access collision detect (CSMA/CD) technology, which requires that transmitting stations back off for a random period of time when a collision occurs.

Coaxial cable was the first media system specified in the Ethernet standard. Coaxial Ethernet cable comes in two major categories: **Thicknet** (10Base5) and **Thinnet** (10Base2). These cables differed in their size and their length limitation. Although Ethernet coaxial cable lengths can be quite long, they are susceptible to electromagnetic interference (EMI) and eavesdropping.

TABLE 1.3: *Coaxial Cable for Ethernet*

Cable	Diameter	Resistance	Bandwidth	Length
Thinnet (10Base2)	10 mm	50 ohms	10 Mbps	185 m
Thicknet (10Base5)	5 mm	50 ohms	10 Mbps	500 m

Today most wired networks use twisted-pair media for connections to the desktop. Twisted-pair also comes in two major categories: **Unshielded twisted-pair (UTP)** and **Shielded twisted-pair (STP)**. One pair of insulated copper wires twisted about each other forms a twisted-pair. The pairs are twisted to reduce interference and crosstalk. Both STP and UTP suffer from high attenuation, therefore these lines are usually restricted to an end-to-end distance of 100 meters between active devices. Furthermore, these cables are sensitive to EMI and eavesdropping. Most networks use 10BaseT UTP cable.

640-801 ICND 2.1 & INTRO

An alternative to twisted-pair cable is fiber optic cable (10BaseFL), which transmits light signals, generated either by light emitting diodes (LEDs) or laser diodes (LDs), instead of electrical signals. These cables support higher transmission speeds and longer distances but are more expensive. Because they do not carry electrical signals, fiber optic cables are immune to EMI and eavesdropping. They also have low attenuation which means they can be used to connect active devices that are up to 2 km apart. However, fiber optic devices are not cost effective while cable installation is complex.

TABLE 1.4: Twisted-Pair and Fiber Optic Cable for Ethernet

Cable	Technology	Bandwidth	Cable Length
Twisted-Pair	(10BaseT)	10 Mbps	100 m
Fiber Optic	(10BaseFL)	10 Mbps	2,000 m

1.3.4.2 Fast Ethernet

Fast Ethernet operates at 100 Mbps and is based on the IEEE 802.3u standard. The Ethernet cabling schemes, CSMA/CD operation, and all upper-layer protocol operations have been maintained with Fast Ethernet. Fast Ethernet is also backward compatible with 10 Mbps Ethernet. Compatibility is possible because the two devices at each end of a network connection can automatically negotiate link capabilities so that they both can operate at a common level. This negotiation involves the detection and selection of the highest available bandwidth and half-duplex or full-duplex operation. For this reason, Fast Ethernet is also referred to as **10/100 Mbps Ethernet**.

Cabling for Fast Ethernet can be either UTP or fiber optic. Specifications for these cables are shown in Table 1.5.

TABLE 1.5: Fast Ethernet Cabling and Distance Limitations

Technology	Wiring Type	Pairs	Cable Length
100BaseTX	EIA/TIA Category 5 UTP	2	100 m
100BaseT2	EIA/TIA Category 3,4,5 UTP	2	100 m
100BaseT4	EIA/TIA Category 3,4,5 UTP	4	100 m
100BaseFX	Multimode fiber (MMF) with 62.5 micron core; 1300 nm laser	1	400 m (half-duplex) 2,000 m (full-duplex)
	Single-mode fiber (SMF) with 62.5 micron core; 1300 nm laser	1	10,000 m

1.3.4.3 Gigabit Ethernet

Gigabit Ethernet is an escalation of the Fast Ethernet standard using the same IEEE 802.3 Ethernet frame format. Gigabit Ethernet offers a throughput of 1,000 Mbps (1 Gbps). Like Fast Ethernet, Gigabit Ethernet is compatible with earlier Ethernet standards. However, the physical layer has been modified to increase data transmission speeds: The IEEE 802.3 Ethernet standard and the American National Standards Institute (ANSI) X3T11 FibreChannel. IEEE 802.3 provided the foundation of frame format, CSMA/CD, full duplex, and other characteristics of Ethernet. FibreChannel provided a base of high-speed ASICs, optical

640-801 ICND 2.1 & INTRO

components, and encoding/decoding and serialization mechanisms. The resulting protocol is termed IEEE 802.3z Gigabit Ethernet.

Gigabit Ethernet supports several cabling types, referred to as 1000BaseX. Table 1.6 lists the cabling specifications for each type.

TABLE 1.6: Gigabit Ethernet Cabling and Distance Limitations

Technology	Wiring Type	Pairs	Cable Length
1000BaseCX	Shielded Twisted Pair (STP)	1	25 m
1000BaseT	EIA/TIA Category 5 UTP	4	100 m
1000BaseSX	Multimode fiber (MMF) with 62.5 micron core; 850 nm laser	1	275 m
	Multimode fiber (MMF) with 50 micron core; 1300 nm laser	1	550 m
1000BaseLX/LH	Multimode fiber (MMF) with 62.5 micron core; 1300 nm laser	1	550 m
	Single-mode fiber (SMF) with 50 micron core; 1300 nm laser	1	550 m
	Single-mode fiber (SMF) with 9 micron core; 1300 nm laser	1	10 km
1000BaseZX	Single-mode fiber (SMF) with 9 micron core; 1550 nm laser	1	70 km
	Single-mode fiber (SMF) with 8 micron core; 1550 nm laser	1	100 km

1.3.4.4 Token Ring

Like Ethernet, Token Ring is a LAN technology that provides shared media access to many connected hosts. Token Ring hosts are arranged using the ring topology. A token is passed from host to host around the ring, giving the current token holder permission to transmit a frame onto the ring. Once the frame is sent, it is passed around the ring until it is received again by the source. The sending host is responsible for removing the frame from the ring and for introducing a new token to the next neighboring host. This means that only one station can transmit at a given time, and prevents a Token Ring network experiencing collisions.

A Token Ring network offers a bandwidth of 4 Mbps or 16 Mbps. At the higher rate, hosts are allowed to introduce a new token as soon as they finish transmitting a frame. This early token release increases efficiency by letting more than one host transmit a frame during the original token's round trip. One station is elected to be the ring monitor, to provide recovery from runaway frames or tokens. The ring monitor will remove frames that have circled the ring once, if no other station removes them.

Traditional Token Ring networks use multistation access units (MSAUs) to provide connectivity between hosts. MSAUs have several ports that a host can connect to, with either a B connector for Type 2 cabling or an RJ-45 connector for Category 5 UTP cabling. Internally, the MSAU provides host-to-host connections to

form a ring segment. The Ring-In and Ring-Out connectors of a MSAU can be chained to other MSAUs to form a complete ring topology.

1.3.5 Network Addressing

Network addressing identifies either individual devices or groups of devices on a LAN. A pair of network devices that transmit frames between each other use a source and destination address field to identify each other. These addresses are called **unicast** addresses, or individual addresses, because they identify an individual network interface card (NIC).

The IEEE defines the format and assignment of network addresses by requiring manufacturers to encode globally unique unicast Media Access Control (MAC) addresses on all NICs. The first half of the MAC address identifies the manufacturer of the card and is called the **organizationally unique identifier (OUI)**.

1.3.6 Bridging

Bridging is used to connect two network segments. This alleviates congestion problems on a single Ethernet segment and extends allowed cabling distances because the segments on each side of the bridge conformed to the same distance limitation as a single segment. This bridge is called “**transparent bridging**” because the end-point devices do not need to know that the bridge exists.

Transparent bridges forward frames only when necessary and, thus, reduces network overhead. To accomplish this, transparent bridges learn MAC addresses by examining the source MAC address of each frame received by the bridge; decides when to forward a frame or when to filter a frame, based on the destination MAC address; and creates a loop-free environment with other bridges by using the Spanning Tree Protocol.

Generally, broadcasts and multicast frames are forwarded by the bridge in networks that use bridges. In addition, transparent bridges perform switching of frames using Layer 2 headers and Layer 2 logic and are Layer 3 protocol-independent. Store-and-forward operation, which means that the entire frame is received before the first bit of the frame is forwarded, is also typical in transparent bridging devices. However, the transparent bridge must perform processing on the frame, which also can increase latency.

A transparent bridge operates in the following manner:

- The bridge has **no initial knowledge** of the location of any end device; therefore, the bridge must listen to frames coming into each of its ports to figure out on which network a device resides.
- The bridge constantly **updates its bridging table** upon detecting the presence of a new MAC address or upon detecting a MAC address that has changed location from one bridge port to another. The bridge is then able to forward frames by looking at the destination address, looking up the address in the bridge table, and sending the frame out the port where the destination device is located.
- If a frame arrives with the broadcast address as the destination address, the bridge must **forward** or flood the frame **out all available ports**. However, the frame is not forwarded out the port that initially received the frame. Hence, broadcasts are able to reach all available networks. A bridge only segments collision domains but does not segment broadcast domains.
- If a frame arrives with a **destination address that is not found** in the bridge table, the bridge is unable to determine which port to forward the frame to for transmission. This is known as an **unknown unicast**. In this case, the bridge treats the frame as if it was a broadcast and forwards it out all remaining ports.

After a reply to that frame is received, the bridge will learn the location of the unknown station and add it to the bridge table.

- Frames that are forwarded across the bridge **cannot be modified**.

1.3.7 LAN Switching

An Ethernet switch uses the same logic as a transparent bridge, but performs more functions, has more features, and has more physical ports. Switches use hardware to learn MAC addresses and to make forwarding and filtering decisions, whereas bridges use software.

A switch listens for frames that enter all its interfaces. After receiving a frame, a switch decides whether to forward a frame and out which port(s). To perform these functions, switches perform three tasks:

- **Learning**, which means that the switch learns MAC addresses by examining the source MAC address of each frame the bridge receives. Switches dynamically learn the MAC addresses in the network to build its MAC address table. With a full, accurate MAC address table, the switch can make accurate forwarding and filtering decisions. Switches build the MAC address table by listening to incoming frames and examining the frame's source MAC address. If a frame enters the switch, and the source MAC address is not in the address table, the switch creates an entry in the table. The MAC address is placed in the table, along with the interface in which the frame arrived. This allows the switch to make good forwarding choices in the future. Switches also forward unknown unicast frames, which are frames whose destination MAC addresses are not yet in the bridging table, out all ports, which is called **flooding**, with the hope that the unknown device will be on some other Ethernet segment and will reply. When the unknown device does reply, the switch will build an entry for that device in the address table.
- **Forwarding or filtering**, which means that the switch decides when to forward a frame or when to filter it, i.e., not to forward it, based on the destination MAC address. Switches reduce network overhead by forwarding traffic from one segment to another only when necessary. To decide whether to forward a frame, the switch uses a dynamically built table called a **bridge table** or **MAC address table**. The switch looks at the previously learned MAC addresses in an address table to decide where to forward the frames.
- **Loop prevention**, which means that the switch creates a loop-free environment with other bridges by using **Spanning Tree Protocol (STP)**. Having physically redundant links helps LAN availability, and STP prevents the switch logic from letting frames loop around the network indefinitely, congesting the LAN.

Frames sent to unicast addresses are destined for a single device; frames sent to a broadcast address are sent to all devices on the LAN. Frames sent to multicast addresses are meant for all devices that care to receive the frame. Thus, when a switch receives a frame, it checks if the address is a unicast address, a broadcast address or a multicast address. If the address is unicast, and the address is in the address table, and if the interface connecting the switch to the destination device is not the same interface on which the frame arrived, the switch forwards the frame to the destination device. If the address is not in the address table, the switch forwards the frame on all ports. If the address is a broadcast or multicast address, the switch also forwards the frame on all ports.

The internal processing on a switch can decrease latency for frames. Switches can use store-and-forward processing as well as **cut-through** processing logic. With cut-through processing, the first bits of the frame are sent out the outbound port before the last bit of the incoming frame is received. However, because the

frame check sequence (FCS) is in the Ethernet trailer, a cut-through forwarded frame might have bit errors that the switch will not notice before sending most of the frame.

1.3.8 Wireless Networks

Conventional Ethernet networks require cables connected computers via hubs and switches. This has the effect of restricting the computer's mobility and requires that even portable computers be physically connected to a hub or switch to access the network. An alternative to cabled networking is wireless networking. The first wireless network was developed at the University of Hawaii in 1971 to link computers on four islands without using telephone wires. Wireless networking entered the realm of personal computing in the 1980s, with the advent to networking computers. However, it was only in the early 1990s that wireless networks started to gain momentum when CPU processing power became sufficient to manage data transmitted and received over wireless connections.

Wireless networks use network cards, called Wireless Network Adapters, that rely radio signals or infrared (IR) signals to transmit and receive data via a Wireless Access Point (WAP). The WAP uses has an RJ-45 port that can be attached to attach to a 10BASE-T or 10/100BASE-T Ethernet hub or switch and contains a radio transceiver, encryption, and communications software. It translates conventional Ethernet signals into wireless Ethernet signals it broadcasts to wireless network adapters on the network and performs the same role in reverse to transfer signals from wireless network adapters to the conventional Ethernet network. WAP devices come in many variations, with some providing the Cable Modem Router and Switch functions in addition to the wireless connectivity.

Note: Access points are not necessary for direct peer-to-peer networking, which is called ad hoc mode, but they are required for a shared Internet connection or a connection with another network. When access points are used, the network is operating in the infrastructure mode.

1.3.8.1 Wireless Network Standards

In the absence of an industry standard, the early forms of wireless networking were single-vendor proprietary solutions that could not communicate with wireless network products from other vendors. In 1997, the computer industry developed the IEEE 802.11 wireless Ethernet standard. Wireless network products based on this standard are capable of multivendor interoperability.

The IEEE 802.11 wireless Ethernet standard consists of the IEEE 802.11b standard, the IEEE 802.11a standard, and the newer IEEE 802.11g standard.

Note: The Bluetooth standard for short-range wireless networking is designed to complement, rather than rival, IEEE 802.11-based wireless networks.

- **IEEE 802.11** was the original standard for wireless networks that was ratified in 1997. It operated at a maximum speed of 2 Mbps and ensured interoperability between wireless products from various vendors. However, the standard had a few ambiguities allowed for potential problems with compatibility between devices. To ensure compatibility, a group of companies formed the Wireless Ethernet Compatibility Alliance (WECA), which has come to be known as the **Wi-Fi Alliance**, to ensure that their products would work together. The term **Wi-Fi** is now used to refer to any IEEE 802.11 wireless network products that have passed the Wi-Fi Alliance certification tests.

640-801 ICND 2.1 & INTRO

- **IEEE 802.11b**, which is also called 11 Mbps Wi-Fi, operates at a maximum speed of 11 Mbps and is thus slightly faster than 10BASE-T Ethernet. Most IEEE 802.11b hardware is designed to operate at four speeds, using three different data-encoding methods depending on the speed range. It operates at 11 Mbps using quaternary phase-shift keying/complimentary code keying (QPSK/CCK); at 5.5 Mbps also using QPSK/CCK; at 2 Mbps using differential quaternary phase-shift keying (DQPSK); and at 1 Mbps using differential binary phase-shift keying (DBPSK). As distances change and signal strength increases or decreases, IEEE 802.11b hardware switches to the most suitable data-encoding method.

Wireless networks running IEEE 802.11b hardware use the 2.4 GHz radio frequency band that many portable phones, wireless speakers, security devices, microwave ovens, and the Bluetooth short-range networking products use. Although the increasing use of these products is a potential source of interference, the short range of wireless networks (indoor ranges up to 300 feet and outdoor ranges up to 1,500 feet, varying by product) minimizes the practical risks. Many devices use a spread-spectrum method of connecting with other products to minimize potential interference.

IEEE 802.11b networks can connect to wired Ethernet networks or be used as independent networks.

- **IEEE 802.11a** uses the 5 GHz frequency band, which allows for much higher speeds, reaching a maximum speed of 54 Mbps. The 5 GHz frequency band also helps avoid interference from devices that cause interference with lower-frequency IEEE 802.11b networks. IEEE 802.11a hardware maintains relatively high speeds at both short and relatively long distances.

Because IEEE 802.11a uses the 5 GHz frequency band rather than the 2.4 GHz frequency band used by IEEE 802.11b, standard IEEE 802.11a hardware cannot communicate with 802.11b hardware. A solution to this compatibility problem is the use of dual-band hardware. Dual-band hardware can work with either IEEE 802.11a or IEEE 802.11b networks, enabling you to move from an IEEE 802.11b wireless network at home or at Starbucks to a faster IEEE 802.11a office network.

- **IEEE 802.11g** is also known as Wireless-G and combines compatibility with IEEE 802.11b with the speed of IEEE 802.11a at longer distances. This standard was ratified in mid-2003, however, many network vendors were already selling products based on the draft IEEE 802.11g standard before the final standard was approved. These early IEEE 802.11g hardware was slower and less compatible than the specification promises. In some cases, problems with early-release IEEE 802.11g hardware can be solved through firmware upgrades.

1.3.8.2 Wireless Network Modes

Wireless networks work in one of two modes that are also referred to as topologies. These two modes are ad-hoc mode and infrastructure mode. The mode you implement depends on whether you want your computers to communicate directly with each other, or via a WAP.

- In ad-hoc mode, data is transferred to and from wireless network adapters connected to the computers. This cuts out the need to purchase a WAP. Throughput rates between two wireless network adapters are twice as fast as when you use a WAP. However, a network in ad-hoc mode cannot connect to a wired network as a WAP is required to provide connectivity to a wired network. An ad-hoc network is also called a peer-to-peer network.
- In infrastructure mode, data is transferred between computers via a WAP. Because a WAP is used in infrastructure mode, it provides connectivity with a wired network, allowing you to expand a wired network with wireless capability. Your wired and wirelessly networked computers can communicate with each other. In addition, a WAP can extend your wireless network's range as placing a WAP

between two wireless network adapters doubles their range. Also, some WAPs have a built-in router and firewall. The router allows you to share Internet access between all your computers, and the firewall hides your network. Some of these multifunction access points include a hub with RJ-45 ports.

1.3.8.3 Security Features

Because wireless networks can be accessed by anyone with a compatible wireless network adapter, most models of wireless network adapters and WAPs provide for encryption options. Some devices with this feature enable you to set a security code known as an SSID on the wireless devices on your network. This seven-digit code prevents unauthorized users from accessing your network and acts as an additional layer of security along with your normal network authentication methods, such as user passwords. Other wireless network adapters and WAPs use a list of authorized MAC numbers to limit access to authorized devices only.

All Wi-Fi products support at least 40-bit encryption through the wired equivalent privacy (WEP) specification, but the minimum standard on newer products is 64-bit WEP encryption. Many vendors also offer 128-bit or 256-bit encryption on some of their products. However, the WEP specification is insecure. It is vulnerable to brute-force attacks at shorter key lengths, and it is also vulnerable to **differential cryptanalysis** attacks, which is the process of comparing an encrypted text with a known portion of the plain text and deriving the key by computing the difference between them. Because WEP encrypts TCP headers, hackers know what the headers should contain in many cases, and they can attempt to find patterns in a large body of collected WEP communications in order to decrypt the key. The attack is complex and difficult to automate, so it is unlikely to occur for most networks, especially at key lengths greater than 128 bits. Furthermore, WEP does not prevent an intruder from attaching a hidden WAP on the network and using it to exploit the network.

New network products introduced in 2003 and beyond now incorporate a new security standard known as Wi-Fi Protected Access (WPA). WPA is derived from the developing IEEE 802.11i security standard, which will not be completed until mid-decade. WPA-enabled hardware works with existing WEP-compliant devices, and software upgrades might be available for existing devices.

1.4 The Cisco IOS Software

Cisco routers run the Cisco Internetworking Operating System (IOS) with a command-line interface (CLI). The IOS also runs on some Cisco switch models, and it uses CLI. However, in some cases, the IOS CLI on a switch is slightly different than on a router. Furthermore, the IOS on the 1900 series switches is slightly different than on some other Cisco IOS-based switches.

1.4.1 The Cisco IOS Software Command-Line Interface

The majority of Cisco routers run Cisco IOS Software with the command-line interface (CLI). The CLI is used to interface with the device and send commands to the device. This is achieved through the use of a terminal, a terminal emulator, or a Telnet connection. Some routing cards, such as the Multilayer Switch Feature Card (MSFC) daughter card for the Catalyst 6000 series LAN switches, also run Cisco IOS Software. Understanding the Cisco IOS Software CLI is as fundamental to supporting routers.

There are three ways in which you can access the CLI: you access the router through the console; through a dialup device through a modem attached to the auxiliary port; or by using a Telnet connection. Which ever

method you use, you enter **user exec mode** first. User exec mode is one of three command exec modes in the IOS user interface. **Enable mode**, also known as privileged mode or Privileged exec mode, and command mode are the others. Enable mode is so named because the **enable** command is used to reach this mode. User mode allows commands that are not disruptive to be issued, with some information being displayed to the user. Privileged mode supports a superset of commands compared to user mode. However, none of the commands in user mode or privileged mode changes the configuration of the router.

Passwords are required for Telnet and auxiliary access as of Cisco IOS Release 12.x and later. However, there are no preconfigured passwords; therefore, you must configure passwords for Telnet and auxiliary access from the console first.

All Cisco routers have a console port, and most have an auxiliary port. The console port is intended for local administrative access from an ASCII terminal or a computer using a terminal emulator. The auxiliary port is intended for asynchronous dial access from an ASCII terminal or terminal emulator; the auxiliary port is often used for dial backup.

1.4.1.1 The CLI Help Features

Typing **?** in the console displays help for all commands supported by the CLI mode. In other words, the information supplied by using help depends on the CLI mode. If **?** is typed in user mode, the commands allowed only in privileged exec mode are not displayed. Also, help is available in configuration mode; only configuration commands are displayed in that mode of operation. IOS stores the commands that you type in a history buffer. The last ten commands are stored by default. You can change the history size with the **terminal history size size** command, where **size** is the number of IOS commands for the CLI to store; this can be set to a value between 0 and 256. You can then retrieve commands so that you do not have to retype the commands.

1.4.1.2 Syslog Messages and the `debug` Command

IOS creates messages, which are called **syslog messages**, when different events occur and, by default, sends them to the console. The router also generates messages that are treated like syslog messages in response to some troubleshooting tasks that you might perform. The **debug** command is one of the key diagnostic tools for troubleshooting problems on a Cisco router. It enables monitoring points in the IOS and generates messages that describe what the IOS is doing and seeing. When any **debug** command option is enabled, the router processes the messages with the same logic as other syslog messages.

The console port always receives syslog messages; however, when you Telnet to the router no syslog messages are seen unless you issue the **terminal monitor** command. Another alternative for viewing syslog messages is to have the IOS record the syslog messages in a buffer in RAM and then use the **show logging** command to display the messages. For Telnet users, having the messages buffered using the global config command **logging buffered** is particularly useful. Finally, the **logging synchronous** line-configuration subcommand can be used for the console and vtys to tell the router to wait until the last command output is displayed before showing any syslog messages onscreen.

Syslog messages also can be sent to another device. Two alternatives exist: sending the messages to a syslog server, and sending the messages as SNMP traps to a management station. The **logging host** command, where **host** is the IP address or host name of the syslog server, is used to enable sending messages to the

external server. After SNMP is configured, the `snmp-server enable traps` command tells the IOS to forward traps, including syslog messages.

1.4.2 Configuring Cisco IOS Software

Configuration mode is one of the modes for the Cisco CLI. It is similar to user mode and privileged mode. User mode allows commands that are not disruptive to be issued, with some information being displayed to the user. Privileged mode supports a superset of commands compared to user mode. However, none of the commands in user or privileged mode changes the configuration of the router. Configuration mode is another mode in which configuration commands are typed.

Commands typed in configuration mode update the active configuration file. These changes to the configuration occur immediately each time you press the Enter key at the end of a command. Configuration mode itself contains a multitude of **subcommand modes**. The type of command you enter moves you from one configuration subcommand mode to which ever subcommand mode is appropriate. For example, the `interface` command, which is the most commonly used configuration command, would move you to interface configuration mode.

Generally, when multiple instances of a parameter can be set on a single router, the command used to set the parameter is likely a configuration subcommand. Items that are set once for the entire router are likely global commands. For example, the `hostname` command is a global command because there is only one host name per router.

You can use **CTRL + Z** from any part of configuration mode, or use the `exit` command from global configuration mode, to exit configuration mode and return to privileged exec mode. The configuration mode `end` command also exits from any point in the configuration mode back to privileged exec mode. The `exit` commands from subcommand modes back up one level toward global configuration mode.

1.4.2.1 Managing Configuration Files

Your configuration commands, as well as some default configuration commands are stored in the configuration file. No hard disk or diskette storage exists on Cisco routers therefore; the configuration file is stored in memory. The configuration files can also be stored as **ASCII text files** anywhere exterior to the router using TFTP or FTP. Cisco routers support a number of types of memory. This includes:

- RAM, which is sometimes called DRAM for dynamic random-access memory, is used by the router in the same way it is used by any other computer: for storing data being used by the processor. The **active** configuration file, `running-config`, which is the configuration file that the router uses during operation, is stored in RAM.
- ROM, or read-only memory, stores a bootable IOS image, which is not typically used for normal operation. It contains the code that is used to boot the router and allows the router to access the IOS image.
- Flash memory, which can be either an EEPROM or a PCMCIA card, stores fully functional IOS images and is the default location where the router accesses its IOS at boot time. Flash memory also can be used to store configuration files on some Cisco routers.
- NVRAM, which is nonvolatile RAM, stores the initial or **startup** configuration file, `startup-config`.

All these types of memory, except RAM, are permanent memory.

640-801 ICND 2.1 & INTRO

When the router first comes up, the router copies the stored configuration file from NVRAM into RAM, so the active and startup configuration files are identical at that point. The `show running-config` and `show startup-config` commands are used to verify the active and startup configuration files respectively. You can use the `copy running-config startup-config` command to overwrite the current startup configuration file with the current active configuration file. The `copy` command can be used to copy files in a router, most typically a configuration file, or a new version of the IOS Software. The most basic method for moving configuration files in and out of a router is by using a TFTP server. The `copy` command is used to copy configuration files among RAM, NVRAM, and a TFTP server. The syntax for `copy` command used to copy configuration files among RAM, NVRAM, and a TFTP server specifies the source location and the destination of the configuration file as in:

```
copy source destination
```

The *source* and the *destination* parameters can be `running-config`, `startup-config`, or `tftp` for RAM, NVRAM, and a TFTP server respectively. However, the *source* and the *destination* parameters cannot be the same. Thus, the following syntax copies the configuration from RAM to NVRAM, overwriting the current startup configuration file with the active configuration file:

```
copy running-config startup-config
```

The `copy` command does not always replace the existing file that it is copying. Any `copy` command option moving a file into NVRAM or a TFTP server replaces the existing file, however, any `copy` into RAM works by adding the commands to the active configuration file. Thus, if you change the active configuration file and then want to revert to the startup configuration file, you must use the `reload` command, which reboots the router.

Two commands can be used to erase the contents of NVRAM. These are the `write erase` command, which is the older command, and the `erase startup-config` command, which is the newer command.

1.4.2.2 Upgrading Cisco IOS Software

Typically, a router has one IOS image and that is the IOS that is used. This IOS image is typically stored in Flash memory, which is a rewriteable, permanent form of storage. The IOS image can also be placed on an external TFTP server, but this is typically done for testing. In the IOS upgrade process you first must obtain the IOS image from Cisco. Then you must place the IOS image into the default directory of a TFTP server. Finally, you must use the `copy tftp flash` command from the router to copy the files into Flash memory. During this process, the router will need to discover the IP address or host name of the TFTP server; the name of the file; the space available in Flash memory for this file; and whether you want to erase the old files. The router will prompt you for answers, as necessary. Afterward, the router erases Flash memory as needed, copies the file, and then verifies that the checksum for the file shows that no errors occurred in transmission. The `show flash` command then can be used to verify the contents of Flash memory. Before the new IOS is used, however, the router must be reloaded.

1.4.2.3 The Cisco IOS Software Boot Sequence

The basic boot sequence for a Cisco router is:



640-801 ICND 2.1 & INTRO

- Step 1:** The router performs a power-on self-test (POST) to discover and verify the hardware.
- Step 2:** The router loads and runs bootstrap code from ROM.
- Step 3:** The router finds the IOS or other software and loads it.
- Step 4:** The router finds the configuration file and loads it into running config.

All routers attempt all four steps each time that the router is powered on or reloaded. The POST code and functions cannot be changed by the router administrator. The location of the bootstrap code, the IOS to load, and the configuration file can be changed by the administrator—but you almost always use the default location for the bootstrap code (ROM) and for the initial configuration (NVRAM). So, the location of IOS or other software is the only part that typically is changed.

Three categories of operating systems can be loaded into the router:

- The **full-function IOS** image, which is typically located in Flash memory but can also be located on a TFTP server. This is the normal, full-feature IOS used in production;
- A **limited-function IOS** that resides in ROM; and provides basic IP connectivity when Flash memory is faulty and you need IP connectivity to copy a new IOS into Flash memory. This limited-function IOS is called **RXBOOT mode**.
- A different **non-IOS** operating system that is also stored in ROM. This operating system, called **ROM Monitor (ROMMON) mode**, is used for low-level debugging and for password recovery. Unless you are performing password recovery, you would seldom use ROMMON mode.

The **configuration register** tells the router whether to use a full-featured IOS, ROMMON, RXBOOT mode. The **configuration register** is a 16-bit software register in the router, and its value is set using the **config-register** global configuration command. The **boot field** is the name of the low-order 4 bits of the configuration register. This field can be considered a 4-bit value, represented as a single hexadecimal digit. If the boot field is hex 0, ROMMON is loaded. If the boot field is hex 1, RXBOOT mode is used. For anything else, it loads a full-featured IOS.

The second method used to determine where the router tries to obtain an IOS image is through the use of the **boot system** configuration command. If the configuration register calls for a full-featured IOS, the router reads the configuration file for boot system commands.

If there are no **boot system** commands, the router takes the default action, which is to load the first file in Flash memory. Table 1.7 lists the configuration register and the **boot system** command.

TABLE 1.7: *The boot system Commands*

Boot Filed Value	Function
0x0	Loads ROMMON and ignores boot system commands.
0x1	Loads IOS from ROM and ignores boot system commands. This is also known as RXBOOT mode.
0x2–0xF	If used with the no boot command, the first IOS file in Flash memory is loaded; if that fails, the router broadcasts looking for an IOS on a TFTP server. If that fails, IOS from ROM is loaded.

640-801 ICND 2.1 & INTRO

0x2-0xF	If used with the <code>boot system ROM</code> command, IOS from ROM is loaded.
0x2-0xF	If used with the <code>boot system flash</code> command, the first file from Flash memory is loaded.
0x2-0xF	If used with the <code>boot system flash file_name</code> command, IOS with the specified <code>file_name</code> is loaded from Flash memory.
0x2-0xF	If used with the <code>boot system tftp file_name 10.1.1.1</code> command, IOS with the specified <code>file_name</code> is loaded from the TFTP server.
0x2-0xF	If used with multiple <code>boot system</code> commands, an attempt occurs to load IOS based on the first boot command in configuration. If that fails, the second boot command is used, etc., until an IOS is loaded successfully.

1.5 Spanning-Tree Protocol (STP)

A Layer 2 switch, which functions as a transparent bridge, offers no additional links for redundancy purposes. To add redundancy, a second switch must be added. Now two switches offer the transparent bridging function in parallel. LAN designs with redundant links introduce the possibility that frames might loop around the network forever. These looping frames would cause network performance problems. For example, when the switches receive an unknown unicast, both will flood the frame out all their available ports, including the ports that link to the other switch, resulting in what is known as a bridging loop, as the frame is forwarded around and around between two switches. This occurs because parallel switches are unaware of each other. The **Spanning Tree Protocol (STP)**, which allows the redundant LAN links to be used while preventing frames from looping around the LAN indefinitely through those redundant links, was developed to overcome the possibility of bridging loops. It enables switches to become aware of each other so that they can negotiate a loop-free path through the network. Loops are discovered before they are opened for use, and redundant links are shut down to prevent the loops from forming. STP is communicated between all connected switches on a network. Each switch executes the **Spanning-Tree Algorithm (STA)** based on information received from other neighboring switches. The algorithm chooses a reference point in the network and calculates all the redundant paths to that reference point. When redundant paths are found, STA picks one path to forward frames with and disables or blocks forwarding on the other redundant paths. STP computes a tree structure that spans all switches in a subnet or network. Redundant paths are placed in a blocking or standby state to prevent frame forwarding. The switched network is then in a loop-free condition. However, if a forwarding port fails or becomes disconnected, the STA will run again to recompute the Spanning-Tree topology so that blocked links can be reactivated.

By default, STP is enabled on all ports of a switch. STP should remain enabled in a network to prevent bridging loops from forming. However, if STP has been disabled on a CLI-based switch, it can be re-enabled with the following command:

```
Switch (enable) set spantree enable [ all | module_number/port_number ]
```

If STP has been disabled on an IOS-based switch, it can be re-enabled with the following command:

```
Switch (config)# spantree vlan_list
```



You can use the `show spantree [vlan]` command to view the status of STP on either a CLI- or IOS-based switch.

The STA places each bridge/switch port in either a forwarding state or a blocking state. All the ports in forwarding state are considered to be in the current **spanning tree**. The collective set of forwarding ports creates a single path over which frames are sent between Ethernet segments. Switches can forward frames out ports and receive frames in ports that are in forwarding state; switches do not forward frames out ports and receive frames in ports that are in blocking state.

STP uses three criteria to choose whether to put an interface in forwarding state or a blocking state:

- STP elects a root bridge and puts all interfaces on the root bridge in forwarding state.
- Each nonroot bridge considers one of its ports to have the lowest administrative cost between itself and the root bridge. STP places this **lowest-root-cost interface**, called that bridge's **root port**, in forwarding state.
- Many bridges can attach to the same Ethernet segment. The bridge with the lowest administrative cost from itself to the root bridge, as compared with the other bridges attached to the same segment, is placed in forwarding state. The **lowest-cost bridge** on each segment is called the **designated bridge**, and that bridge's interface, attached to that segment, is called the **designated port**.

All other interfaces are placed in blocking state.

1.5.1 Root Bridge Election

For all switches in a network to agree on a loop-free topology, a common frame of reference must exist. This reference point is called the **Root Bridge**. The Root Bridge is chosen by an election process among all connected switches. Each switch has a unique **Bridge ID** that it uses to identify itself to other switches. The Bridge ID is an 8-byte value. 2 bytes of the Bridge ID is used for a **Bridge Priority** field, which is the priority or weight of a switch in relation to all other switches. The other 6 bytes of the Bridge ID is used for the **MAC Address** field, which can come from the Supervisor module, the backplane, or a pool of 1024 addresses that are assigned to every Supervisor or backplane depending on the switch model. This address is hardcoded, unique, and cannot be changed.

The election process begins with every switch sending out BPDUs with a Root Bridge ID equal to its own Bridge ID as well as a Sender Bridge ID. The latter is used to identify the source of the BPDU message. Received BPDU messages are analyzed for a lower Root Bridge ID value. If the BPDU message has a Root Bridge ID of the lower value than the switch's own Root Bridge ID, it replaces its own Root Bridge ID with the Root Bridge ID announced in the BPDU. If two Bridge Priority values are equal, then the lower MAC address takes preference. The switch is then nominates the new Root Bridge ID in its own BPDU messages although it will still identify itself as the Sender Bridge ID. Once the process has converged, all switches will agree on the Root Bridge until a new switch is added.

The Root Bridge election is based on the idea that one switch is chosen as a common reference point, and all other switches choose ports that are closest to the Root. The Root Bridge election is also based on the idea that the Root Bridge can become a central hub that interconnects other legs of the network. Therefore, the Root Bridge can be faced with heavy switching loads in its central location. If heavy loads of traffic are expected to pass through the Root Bridge, the slowest switch is not the ideal candidate. Furthermore, only

640-801 ICND 2.1 & INTRO

one Root Bridge is elected. This is thus not fault tolerant. To overcome these problems, you should set a Root Bridge in a determined fashion, and set a secondary Root Bridge in case of primary Root Bridge failure. The Root Bridge and the secondary Root Bridge should be placed near the center of the network.

To configure a CLI-based Catalyst switch to become the Root Bridge, use the following command to modify the Bridge Priority value so that a switch can be given a lower Bridge ID value to win a Root Bridge election:

```
Switch (enable) set spantree priority bridge_priority [ vlan ]
```

Alternatively, you can use the following command:

```
Switch (enable) set spantree root [ secondary ] [ vlan_list ]  
[ dia diameter ] [ hello hello_time ]
```

This command is a macro that executes several other commands. The result is a more direct and automatic way to force one switch to become the Root Bridge. Actual Bridge Priorities are not given in the command. Rather, the switch will modify STP values according to the current values in use within the active network.

To configure an IOS-based Catalyst switch to become the Root Bridge, use the following command to modify the Bridge Priority value so that a switch can be given a lower Bridge ID value to win a Root Bridge election:

```
Switch (config)# spanning-tree [ vlan vlan_list ] priority  
bridge_priority
```

1.5.2 Root Ports Election

Once a reference point has been nominated and elected for the entire switched network, each non-root switch must find its relation to the Root Bridge. This action can be performed by selecting only one Root Port on each non-root switch. STP uses the Root Path Cost to select a Root Port. The Root Path Cost is the cumulative cost of all the links leading to the Root Bridge. A particular switch link has a cost associated with it called the Port or Path Cost. This cost is inversely proportional to the port's bandwidth. As the Path Cost travels along, other switches can modify its value to make it cumulative. The Path Cost is known only to the local switch where the port or "path" to a neighboring switch resides as it is not contained in the BPDU. Only the Root Path Cost is contained in the BPDU. Path Costs are defined as a one-byte value.

The Root Bridge sends out a BPDU with a Root Path Cost value of zero because its ports sit directly on the Root Bridge. When the next closest neighbor receives the BPDU, it adds the Path Cost of its own port where the BPDU arrived. The neighbor then sends out BPDUs with this new cumulative value as the Root Path Cost. This value is incremented by subsequent switch port Path Costs as the BPDU is received by each switch on down the line. After incrementing the Root Path Cost, a switch also records the value in its memory. When a BPDU is received on another port and the new Root Path Cost is lower than the previously recorded value, this lower value becomes the new Root Path Cost. In addition, the lower cost tells the switch that the Root Bridge must be closer to this port than it was on other ports. The switch has now determined which of its ports is the closest to the root—the Root Port.

640-801 ICND 2.1 & INTRO

If desired, the cost of a port can be modified from the default value. However, changing one port's cost may influence STP to choose that port as a Root Port. Therefore careful calculation is required to ensure that the desired path will be elected. On a CLI-based switch, the port cost can be modified by using one of the following commands:

```
Switch (enable) set spantree portcost module_number/port_number cost
```

or

```
Switch (enable) set spantree portvlancost module_number/port_number  
[ cost cost ] [ vlan_list ]
```

On an IOS-based switch, the port cost for individual VLANs can be modified by using the following command:

```
Switch (config-if)# spanning-tree [ vlan vlan_list ] cost cost
```

1.5.3 Designated Ports Election

Once the Root Path Cost values have been computed, the Root Ports have been identified; however, all other links are still connected and could be active, leaving bridging loops. To remove the bridging loops, STP makes a final computation to identify one Designated Port on each network segment which would forward traffic to and from that segment. Switches choose a Designated Port based on the lowest cumulative Root Path Cost to the Root Bridge. All ports are still active and bridging loops are still possible. STP has a set of progressive states that each port must go through, regardless of the type or identification. These states will actively prevent loops from forming.

1.5.4 STP States

To participate in STP, each port of a switch must progress through several states. A port begins in a Disabled state moving through several passive states and finally into an active state if allowed to forward traffic. The STP port states are: Disabled, Blocking, Listening, Learning, and Forwarding.

- Ports that are administratively shut down by the network administrator or by the system due to a fault condition are in the **Disabled** state. This state is special and is not part of the normal STP progression for a port.
- After a port initializes, it begins in the **Blocking** state so that no bridging loops can form. In the Blocking state, a port cannot receive or transmit data and cannot add MAC addresses to its address table. Instead, a port is only allowed to receive BPDUs. Also, ports that are put into standby mode to remove a bridging loop enter the Blocking state.
- The port will be moved from the Blocking state to the **Listening** state if the switch thinks that the port can be selected as a Root Port or Designated Port. In the Listening state, the port still cannot send or receive data frames. However, the port is allowed to receive and send BPDUs so that it can actively participate in the Spanning-Tree topology process. Here the port is finally allowed to become a Root Port or Designated Port because the switch can advertise the port by sending BPDUs to other switches. Should the port lose its Root Port or Designated Port status, it is returned to the Blocking state.

640-801 ICND 2.1 & INTRO

- After a period of time called the **Forward Delay** in the Listening state, the port is allowed to move into the **Learning** state. The port still sends and receives BPDUs as before. In addition, the switch can now learn new MAC addresses to add into its address table.
- After another Forward Delay period in the Learning state, the port is allowed to move into the **Forwarding** state. The port can now send and receive data frames, collect MAC addresses into its address table, and send and receive BPDUs. The port is now a fully functioning switch port within the Spanning-Tree topology.

1.5.5 STP Timers

STP operates as switches send BPDUs to each other in an effort to form a loop-free topology. The BPDUs take a finite amount of time to travel from switch to switch. In addition, news of a topology change such as a link or Root Bridge failure can suffer from propagation delays as the announcement travels from one side of a network to the other. Because of the possibility of these delays, preventing the Spanning-Tree topology from converging until all switches have had time to receive accurate information is important. STP uses three timers for this purpose. There are three timers: Hello Time, Forward Delay, and Max Age.

- **Hello Time** is the time interval between Configuration BPDUs sent by the Root Bridge. The Hello Time value configured in the Root Bridge switch will determine the Hello Time for all non-root switches. However, all switches have a locally configured Hello Time that is used to time Topology Change Notification (TCN) BPDUs when they are retransmitted. The IEEE 802.1D standard specifies a default Hello Time value of two seconds.
- **Forward Delay** is the time interval that a switch port spends in both the Listening and Learning states. The default value is 15 seconds.
- **Max Age** is the time interval that a switch stores a BPDU before discarding it. While executing the STP, each switch port keeps a copy of the "best" BPDU that it has heard. If the source of the BPDU loses contact with the switch port, the switch will notice that a topology change has occurred after the Max Age time elapses and the BPDU is aged out. The default Max Age value is 20 seconds.

To announce a change in the active network topology, switches send a Topology Change Notification (TCN) BPDU. This occurs when a switch either moves a port into the **Forwarding** state or moves a port from **Forwarding** or **Learning** into the **Blocking** state. The switch sends a TCN BPDU out its **Designated Port**. The TCN BPDU carries no data about the change, but only informs recipients that a change has occurred. However, the switch will not send TCN BPDUs if the port has been configured with **PortFast enabled**. The switch will continue sending TCN BPDUs every **Hello Time** interval until it gets an acknowledgement from an upstream neighbor. As the upstream neighbors receive the TCN BPDU, they will propagate it on toward the **Root Bridge**. When the Root Bridge receives the BPDU, the Root Bridge sends out an acknowledgement. The Root Bridge also sends out the Topology Change flag in a Configuration BPDU so that all other bridges will shorten their bridge table aging times down from the default 300 seconds to the Forward Delay value. This condition causes the learned locations of MAC addresses to be flushed out sooner than they normally would, easing the bridge table corruption that might occur due to the change in topology. However, any stations that are actively communicating during this time will be kept in the bridge table. This condition lasts for the sum of the Forward Delay and the Max Age.

The three STP timers can be adjusted. These timers need only be modified on the Root Bridge and any secondary or backup Root Bridges because the Root Bridge propagates all three timer values throughout the network in the Configuration BPDU.

1.5.6 Optional STP Features

Cisco has added several proprietary enhancements to STP and to the logic used by its switches. Also, the IEEE, which owns the STP specifications, has made other enhancements, some similar to Cisco's proprietary enhancements.

1.5.6.1 EtherChannel

EtherChannel combines from two to eight parallel Ethernet trunks between the same pair of switches, bundled into an EtherChannel. STP treats an EtherChannel as a single link, so if at least one of the links is up, STP convergence does not have to occur. With each pair of Ethernet links configured as an EtherChannel, STP treats each EtherChannel as a single link. Thus, both links to the same switch must fail for a switch to need to cause STP convergence. Without EtherChannel, if you have multiple parallel links between two switches, STP blocks all the links except one. With EtherChannel, all the parallel links can be up and working at the same time, while reducing the number of times STP must converge, which in turn makes the network more available.

EtherChannel also provides more network bandwidth. All trunks in an EtherChannel are either forwarding or blocking, because STP treats all the trunks in the same EtherChannel as one trunk. When an EtherChannel is in forwarding state, the switches forward traffic over all the trunks, providing more bandwidth.

1.5.6.2 PortFast

PortFast allows a switch to place a port in forwarding state immediately when the port becomes physically active. However, the only ports on which you can safely enable PortFast are ports on which you know that no bridges, switches, or other STP devices are connected. Thus, PortFast is most appropriate for connections to end-user devices. If you turn on PortFast for end-user devices, when an end-user PC boots, as soon as the Ethernet card is active, the switch port can forward traffic. Without PortFast, each port must wait **MaxAge** plus twice **Forwarding Delay**, which is 50 seconds with the default **MaxAge** and **Forward Delay** settings.

1.5.6.3 Rapid Spanning Tree (IEEE 802.1w)

The IEEE has improved the 802.1d protocol, which defines STP, with the definition of **Rapid Spanning Tree Protocol (RSTP)**, as defined in standard 802.1w. RSTP is similar to STP in that it elects the root switch using the same parameters and tiebreakers; elects the root port on nonroot switches with the same rules; elects designated ports on each LAN segment with the same rules; and places each port in either a forwarding state or a blocking state, with the latter being called the discarding state instead of the blocking state.

RSTP can be deployed alongside traditional STP bridges and switches, with RSTP features working in switches that support it, and STP features working in the switches that support only STP.

The advantage RSTP has over STP is improved network convergence when network topology changes occur. STP convergence has essentially wait periods: a switch must first cease to receive root BPDUs for **MaxAge** seconds before it can begin to transition any interfaces from blocking to forwarding. For any interfaces that need to transition from blocking to forwarding, the interface must endure **Forward Delay** seconds in

640-801 ICND 2.1 & INTRO

listening state and **Forward Delay** more seconds in learning state before being placed in forwarding state. By default, these three wait periods of are 20, 15, and 15 seconds.

RSTP convergence times typically take less than 10 seconds. In some cases, they can be as low as 1 to 2 seconds.