Internetwork Design Guide

September 1994

The products and specifications, configurations, and other technical information regarding the products contained in this manual are subject to change without notice. All statements, technical information, and recommendations contained in this manual are believed to be accurate and reliable but are presented without warranty of any kind, express or implied, and users must take full responsibility for their application of any products specified in this manual.

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

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

The following are trademarks of Digital Equipment Corporation: DDCMP, DEC, DECnet, DECNIS, DECserver, DECsystem, DECwindows, Digital, DNA, OpenVMS, ULTRIX, VAX, VAXstation, VMS, VMScluster, and the DIGITAL logo.

Portions of this document is used with permission of Cisco Systems, Incorporated. Copyright © 1990 - 1995, Cisco Systems, Inc.

The following third-party software may be included with your product and will be subject to the software license agreement:

CiscoWorks software and documentation are based in part on HP OpenView under license from the Hewlett-Packard Company. HP OpenView is a trademark of the Hewlett-Packard Company. Copyright © 1992, 1993 Hewlett-Packard Company.

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

Network Time Protocol (NTP). Copyright © 1992, David L. Mills. The University of Delaware makes no representations about the suitability of this software for any purpose.

TABLE OF CONTENTS

Chapter 2

Chapter 3

LIST OF FIGURES

LIST OF TABLES

About This Guide

This section discusses the objectives, audience, organization, and conventions of the *Internetwork Design Guide*.

Document Objectives

This guide presents a set of general guidelines for planning internetworks and provides specific suggestions for several key internetworking implementations. This guide focuses on design issues of large-scale implementations for the following environments:

- Large-scale Internetwork Protocol (IP) internetworks
  - Enhanced Interior Gateway Routing Protocol (IGRP) design
  - Open Short Path First (OSPF) design
- IBM System Network Architecture (SNA) internetworks
  - Source-route bridging (SRB) design
  - Synchronous Data Link Control (SDLC) and serial tunneling (STUN), SDLC Logical Link Control type 2 (SDLLC), and Qualified Logical Link Control (QLLC) design
- Asynchronous Transfer Mode (ATM) internetworks
- Packet service internetworks
  - Frame Relay design
- Dial-on-demand routing (DDR) internetworks

Note The term *router* is used throughout this guide to refer to internetworking devices that also offer bridging and gateway functions. Routers are sometimes called *intermediate systems*. End stations are also called *end systems*.

The objective of this guide is to help you identify and implement *practical* internetworking strategies that are flexible enough to fit a variety of situations and that can scale up as your network requirements change. The *Internetwork Design Guide* focuses on identifying the essential technologies and appropriate implementations for specific environments. It is not the final word in internetwork design. Do not try to use this as a step-by-step handbook for designing every facet of your internetwork.

This guide is not a *network* design guide. In other words, it is not a comprehensive encyclopedia of network design strategy. The emphasis is not on issues such as maximum cable runs or the relative merits of IEEE 10BaseT and thin Ethernet. Instead, the *Internetwork Design Guide* is a tool for identifying router features and capabilities that meet specific internetworking requirements.

The central elements of this guide are the technology chapters which consist of three chief elements:

- Technology-specific issues
- Router-related implications of design implementation
- Implementation recommendations

The technology chapters do not cover every possible implementation, but address a variety of environments that are commonly encountered when designing internetworks.

Note The *Internetworking Applications: Case Studies* publications are companion guides to this design guide. Case studies provide internetwork scenarios with detailed configuration examples for specific Cisco features.

Audience

This guide addresses the network administrator who designs and implements router-based internetworks. Readers should know how to configure a Cisco router and should be familiar with the protocols and media that their routers have been configured to support. Awareness of basic network topology is essential.

Document Organization

This document consists of the following major topic areas:

- Introductory material in the first chapter outlines the key issues in designing effective large-scale internetworks, contrasts bridging and routing, and describes the three key service layers associated with internetworks: access, distribution, and backbone. The introduction also provides a general mapping of feature capabilities into this hierarchical approach to internetwork design.
- Chapters 2 through 7 focus on the specific design recommendations for technologies covered in this design guide:
  - Enhanced IGRP and OSPF internetworks
  - Source-route bridging (SRB)
  - SDLC, SDLLC, STUN, and QLLC conversion
  - ATM internetworks
  - Frame Relay internetworks
  - DDR internetworks
- Appendixes in this document provide supplemental information.

Document Conventions

This guide uses the following conventions:

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

Caution Means *reader be careful*. It means that you are capable of doing something that might result in equipment damage, or that you might have to take something apart and start over again.

Command descriptions use these conventions:

- Commands and keywords are in **boldface** font.
- Arguments for which you supply values are in *italic* font.

Examples use these conventions:

- Examples that contain system prompts denote interactive sessions, indicating that the user enters commands at the prompt. The system prompt indicates the current command mode. For example, the prompt `router(config)#` indicates global configuration mode.
- Terminal sessions and information the system displays are in `screen` font.
- Exclamation points (!) at the beginning of a line indicate a comment line.

CHAPTER 1

Internetworking Design Basics

Designing an internetwork can be a daunting task. An internetwork consisting of only 50 meshed routing nodes can pose complex problems that lead to unpredictable results. Attempting to optimize internetworks featuring thousands of nodes can be overwhelming.

Despite improvements in equipment performance and media capabilities, internetwork design is not getting any easier. Indeed, these changes might be making design more difficult than ever. The trend is toward increasingly complex environments involving multiple media, multiple protocols, and interconnection to networks outside any single organization's dominion of control. Carefully designing internetworks can reduce the hardships associated with growth as a networking environment evolves.

This chapter provides an overview of planning and design guidelines. Discussions are divided into the following general topics:

- Determining your internetworking requirements
- Identifying and selecting internetworking capabilities
- Choosing internetworking reliability options

Determining Your Internetworking Requirements

Routers and other internetworking devices must reflect the goals, characteristics, and policies of the organizations in which they operate.

Two primary goals drive internetworking design and implementation:

- Application availability—Networks exist to carry application information between computers. If the applications are not available to network users, the network is not doing its job.
- Cost of ownership—Information system (IS) budgets today often run in the millions of dollars. As large organizations increasingly rely on electronic data for managing business activities, the associated costs of computing resources will continue to rise.

A well-designed internetwork can help to balance these objectives. When properly implemented, routers can optimize application availability while allowing for the cost-effective use of existing network resources.

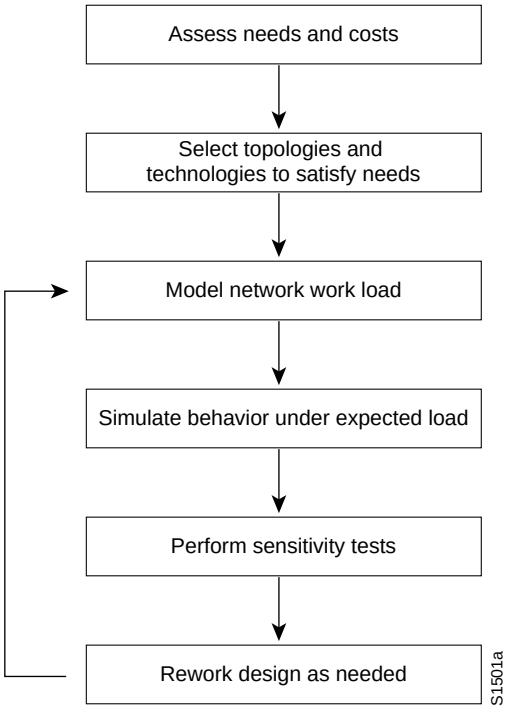
The Design Problem: Optimizing Availability and Cost

In general, the network design problem can be characterized as consisting of three general elements: environmental *givens*, performance *constraints*, and internetworking *variables*. The goal is to minimize cost based on these parameters and variables while delivering overall service that does not compromise established availability requirements. Each element of this problem consists of several components. Environmental givens include the location of hosts, servers, terminals and other end nodes; the projected traffic for the environment; and the projected costs for delivering different service levels. Performance constraints consist essentially of network reliability and traffic throughput. Internetworking variables include the network topology, line capacities, and packet flow assignments.

The two primary concerns facing internetwork designers— availability and cost—are essentially at odds. Any increase in availability must generally be reflected as an increase in cost. As a result, network designers must weigh the relative importance of resource availability and overall cost carefully.

Designing your network is an iterative, largely trial-and-error activity. Figure 1-1 illustrates the basic process associated with internetwork design. The discussions that follow outline several areas that require careful consideration when planning your internetworking implementation.

Figure 1-1 General Network Design Process



Assessing User Requirements

In general, users are concerned with one thing when it comes to networking: application availability. The chief components of application availability are *response time*, *throughput*, and *reliability*.

Response time is the time between entry of a command or keystroke and the host system's execution of the command, or delivery of a response. User satisfaction about response time is generally considered to be a *monotonic* function up to some limit, at which point user satisfaction falls off to nearly zero. Applications where fast response time is considered critical include interactive online services such as automated tellers and point-of-sale machines.

Applications that put high-volume traffic onto the network have more effect on throughput than end-to-end connections. Throughput-intensive applications generally involve file-transfer activities. However, throughput-intensive applications also usually have low response-time requirements. Indeed, they can often be scheduled at times when response-time-sensitive traffic is low (for example, after normal work hours).

Although reliability is always important, some applications have genuine requirements that exceed typical needs. Organizations where all activities are conducted online or over the telephone often require nearly 100 percent uptime. Financial services, securities exchanges, and emergency/police/military operations are a few examples. These situations imply a requirement for a high level of hardware and topological redundancy. Determining the cost of any downtime is essential in determining the relative importance of reliability to your internetwork.

You can assess user requirements in a number of ways. The more involved your users are in the process, the more likely that your evaluation will be accurate. In general, you can use the following methods to get information:

- **User community profiles**—Using your experience, outline what different user groups will require. This is the first step in determining what will be required of an internetwork. Although everyone will have roughly the same requirements of an electronic mail system, engineering groups using X Window terminals and Sun workstations in an NFS environment will have different needs from PC users sharing print servers in a finance department.
- **Interviews, focus groups, and surveys**—Your intuition might serve to build a baseline for implementing an internetwork, but every situation is unique in one way or another. Particular groups might require access to common servers, certain groups might want to allow external access to specific internal computing resources, or certain organizations might require that IS support systems be managed in a particular way according to some external standard. In any event, there might be a range of special requirements that must be evaluated on a case-by-case basis. The least-formal method of obtaining information is to conduct interviews with key user groups. Focus groups can also be used to gather information and generate discussion among different organizations with similar (or opposing) interests. Finally, formal surveys can be used to get a statistically valid reading of user sentiment regarding a particular service level or proposed internetworking architecture.
- **Human factors tests**—The most expensive, time-consuming, and possibly the most revealing method is to conduct a test involving representative users in a lab environment. This is most applicable when evaluating response time requirements. As an example, you might set up working systems and have users perform normal remote host activities from the lab network. By evaluating user reactions to variations in host responsiveness, you can create benchmark thresholds for acceptable performance.

Compatibility, Conformance, and Interoperability

Compatibility, conformance, and interoperability are related to the problem of balancing proprietary functionality and open internetworking flexibility. As a network designer, you might be forced to choose between implementing a multivendor environment and implementing a specific, proprietary capability.

For example, the Interior Gateway Routing Protocol (IGRP) provides many useful capabilities, such as fast convergence and efficient route handling in large internetworks, but it is a proprietary routing protocol. In contrast, the integrated Intermediate System-to-Intermediate System (IS-IS) protocol is an open internetworking alternative that also provides a fast converging routing environment; however, implementing an open routing protocol can potentially result in greater multivendor configuration complexity.

The protocol that you choose will have far-ranging effects on your overall internetwork design. Assume that you decide to implement integrated IS-IS instead of IGRP. In doing this, you gain a measure of interoperability; however, you lose some functionality. For instance, you will not be able to load balance traffic over unequal parallel paths. Similarly, some modems provide a high level of proprietary diagnostic capabilities, but require that all modems throughout a network be of the same vendor type to fully exploit proprietary diagnostics.

The internetworking decisions you make are not made in vacuum. Previous internetworking (and networking) investments and expectations for future requirements have considerable influence on the kinds of implementations you pursue today. Things to consider include installed internetworking and networking equipment; applications running (or to be run) on the network; physical location of sites, hosts, and users; rate or growth of user community; and both physical and logical network layout.

Assessing Costs

The internetwork is a strategic element in your overall information system design. As such, the cost of your internetwork is much more than the sum of your router purchase orders. View it as a total cost-of-ownership issue. You must consider the entire life cycle of your internetworking environment. A brief list of costs associated with internetworks follows:

- **Router hardware and software costs**—Consider what is really being bought when you purchase your systems; costs should include initial purchase and installation, maintenance, and projected upgrade costs.
- **Performance tradeoff costs**—Consider the cost of going from a five-second response time to a half-second response time. Such improvements can cost quite a bit in terms of media selection, network interfaces, internetworking nodes, modems, and wide-area network (WAN) services.
- **Installation costs**—Installing a site's physical cable plant is by far the most expensive element of a large network. The costs include installation labor, site modification, fees associated with local code conformance, and costs incurred to ensure compliance with environmental restrictions (such as asbestos removal). Other important elements in keeping your costs to a minimum will include developing a well-planned wiring closet layout and implementing color code conventions for cable runs.
- **Expansion costs**—Calculate the cost of ripping out all that thick Ethernet, adding additional functionality, or moving to a new location. Although there is no way to know the future, projecting your future requirements and building in accommodations for future needs will save you time and money.
- **Support costs**—More complicated internetworks cost more to monitor, configure, and maintain. Your internetwork should be no more complicated than necessary. Costs include training, direct labor (network managers/administrators), sparing, and replacement costs.

- **Cost of downtime**—Evaluate the cost for every minute that a user is unable to access a file server or a centralized database. If this cost is high, you must attribute a high cost to downtime. If the cost is high enough, fully redundant internetworks might be your only option.
- **Opportunity costs**—Every choice you make will have an opposing alternative option. Whether that option is a specific hardware platform, topology solution, level of redundancy, or system integration alternative, there are always options. *Opportunity costs* are the costs of not picking one of those options. The opportunity costs of not switching to newer technologies and topologies might be lost competitive advantage, lower productivity, and slower overall performance. These are difficult to quantify, but any effort to integrate opportunity costs into your analysis can help in making accurate comparisons at the beginning of your project.
- **Sunken costs**—What you have invested in existing cable plant, routers, concentrators, hosts, and other equipment and software are your *sunken costs*. If the sunken cost is high, you might be forced to modify networks so that your existing internetwork can continue to be exploited. Although comparatively low incremental costs might appear to be more attractive than significant redesign costs, your organization might pay more in the long run by not upgrading systems. Over-reliance on sunken costs when calculating the cost of internetwork modifications and additions can cost your organization sales and market share.

Estimating Traffic: Work-Load Modeling

Empirical *work-load modeling* consists of instrumenting a working internetwork and monitoring traffic for a given number of users, applications, and network topology. Try to characterize activity throughout a normal work day in terms of type of traffic passed, level of traffic, response time of hosts, time to execute file transfers, and so on. You can also observe utilization on existing routers over the test period with the router's **show** and **debug** commands.

If the tested internetwork's characteristics are close to the new internetwork, you can try extrapolating to the new internetwork's number of users, applications, and topology. This is unquestionably a *best-guess* approach to traffic estimation. However, with no widely available tools for characterizing behavior in a dynamically routed environment, it might be your best bet.

In addition to passive monitoring of an existing network, you can measure activity and traffic generated by a known number of users attached to a representative test network and then extrapolate findings to your anticipated population.

One problem with modeling work loads on networks is that it is difficult to accurately pinpoint traffic load and network device performance as functions of the number of users, type of application, and geographical location. This is especially true without a real network in place.You should consider the following factors that influence the dynamics of the network:

- The time-dependent nature of network access—Peak periods can vary; measurements must reflect a range of observations that includes peak demand.
- Differences associated with type of traffic—Routed and bridged traffic place different demands on internetwork devices and protocols; some protocols are sensitive to dropped packets; some application types require more bandwidth.
- The random (nondeterministic) nature of network traffic—Exact arrival time and specific effects of traffic are unpredictable.
- Competing protocols—In multiprotocol environments, routers undergo dynamic demands from different users, resulting in unpredictable effects.

Sensitivity Testing

From a practical point of view, sensitivity testing involves breaking stable links and observing what happens. When working with a test network, this is relatively easy. Perturb the network by removing an active interface and monitor how the change is handled by the internetwork: how traffic is rerouted, the speed of convergence, whether any connectivity is lost, and whether problems arise in handling specific types of traffic. You can also change the level of traffic on a network to determine the effects on the network when traffic levels approach media saturation. This empirical testing is a type of *regression* testing: a series of specific modifications (tests) are repeated on different versions of network configurations. By monitoring the effects on the design variations, you can characterize the relative resilience of the design.

Modeling sensitivity tests using a computer is beyond the scope of this publication. One source for more information about computer-based network design and simulation is Andrew S.Tannenbaum's book *Computer Networks.*

Identifying and Selecting Internetworking Capabilities

Once you understand your internetworking requirements, you must identify and then select the specific capabilities that fit your computing environment. The following discussions provide a starting point for making these decisions. Three topics are addressed:

- Contrasting bridging and routing capabilities
- General hierarchical model for internetworking
- Capabilities associated with backbone, distribution, and local-access services

---

Note This material presents an introduction to the primary router capabilities and outlines how each fulfills various internetworking requirements. The technology chapters that follow, addressing routing protocol implementation, IBM internetworking, and packet-service internetworking, present detailed discussions about specific implementations of large-scale internetworking.

Contrasting Bridging and Routing Capabilities

Data communications experts generally agree that bridges and routers are moving away from once-clear distinctions between the two technologies and converging toward the all-in-one brouter, routing bridge, or router/bridge. Performance enhancements are making the question of which is better (bridges or routers) an arcane and sometimes moot point. There are specific situations when routing is preferred, when bridging is preferred, and when you have little or no choice in the matter. The discussion that follows outlines the key criteria to use when determining which technology best suits your situation.

Bridging and Routing Definitions

Before bridging and routing capabilities can be contrasted, you must understand where each falls within a common framework of internetworking terminology. For this comparison, the bridging discussion focuses on transparent bridging. Transparent bridging is emphasized here because source-route bridging (SRB) can be argued to have more in common with routing than with bridging. Refer to the *Internetworking Technology Overview* for more information about each of these technologies.

By convention, bridging is said to occur at the data link layer, while routing is said to occur at the network layer of the International Organization for Standardization's (ISO's) seven-layer protocol model. In general, data link layer devices assume a common logical network (information traverses a single hop to reach a destination). Network layer devices are designed to handle multiple hops and multiple networks. These distinctions lead to certain constraints for bridging that result in four important differences between routers and bridges:

- The header associated with data link packets lacks information fields present in network layer packets. Examples of fields provided in network layer packets include final destination address, hop count, and fragmentation and reassembly information.
- Bridges do not support handshaking protocols, such as the Internet Control Message Protocol (ICMP) associated with the Internet Protocol (IP), used by end nodes and routers to learn about each other.
- Bridges cannot reorder packets from the same source; network layer protocols expect some degree of reordering (caused by fragmentation).
- Bridges use the Media Access Control (MAC) addresses defined at the time of equipment manufacture to identify an end node. Thus, the address has no topological meaning. With routers, a network address is associated with the local-area network (LAN) to which a particular node is attached.

Despite the constraints associated with bridging, there are often situations that require the use of bridging technology. Similarly, routing might be necessary to ensure proper segmentation of traffic or to support a specific topology that does not permit a single logical internetwork. The following sections summarize common reasons for choosing routing or bridging.

Routing Advantages

Routing offers the following advantages over bridging:

- Routers can choose the best path that exists between source and destination; bridges are limited to a specific path (referred to as a spanning tree) through an internetwork. Two characteristics of bridges result in this difference: bridges must learn the location of stations based on the direction from which traffic is received, and bridges are transparent (in other words, they are not permitted to modify a packet in any way). These characteristics do not apply to source-route bridges. Unlike transparent bridges, source-route bridges do not maintain a station location table. In addition, loops are not possible in an SRB environment. The SRB standard states that an SRB must verify that the output network segment has not been traversed before transmitting packets.
- Routers reconfigure topology after changes much more quickly than bridges, resulting in reduced service loss. Because bridges are transparent, loops pose a substantially greater risk to a bridged internetwork than with a routed internetwork. Thus, new bridge paths are recognized gradually, while new routing paths are recognized as soon as routing information is received.
- The total number of stations supportable in a routed internetwork is virtually unlimited, particularly for ISO Connectionless Network Service (CLNS) internetworks, while the maximum number of stations in a bridged internetwork is constrained to thousands of end stations. Routers can accommodate a much larger address space because network layer addresses include information that groups nodes into areas or domains. Bridges have no hierarchical addressing component and cannot direct traffic in a hierarchical manner.
- Routers can provide a barrier against broadcast storms; bridges cannot. By design, when bridges join a series of LAN segments, those segments form a single LAN from the perspective of upper-layer protocols. A broadcast storm disables the entire bridged internetwork because all the bridges forward the broadcast traffic throughout the internetwork and are unable to intervene. Routers block broadcasts by default.

- Routers fragment and reassemble large packets; bridges drop packets that are too big to forward. For some protocols, the network layer header includes fragmentation and reassembly information; the data link layer header does not. As a result, bridges simply drop packets that are too large to forward. In addition, bridges are unable to inform the source that the packet was dropped.
- Routers provide congestion feedback to end stations when traffic is heavy; bridges do not. In ISO CLNS, mechanisms in the network layer protocols provide congestion-based information that can be relayed to source nodes to force them to reduce transmission rates. The data link layer provides no analogous capability.

Note In general, the preceding discussion concerning routing advantages assumes a comparison with transparent bridging. These advantages do no apply to SRB. However, one weakness of SRB is that it does not provide any form of dynamic rerouting. This is left to the end stations.

Bridging Advantages

If bridging did not have certain real advantages, the world's router bigots would have surely eradicated every bridging implementation from the face of the earth. Consider the following when choosing between routing or bridging:

- Bridges require minimal configuration; routers require maximum configuration. There is no getting around at least some configuration required for routers. For example, IP routers require the configuration of separate addresses for each interface and substantial configuration of end nodes (addresses and masks). In some situations, basic learning bridges require virtually no configuration. To become operational, you can simply take the bridge out of the box, power it up, and attach it to a network.
- Bridges have a better price-to-performance ratio than routers. With less overhead to handle, bridges have enjoyed an advantage over routers in terms of pure packet traffic handling. However, because router performance has improved significantly (now providing near wire speed performance) and the price difference between routers and bridges has eroded, this advantage is diminishing.
- Bridges are protocol independent; routers are protocol dependent. As routers have become capable of handling multiple protocols as either *integrated* or *ships-in-the-night* environments (both of which are discussed in the section "Backbone Routing Options" later in this chapter), this perceived advantage is also diminishing. Nonetheless, bridges can handle multiple protocols with almost no configuration.
- Bridges forward nonroutable protocols, such as Digital's Local Area Transport (LAT); routers cannot. Some protocols are not routable, even through they were designed to provide upper-layer functionality (just not between LANs). This remains a compelling reason for implementing bridging capabilities for supporting certain end-to-end connectivity.

Integrated Solutions

The trend in internetworking is to provide network designers greater flexibility in solving multiple internetworking problems without creating multiple networks or writing off existing data communications investments.

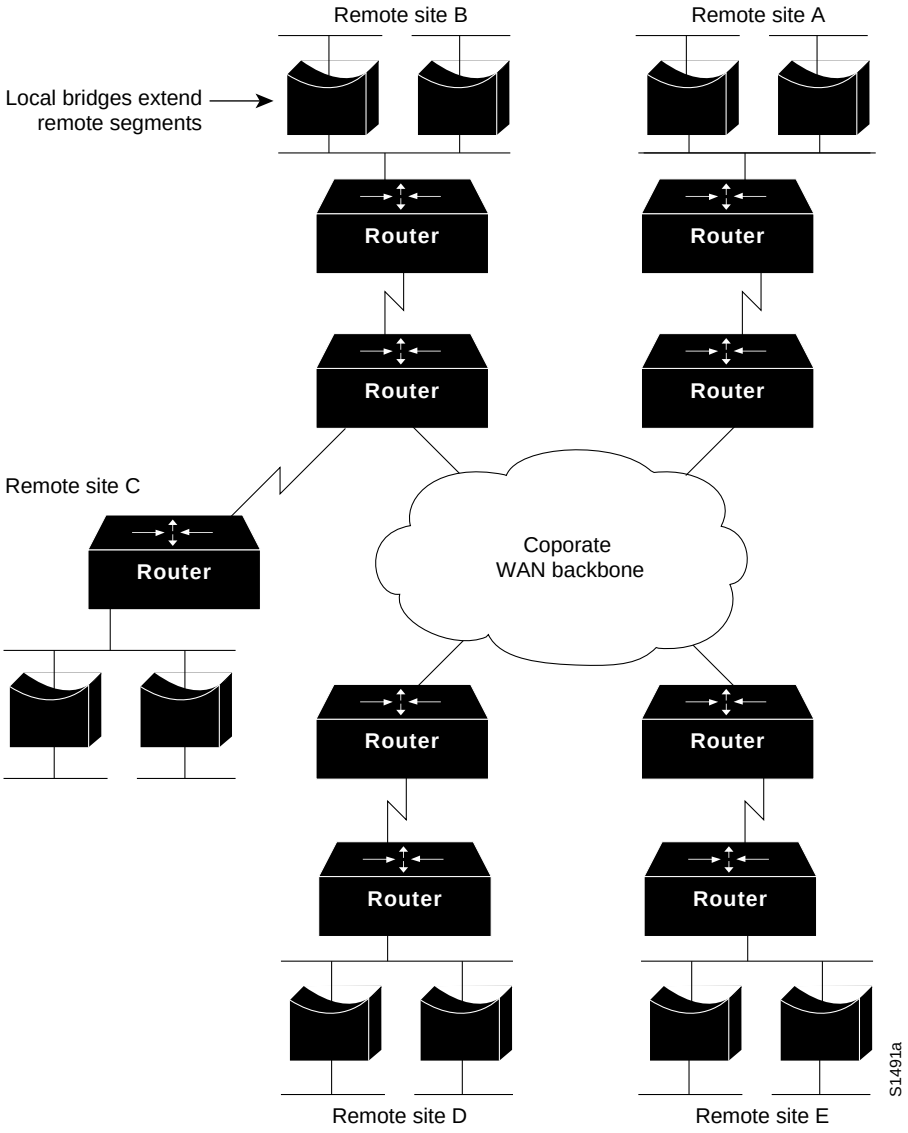
A network designer might employ bridges in a remote site for their ease of implementation, simple topology, and low traffic requirements. Routers might be relied upon to provide a reliable, self-healing backbone, as well as a barrier against inadvertent broadcast storms in the local networks.

There are a number of circumstances in which you can employ a dedicated bridging implementation for simple internetworks (often remote sites). For example, an IGS can be implemented for local bridging at a remote site, while providing the option of routing as the remote network evolves. If you are interconnecting the remote site into a corporate backbone, it is best to implement routing at the point of access from the local internetwork to the backbone.

If you have a very large, meshed local internetwork (for instance a campus consisting of several buildings and running protocols that derive significant benefits from routing), routers provide superior segmentation and more efficient traffic handling than bridges.

Figure 1-2 illustrates an environment featuring a WAN backbone interconnecting remote sites to a corporate internetwork. You can think of bridges implemented in this example as very smart repeaters.

Figure 1-2 Hybrid Internetwork Featuring Bridging and Routing Nodes



Backbone Routing Options

In an ideal world, the perfect enterprise-wide internetwork would feature a single, bullet-proof network protocol capable of transporting all manner of data communications seamlessly, error free, and with sufficient resilience to accommodate any unforeseen connectivity disruption. So much for the ideal world. The real world consists of many protocols with varying levels of resilience.

In designing a backbone for your organization, you might be considering several options. These options typically split into two primary categories:

- Multiprotocol routing backbone
- Single-protocol backbone

The following discussions outline the characteristics and properties of these two strategies.

Multiprotocol Routing Backbone

When multiple network layer protocols are routed throughout a common backbone without encapsulation (also referred to as *native* mode routing), the environment is referred to as a multiprotocol routing backbone. A multiprotocol backbone environment can adopt one of two routing strategies, or both, depending on the routed protocol involved. The two strategies are generally referred to as *integrated routing* and *ships in the night.*

Integrated routing involves the use of a single routing protocol (for example, a link state protocol) that determines the least cost path for different routed protocols.

The ships-in-the-night approach involves the use of a different routing protocol for each network protocol. For instance, some large-scale networks might feature multiple protocols, where Novell IPX traffic is routed using a proprietary version of the Routing Information Protocol (RIP), IP is routed with IGRP, and DECnet Phase V traffic is routed via ISO CLNS-compliant IS-IS. Each of these network layer protocols is routed independently, with separate routing processes handling their traffic and separate paths calculated.

Mixing routers within an internetwork that supports different combinations of multiple protocols can create a confusing situation, particularly for integrated routing. In general, integrated routing is easier to manage if all the routers attached to the integrated routing backbone support the same integrated routing scheme. Routes for other protocols can be calculated separately. As an alternative, you can use encapsulation to transmit traffic over routers that do not support a particular protocol.

Single-Protocol Backbone

With a single-protocol backbone, all routers are assumed to support a single routing protocol for a single network protocol. In this kind of routing environment, all other routing protocols are ignored. If multiple protocols are to be passed over the internetwork, unsupported protocols must be encapsulated within the supported protocol or they will be ignored by the routing nodes.

Why implement a single-protocol backbone? If relatively few other protocols are supported at a limited number of isolated locations, it is reasonable to implement a single protocol backbone. However, encapsulation does add overhead to traffic on the network. If multiple protocols are supported widely throughout a large internetwork, a multiprotocol backbone approach is likely to work better.

In general, you should support all the network layer protocols in an internetwork with a native routing solution and implement as few network layer protocols as possible.

Hierarchical Internetworking Model

Most internetworks can be hierarchically divided into three logical services: backbone, distribution, and local-access internetworks. *Backbone* (or *core*) services aim to optimize communication among routers at different sites or in different logical groupings. *Distribution* services provide a way to implement policy-based traffic control to isolate backbone and local environments. *Local-access* services support communication between end stations and routers.

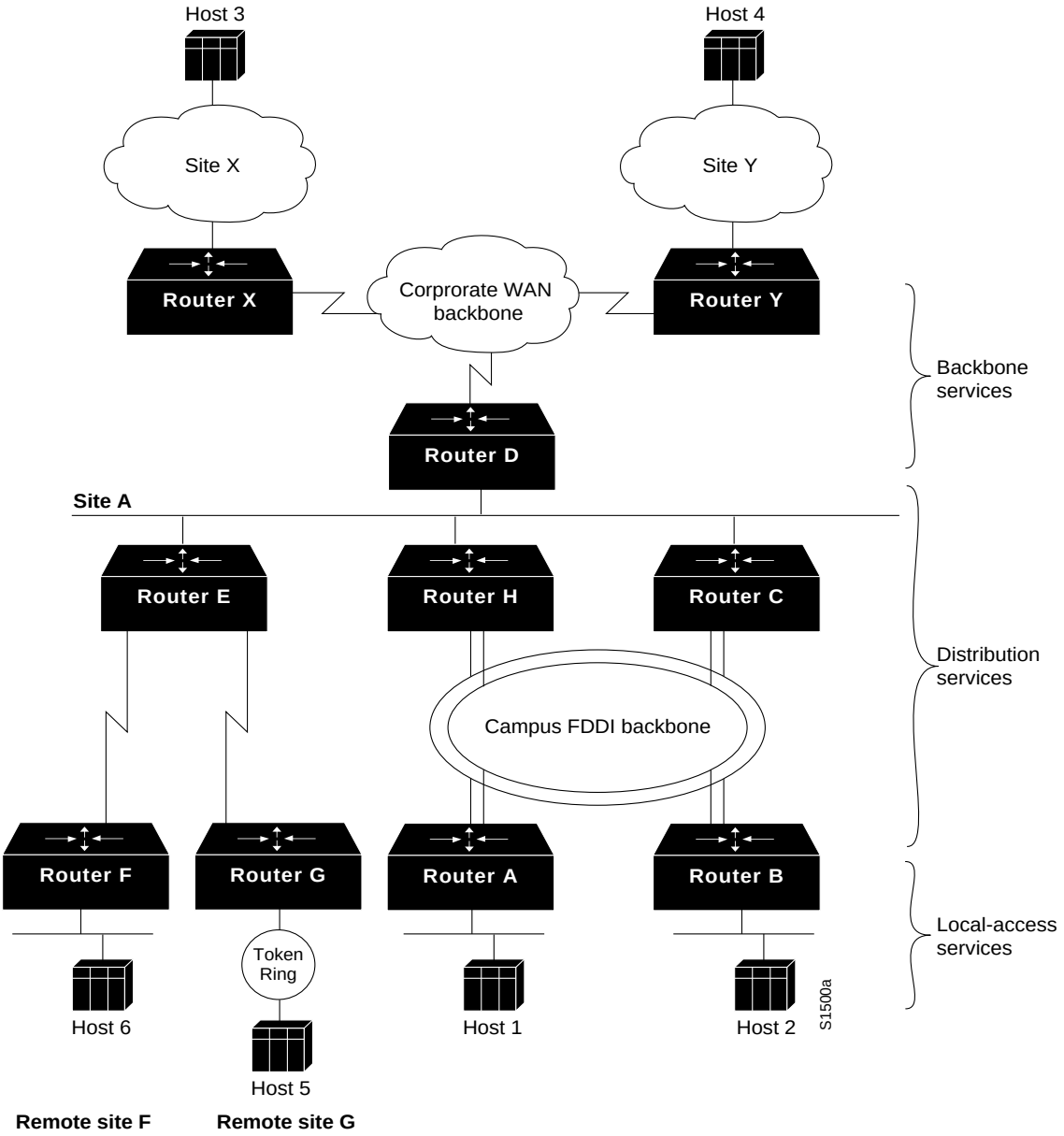
To illustrate the differences between backbone, distribution, and local-access services, consider Figure 1-3. Communication between Host 1 and Router A, or between Host 5 and Router G, is aided by local-access services. Communication between Router D and Router E, or between Router D and Router C, is aided by distribution services. Communication between Router D, Router Y, and Router X is assisted by backbone services.

Assume Host 3 and Host 2 need to communicate. Such a transmission could be routed as follows: Host 3 (through some arbitrary internetworking topology) to Router X to Router D to Router C to Router B to Host 2. The part of the route between Router X and Router D travels through a portion

of the backbone network. Communication between Router D and Router C (and subsequently Router B) is controlled by policy-based rules associated with distribution services. A host and an adjacent router (such as Host 2 and Router B) communicate using local-access services; controlling access to resources on other networks is the primary goal of these local-access routers.

The discussions that follow outline the capabilities and services associated with backbone, distribution, and local access internetworking services.

Figure 1-3 Backbone, Distribution, and Local-Access Router Environment



Evaluating Backbone Services

This section addresses internetworking features that support backbone services. The following topics are discussed:

- Backbone bandwidth management
- Path optimization
- Traffic prioritization
- Load balancing
- Alternative paths
- Switched access
- Encapsulation (tunneling)

Backbone Bandwidth Management

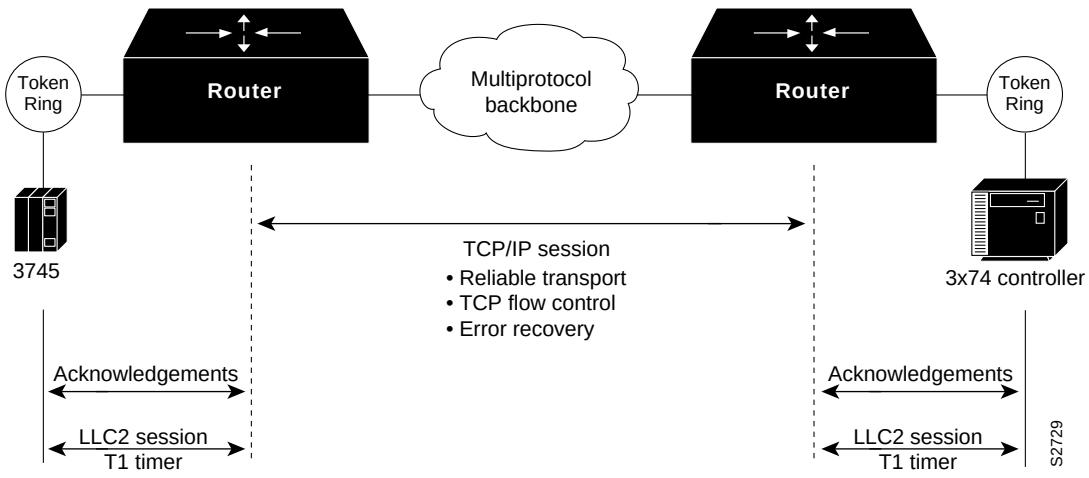
To optimize backbone network operations, routers offer several performance tuning features. Examples include priority queuing, routing protocol metrics, and local session termination.

You can adjust the output queue length on priority queues. If a priority queue overflows, excess packets are discarded and quench messages that halt packet flow are sent, if appropriate, for that protocol.

You can also adjust routing metrics to increase control over the paths traffic takes through the internetwork.

Local session termination allows routers to act as proxies for remote systems representing session endpoints. (A proxy is a device that acts on behalf of another device.) Figure 1-4 illustrates an example of local session termination in an IBM environment.

Figure 1-4 Local Session Termination over Multiprotocol Backbone



In Figure 1-4, the routers locally terminate Logical Link Control type 2 (LLC2) data link control sessions. Instead of end-to-end sessions where all session control information is passed over the multiprotocol backbone, the routers take responsibility for acknowledging packets coming from hosts on directly attached LANs. Local acknowledgment saves WAN bandwidth (and, therefore, WAN utilization costs), solves session timeout problems, and provides faster response to users.

Path Optimization

One of the primary advantages of a router is its ability to help you implement a logical environment in which optimal paths for traffic are automatically selected. Routers rely on routing protocols associated with the various network layer protocols to accomplish this automated path optimization.

Depending on the network protocols implemented, routers permit you to implement routing environments that suit your specific requirements. For example, in an IP internetwork, Cisco routers can support all widely implemented routing protocols, including Open Shortest Path First (OSPF), RIP, IGRP, Border Gateway Protocol (BGP), Exterior Gateway Protocol (EGP), and HELLO. Key built-in capabilities that promote path optimization include: rapid and controllable route convergence and tunable routing metrics and timers.

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

Many different metrics are used in routing algorithms. Some sophisticated routing algorithms base route selection on a combination of multiple metrics, resulting in the calculation of a single hybrid metric. IGRP uses one of the most sophisticated distance vector routing algorithms. It combines values for bandwidth, load, and delay to create a composite metric value. Link state routing protocols, such as OSPF and IS-IS, employ a metric that represents the cost associated with a given path.

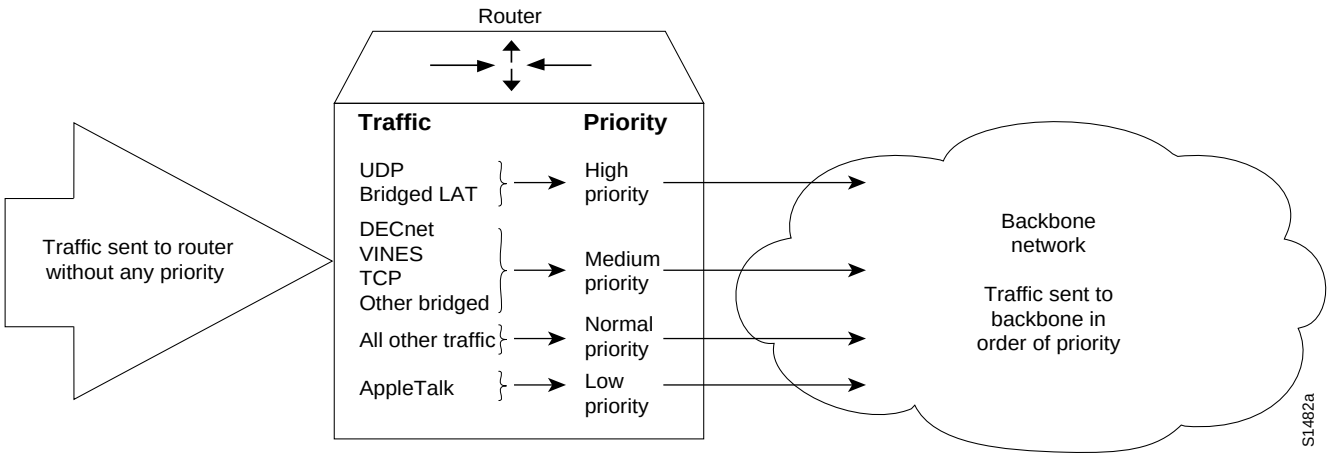
Traffic Prioritization

Although some network protocols can prioritize internal homogeneous traffic, the router prioritizes the heterogeneous traffic flows. Such traffic prioritization enables policy-based routing and ensures that protocols carrying mission-critical data take precedence over less-important traffic.

Priority output queuing allows the network administrator to prioritize traffic. Traffic can be classified according to various criteria, including protocol and subprotocol type, and then queued on one of four output queues (high, medium, normal, or low priority). For IP traffic, additional fine-tuning is possible.

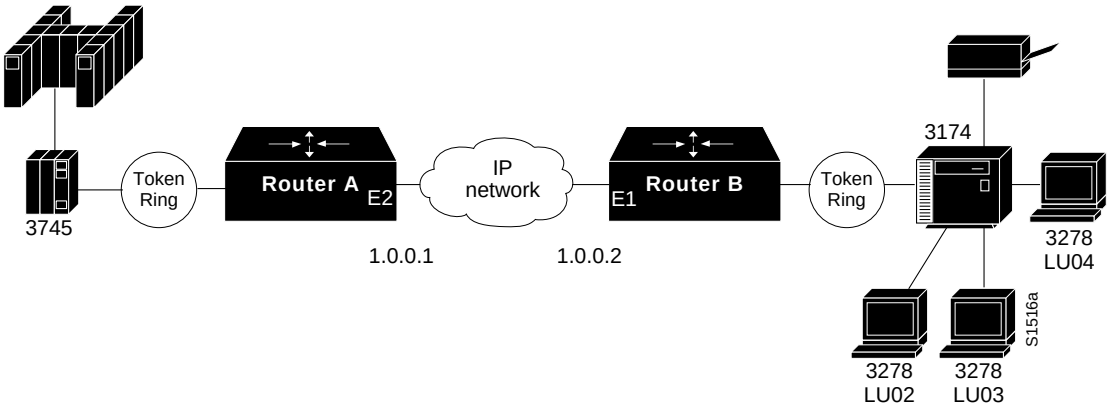
Priority output queuing is most useful on low-speed serial links. Figure 1-5 shows how priority queuing can be used to segregate traffic by priority level, speeding the transit of certain packets through the network.

Figure 1-5 Priority Output Queuing



You can also use intraprotocol traffic prioritization techniques to enhance internetwork performance. IP's type-of-service (TOS) feature and prioritization of IBM logical units (LUs) are intraprotocol prioritization techniques that can be implemented to improve traffic handling over routers. Figure 1-6 illustrates LU prioritization.

Figure 1-6 LU Prioritization Implementation



In Figure 1-6, the IBM mainframe is channel-attached to a 3745 communications controller, which is connected to a 3174 cluster controller via remote source-route bridging (RSRB). Multiple 3270 terminals and printers, each with a unique local LU address, are attached to the 3174. By applying LU address prioritization, you can assign a priority to each LU associated with a terminal or printer; that is, certain users can have terminals that have better response time than others, and printers can have lowest priority. This function increases application availability for those users running extremely important applications.

Finally, most routed protocols (such as AppleTalk, IPX, and DECnet) employ a cost-based routing protocol to assess the relative merit of the different routes to a destination. By tuning associated parameters, you can force particular kinds of traffic to take particular routes, thereby performing a type of manual traffic prioritization.

Load Balancing

The easiest way to add bandwidth in a backbone network is to implement additional links. Routers provide built-in load balancing for multiple links and paths. You can use up to four paths to a destination network. And, in some cases, the paths need not be of equal cost.

Within IP, routers provide load balancing on both a per-packet and a per-destination basis. For per-destination load balancing, each router uses its route cache to determine the output interface. If IGRP or Enhanced IGRP routing is used, unequal-cost load balancing is possible. The router uses metrics to determine which paths the packets will take; the amount of load balancing is user-adjustable.

Load balancing bridged traffic over serial lines is also supported. Serial lines can be assigned to circuit groups. If one of the serial links in the circuit group is in the spanning tree for a network, any of the serial links in the circuit group can be used for load balancing. Data ordering problems are avoided by assigning each destination to a serial link. Reassignment is done dynamically if interfaces go down or come up.

Alternate Paths

Many internetwork backbones carry mission-critical information. Organizations running such backbones are usually interested in protecting the integrity of this information at virtually any cost. Routers must offer sufficient reliability so that they are not the weak link in the internetwork chain. The key is to provide alternate paths that can come on line whenever link failures occur along active networks.

Consider what can go wrong in a simple internetwork. In Figure 1-7, the backbone network consists of the FDDI link between the two corporate buildings as well as serial links A, B, and C connecting the corporate site to the three remote sites. Secondary networks exist within both the corporate site and the remote sites, but they are not important to this analysis.

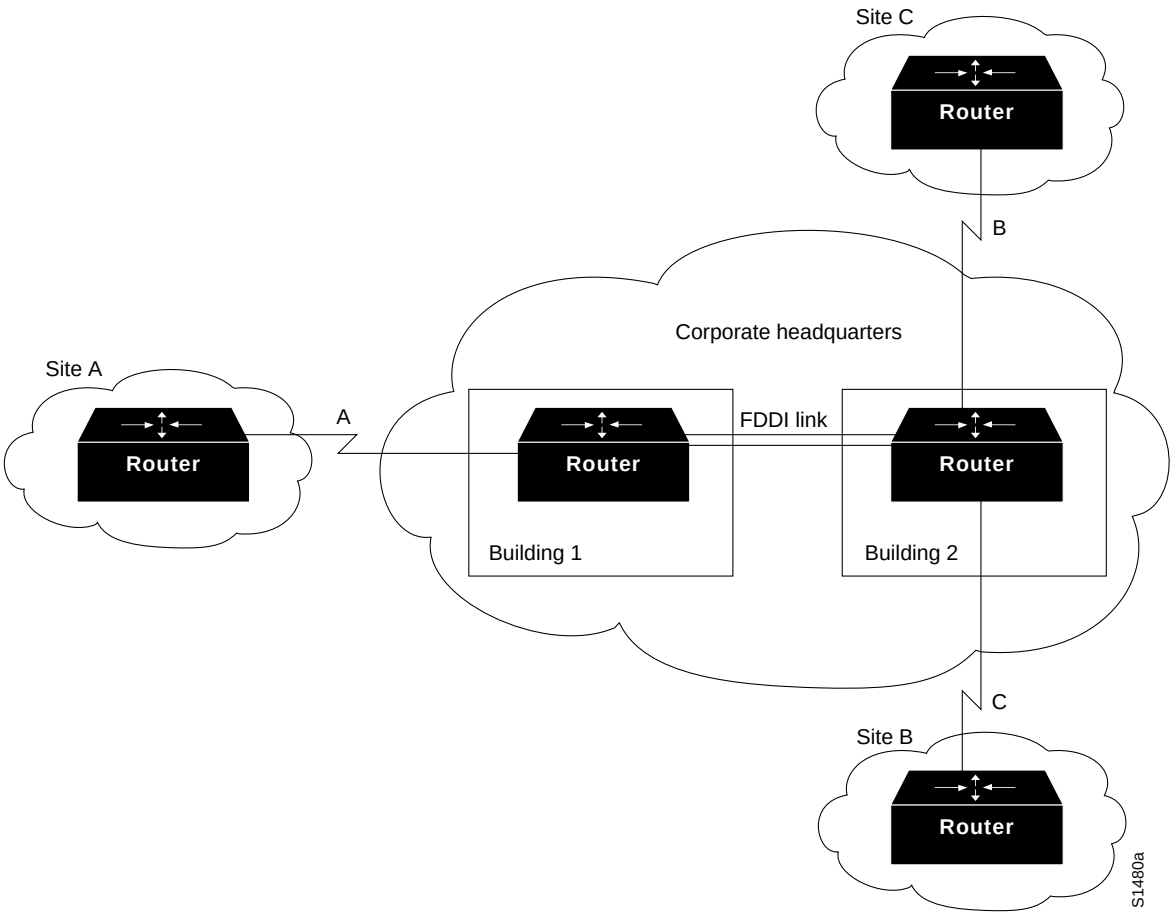
End-to-end reliability is not ensured simply by making the backbone fault tolerant. If communication on a local segment within any building is disrupted for any reason, that information will not reach the backbone. End-to-end reliability is only possible when redundancy is employed throughout the internetwork. Because this is usually cost prohibitive, most companies prefer to employ redundant paths only on those segments that carry mission-critical information.

What does it take to make the backbone reliable? Routers hold the key to reliable internetworking. Depending on the definition of reliability, this can mean duplicating every major system on each router and possibly every component. However, hardware component duplication is not the entire solution because extra circuitry is necessary to link the duplicate components to allow them to communicate. This solution is usually very expensive, but more importantly, it does not completely address the problem. Even assuming all routers in Figure 1-7 are completely reliable systems, link problems between nodes within a backbone can still defeat a redundant hardware solution.

To really address the problem of network reliability, *links* must be redundant. Further, it is not enough to simply duplicate all links. Dual links must terminate at multiple routers unless all backbone routers are completely fault tolerant (no single points of failure). Otherwise, backbone routers that are not fault tolerant become single points of failure. The inevitable conclusion is that a completely redundant router is not the most effective solution to the reliability problem, because it is expensive and still does not address link reliability.

Most network designers do not implement a completely redundant network. Instead, network designers implement partially redundant internetworks. The section “Choosing Internetworking Reliability Options,” presented later in this chapter, addresses several hypothetical networks representing commonly implemented points along the reliability continuum.

Figure 1-7 Simple Internetwork Illustrating Reliability Requirements

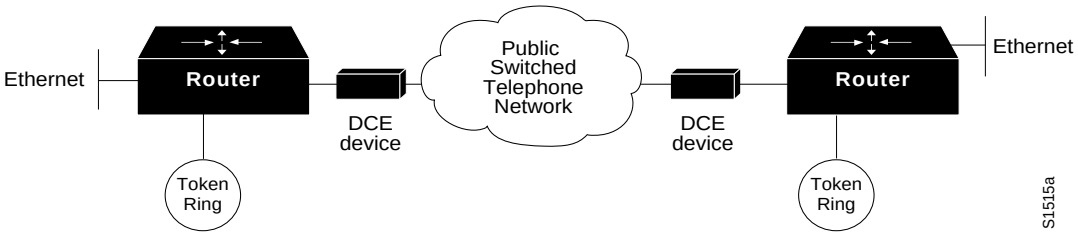


Switched Access

Switched access provides the ability to enable a WAN link on an as-needed basis via automated router controls. One model for a reliable backbone consists of dual, dedicated links and one switched link for idle hot backup. Under normal operational conditions, you can load balance over the dual links, but the switched link is not operational until one of the dedicated links fails.

Traditionally, WAN connections over the Public Switched Telephone Network (PSTN) have used dedicated lines. This can be very expensive when an application requires only low-volume, periodic connections. To reduce the need for dedicated circuits, a feature called dial-on-demand routing (DDR) is available. Using DDR, low-volume, periodic network connections can be made over the PSTN. A router activates the DDR feature when it receives a bridged or routed IP packet destined for a location on the other side of the dial-up line. After the router dials the destination phone number and establishes the connection, packets of any supported protocol can be transmitted. When the transmission is complete, the line is automatically disconnected. By terminating unneeded connections, DDR reduces cost of ownership. Figure 1-8 illustrates a DDR connection.

Figure 1-8 Dial-on-Demand Routing Environment



Encapsulation (Tunneling)

Encapsulation takes packets or frames from one network system and places them inside frames from another network system. This method is sometimes called *tunneling*. Tunneling provides a means for encapsulating packets inside a routable protocol via virtual interfaces. Synchronous Data Link Control (SDLC) transport is also an encapsulation of packets in a routable protocol. In addition, transport provides enhancements to tunneling, such as local data link layer termination, broadcast avoidance, media conversion, and other scalability optimizations.

Cisco routers support the following encapsulation and tunneling techniques:

- The IBM technology feature set provides these methods:
  - Serial tunneling (STUN) or Synchronous Data Link Control (SDLC) Transport
  - SRB with direct encapsulation
  - SRB with Fast Sequenced Transport (FST) encapsulation
  - SRB with Transmission Control Protocol/Internet Protocol (TCP/IP) encapsulation
- Generic Routing Encapsulation (GRE)

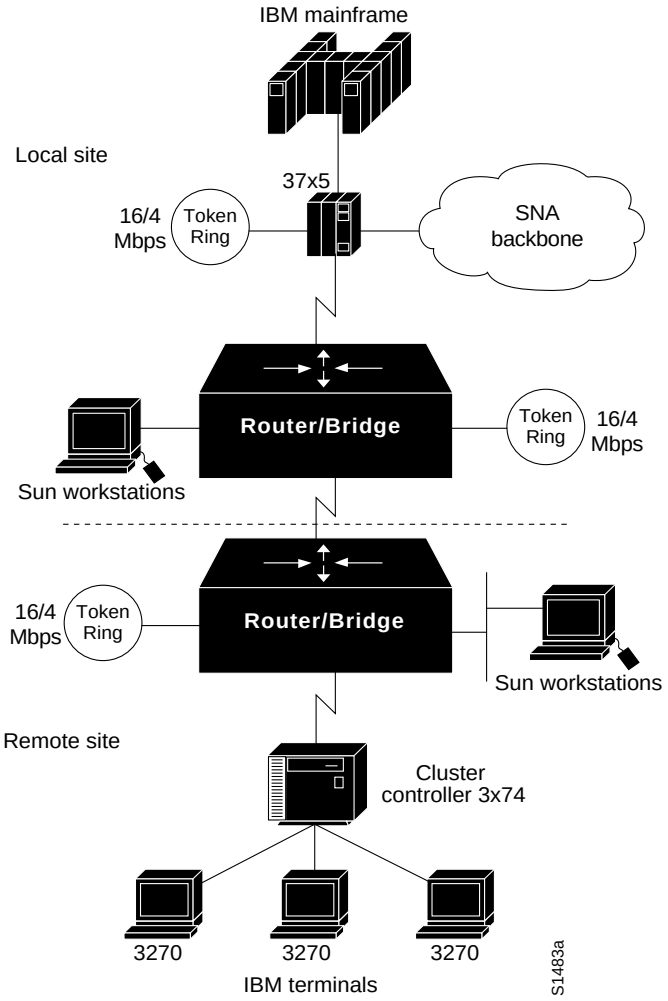
  Cisco supports encapsulating Novell Internetwork Packet Exchange (IPX), Internet Protocol (IP), Connectionless Network Protocol (CLNP), AppleTalk, DECnet Phase IV, Xerox Network Systems (XNS), Banyan Virtual Network System (VINES), and Apollo packets for transport over IP.
- Single-protocol tunneling techniques: Cayman (AppleTalk over IP), AURP (AppleTalk over IP), EON (CLNP over IP), and NOS (IP over IP).

The following discussion focuses on IBM encapsulations and the multiprotocol GRE tunneling feature.

IBM Features

STUN allows two devices that are normally connected by a direct serial link, using protocols compliant with SDLC or High-level Data Link Control (HDLC), to be connected through one or more routers. The routers can be connected via a multiprotocol network of arbitrary topology. STUN allows integration of System Network Architecture (SNA) networks and non-SNA networks using routers and existing network links. Transport across the multiprotocol network connecting the routers can use TCP/IP. This type of transport offers reliability and intelligent routing via any supported IP routing protocol. A STUN configuration is shown in Figure 1-9.

Figure 1-9 STUN Configuration



SDLC Transport is a variation of STUN that allows sessions using SDLC protocols and TCP/IP encapsulation to be locally terminated. SDLC Transport permits participation in SDLC windowing and retransmission activities.

When connecting remote devices that use SRB over a slow-speed serial link, most network designers choose RSRB with direct HDLC encapsulation. In this case, SRB frames are encapsulated in an HDLC-compliant header. This solution adds little overhead, preserving valuable serial link bandwidth. Direct HDLC encapsulation is not restricted to serial links (it can also be used over Ethernet, Token Ring, and FDDI links), but is most useful in situations where additional control overhead on the encapsulating network is not tolerable.

When more overhead can be tolerated, frame sequencing is important, but extremely reliable delivery is not needed, SRB packets can be sent over serial, Token Ring, Ethernet, and FDDI networks using FST encapsulation. FST is similar to TCP in that it provides packet sequencing. However, unlike TCP, FST does not provide packet-delivery acknowledgment.

For extremely reliable delivery in environments where moderate overhead can be tolerated, you can choose to encapsulate SRB frames in TCP/IP packets. This solution is not only reliable, it can also take advantage of routing features including handling via routing protocols, packet filtering, and multipath routing.

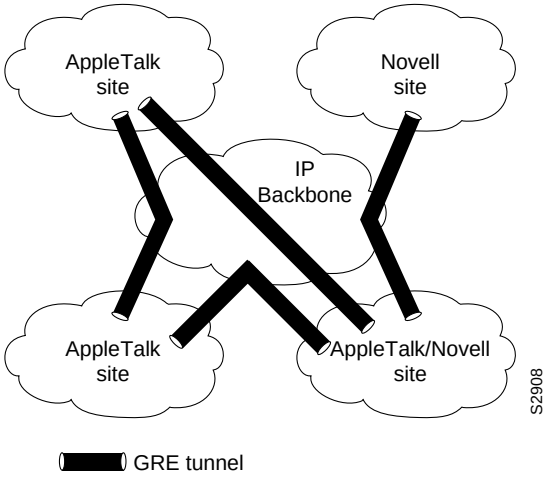
Generic Routing Encapsulation (GRE)

Cisco's Generic Routing Encapsulation (GRE) multiprotocol carrier protocol encapsulates IP, CLNP, IPX, AppleTalk, DECnet Phase IV, XNS, VINES, and Apollo packets inside IP tunnels. With GRE tunneling, a Cisco router at each site encapsulates protocol-specific packets in an IP header, creating a virtual point-to-point link to Cisco routers at other ends of an IP cloud, where the IP header is stripped off. By connecting multiprotocol subnetworks in a single-protocol backbone environment, IP tunneling allows network expansion across a single-protocol backbone environment. GRE tunneling involves three types of protocols:

- Passenger—protocol that is encapsulated (IP, CLNP, IPX, AppleTalk, DECnet Phase IV, XNS, VINES and Apollo)
- Carrier—GRE protocol provides carrier services
- Transport—IP carries the encapsulated protocol

GRE tunneling allows desktop protocols to take advantage of the enhanced route selection capabilities of IP. Many local-area network (LAN) protocols, including AppleTalk and Novell IPX, are optimized for local use. They have limited route selection metrics and hop count limitations. In contrast, IP routing protocols allow more flexible route selection and scale better over large internetworks. Figure 1-10 illustrates GRE tunneling across a single IP backbone between sites. Regardless of how many routers and paths may be associated with the IP cloud, the tunnel is seen as a single hop.

Figure 1-10 Using a Single Protocol Backbone

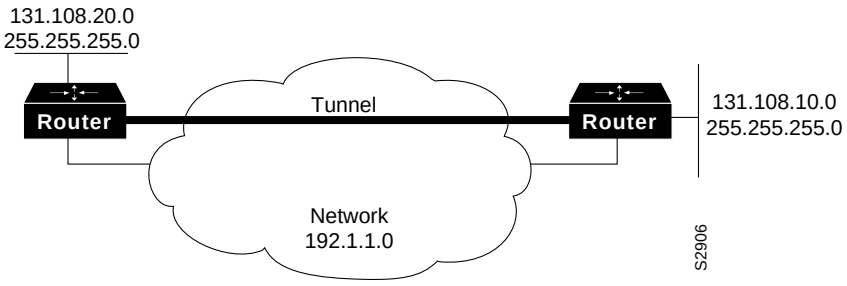


GRE provides key capabilities that other encapsulation protocols lack: sequencing and the ability to carry tunnelled data at high speeds. Some higher-level protocols require that packets are delivered in correct order. The GRE sequencing option provides this capability. GRE also has an optional key feature which allows you to avoid configuration errors by requiring the same key to be entered at each tunnel endpoint before the tunnelled data will be processed. IP tunneling also allows network

designers to implement policies, such as which types of traffic can use which routes or assignment of priority or security levels to particular traffic. Capabilities like these are lacking in many native LAN protocols.

IP tunneling provides communication between subnetworks that have invalid or discontiguous network addresses. With tunneling, virtual network addresses are assigned to subnetworks making discontiguous subnetworks reachable. Figure 1-11 illustrates that with GRE tunneling, it is possible for the two subnetworks of network 131.108.0.0 to talk to each other even though they are separated by another network.

Figure 1-11 Connecting Discontiguous Networks with Tunnels



Because encapsulation requires handling of the packets, it is generally faster to route protocols natively than to use tunnels. Tunneled traffic is switched at approximately half the typical process switching rates. This means approximately 1000 packets per second (pps) aggregate for each router. Tunneling is CPU intensive, and as such, should be turned on cautiously. Routing updates, SAP updates, and other administrative traffic may be sent over each tunnel interface. It is easy to saturate a physical link with routing information if several tunnels are configured over it. Performance will depend on the passenger protocol, broadcasts, routing updates, and bandwidth of the physical interfaces. It is also difficult to debug the physical link if problems occur. This problem can be mitigated in several ways. In the IPX environments, route filters and SAP filters cut down on the size of the updates that travel over tunnels. In AppleTalk networks, keeping zones small and using route filters can limit excess bandwidth requirements.

Tunneling can disguise the nature of a link, making it look slower, faster, or more or less costly than it may actually be in reality. This can cause unexpected or undesirable route selection. Routing protocols that make decisions based only on hop count will usually prefer a tunnel to a real interface. This may not always be the best routing decision because an IP cloud can comprise several different media with very disparate qualities; for example, traffic may be forwarded across both 100-Mbps Ethernet lines and 9.6- kbps serial lines. When using tunneling, pay attention to the media over which virtual tunnel traffic passes and the metrics used by each protocol.

If a network has sites that use protocol-based packet filters as part of a firewall security scheme, be aware that because tunnels encapsulate unchecked passenger protocols, you must establish filtering on the firewall router so that only authorized tunnels are allowed to pass. If tunnels are accepted from unsecured networks, it is a good idea to establish filtering at the tunnel destination or to place the tunnel destination outside the secure area of your network so that the current firewall scheme will remain secure.

When tunneling IP over IP, you must be careful to avoid inadvertently configuring a recursive routing loop. A routing loop occurs when the passenger protocol and the transport protocol are identical. The routing loop occurs because the best path to the tunnel destination is via the tunnel interface. A routing loop could occur, then, when tunneling IP over IP as follows:

1. The packet is placed in the output queue of the tunnel interface.
2. The tunnel interface includes a GRE header and enqueues the packet to the transport protocol (IP) for the destination address of the tunnel interface.
3. IP looks up the route to the tunnel destination address and learns that the path is the tunnel interface.
4. Once again, the packet is placed in the output queue of the tunnel interface as described in step 1; hence, the routing loop.

When a router detects a recursive routing loop, it shuts down the tunnel interface for 1 to 2 minutes and issues a warning message before it goes into the recursive loop. Another indication that a recursive route loop has been detected is if the tunnel interface is up, and the line protocol is down.

To avoid recursive loops, keep passenger and transport routing information in separate locations by implementing the following procedures:

- Use separate routing protocol identifiers (for example, igrp 1 and igrp 2).
- Use different routing protocols.
- Assign the tunnel interface a very low bandwidth so that routing protocols, such as IGRP, will recognize a very high metric for the tunnel interface and will, therefore, choose the correct next hop (that is, choose the best physical interface instead of the tunnel).
- Keep the two IP address ranges distinct; that is, use a major address for your tunnel network that is different from your actual IP network. Keeping the address ranges distinct also aids in debugging because it is easy to identify an address as the tunnel network instead of the physical network and vice versa.

Evaluating Distribution Services

This section addresses internetworking features that support distribution services. The following topics are discussed:

- Area and service filtering
- Policy-based distribution
- Gateway service
- Interprotocol route redistribution
- Media translation

Area and Service Filtering

Traffic filters based on *area* or *service* type are the primary distribution service tools used to provide policy-based access control into backbone services. Both area and service filtering are implemented using *access lists*. An access list is a sequence of statements, each of which either permits or denies certain conditions or addresses. Access lists can be used to permit or deny messages from particular network nodes and messages sent using particular protocols and/or services.

Area, or network, access filters are used to enforce the selective transmission of traffic based on network address. You can apply these on incoming or outgoing ports. Service filters use access lists applied to protocols (such as IP's UDP), applications such as the Simple Mail Transfer Protocol (SMTP), and specific protocols.

Suppose you have a network connected to the Internet, and you want any host on an Ethernet to be able to form TCP connections to any host on the Internet. However, you do not want Internet hosts to be able to form TCP connections to hosts on the Ethernet except to the SMTP port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same two port numbers are used throughout the life of the connection. Mail packets coming in from the Internet will have a destination port of 25. Outbound packets will have the port numbers reversed. The fact that the secure system behind the router always will be accepting mail connections on port 25 is what makes it possible to separately control incoming and outgoing services.The access list can be configured on either the outbound or inbound interface.

In the following example, the Ethernet network is a Class B network with the address 128.88.0.0, and the mail host's address is 128.88.1.2. The keyword **established** is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which indicate that the packet belongs to an existing connection.

```
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.0.0 0.0.255.255 established
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.1.2 0.0.0.0 eq 25
interface ethernet 0
ip access-group 102
```

Policy-Based Distribution

Policy-based distribution is based on the premise that different departments within a common organization might have different policies regarding traffic dispersion through the organization-wide internetwork. Policy-based distribution aims to meet the differing requirements without compromising performance and information integrity.

A *policy* within this internetworking context can be defined as a rule or set of rules that govern end-to-end distribution of traffic to (and subsequently through) a backbone network. One department might send traffic representing three different protocols to the backbone, but might wish to expedite one particular protocol's transit through the backbone because it carries mission-critical application information. To minimize already excessive internal traffic, another department might want to exclude all backbone traffic except electronic mail and one key custom application from entering its network segment.

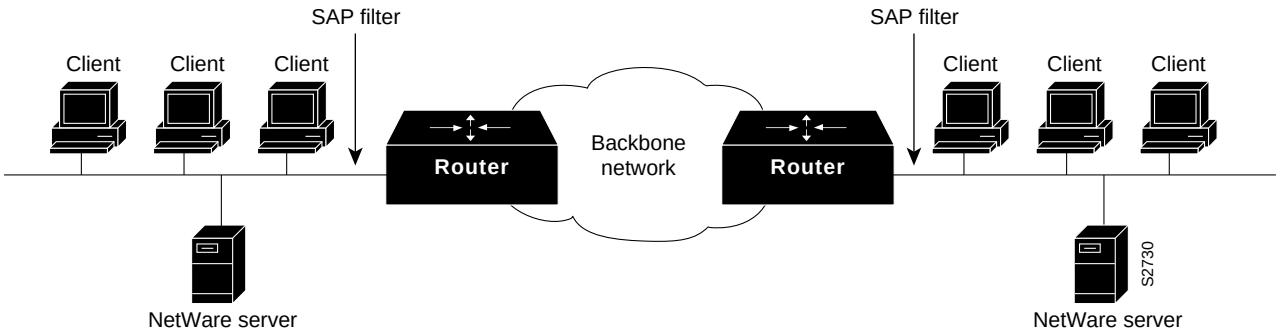
These examples reflect policies specific to a single department. However, policies can reflect overall organizational goals. For example, an organization might wish to regulate backbone traffic to a maximum of 10 percent average bandwidth during the work day and one-minute peaks of 30 percent utilization. Another corporate policy might be to ensure that communication between two remote departments can freely occur, despite any differences in technology.

Different policies frequently require different work-group and department technologies. Therefore, support for policy-based distribution implies support for the wide range of technologies currently used to implement these policies. This in turn allows you to implement solutions that support a wide range of policies, which helps to increase organizational flexibility and application availability.

In addition to support for internetworking technologies, there must be a means both to keep separate and integrate these technologies, as appropriate. The different technologies should be able to coexist peacefully or combine intelligently, as the situation warrants.

Consider the situation depicted in Figure 1-12. Assume a corporate policy limits unnecessary backbone traffic. One way to do this is to restrict the transmission of Service Advertisement Protocol (SAP) messages. SAP messages allow NetWare servers to advertise services to clients. The organization might have another policy stating that all NetWare services should be provided locally. If this is the case, there should be no reason for services to be advertised remotely. SAP filters prevent SAP traffic from leaving a router interface, thereby fulfilling this policy.

Figure 1-12 Policy-Based Distribution: SAP Filtering



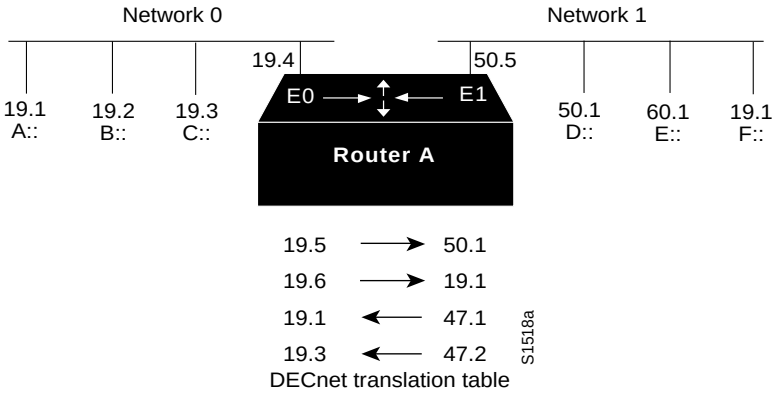
Gateway Service

Protocol gateway capabilities are part of each router's standard software. For example, DECnet is currently in Phase V. DECnet Phase V addresses are different than DECnet Phase IV addresses. For those networks where both types of hosts must coexist, two-way Phase IV/Phase V translation conforms to Digital-specified guidelines. The routers interoperate with Digital routers, and Digital hosts do not differentiate between the different devices.

The connection of multiple independent DECnet networks can lead to addressing problems. Nothing precludes two independent DECnet administrators from assigning node address 10 to one of the nodes in their respective networks. When the two networks are connected at some later time, conflicts result. DECnet *address translation gateways* (ATGs) address this problem.

The ATG solution provides router-based translation between addresses in two different DECnet networks connected by a router. Figure 1-13 illustrates an example of this operation.

Figure 1-13 Example DECnet ATG Implementation



In Network 0, the router is configured at address 19.4 and is a Level 1 router. In Network 1, the router is configured at address 50.5 and is an area router. At this point, no routing information is exchanged between the two networks. The router maintains a separate routing table for each network. By establishing a translation map, packets in Network 0 sent to address 19.5 will be routed to Network 1, and the destination address will be translated to 50.1. Similarly, packets sent to address 19.6 in Network 0 will be routed to Network 1 as 19.1; packets sent to address 47.1 in Network 1 will be routed to Network 0 as 19.1; and packets sent to 47.2 in Network 1 will be sent to Network 0 as 19.3.

AppleTalk is another protocol with multiple revisions, each with somewhat different addressing characteristics. AppleTalk Phase 1 addresses are simple local forms; AppleTalk Phase 2 uses extended (multinetwork) addressing. Normally, information sent from a Phase 2 node cannot be understood by a Phase 1 node if Phase 2 extended addressing is used. Routers support routing between Phase 1 and Phase 2 nodes on the same cable by using transitional routing.

You can accomplish transitional routing by attaching two router ports to the same physical cable. Configure one port to support nonextended AppleTalk and the other to support extended AppleTalk. Both ports must have unique network numbers. Packets are translated and sent out the other port as necessary.

Interprotocol Route Redistribution

The preceding section, "Gateway Service," discussed how *routed* protocol gateways (such as one that translates between AppleTalk Phase 1 and Phase 2) can allow two end nodes with different implementations to communicate. Routers can also act as gateways for *routing* protocols. Information derived from one routing protocol such as the IGRP can be passed to and used by another routing protocol such as RIP. This is useful when running multiple routing protocols in the same internetwork.

Routing information can be exchanged between any supported IP routing protocols. These include RIP, IGRP, OSPF, HELLO, EGP, and BGP. Similarly, route redistribution is supported by ISO CLNS for route redistribution between ISO IGRP and IS-IS. Static route information can also be redistributed. Defaults can be assigned so that one routing protocol can use the same metric for all redistributed routes, thereby simplifying the routing redistribution mechanism.

Media Translation

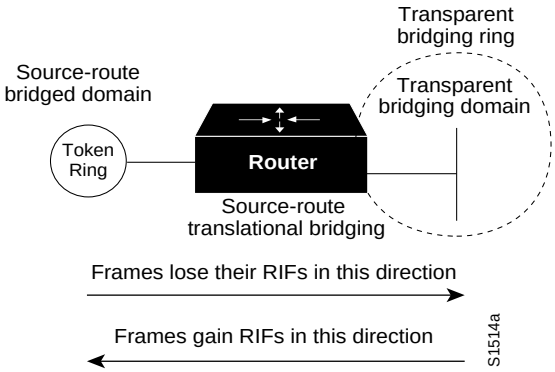
Media translation techniques translate frames from one network system into frames of another. Such translations are rarely 100 percent effective because one system might have attributes with no corollary in the other. For example, Token Ring networks support a built-in priority and reservation system while Ethernet networks do not. Translations between Token Ring and Ethernet networks must somehow account for this discrepancy. It is possible for two vendors to make different decisions about how this discrepancy will be handled, which can prevent multivendor interoperation.

For those situations where communication between end stations on different media is required, routers can translate between Ethernet and Token Ring frames. For direct bridging between Ethernet and Token Ring environments, use either source-route translational bridging or source-route transparent bridging (SRT). Source-route translational bridging translates between Token Ring and Ethernet frame formats; SRT allows routers to use both SRB and the transparent bridging algorithm used in standard Ethernet bridging.

When bridging from the SRB domain to the transparent bridging domain, the SRB fields of the frames are removed. RIFs are cached for use by subsequent return traffic. When bridging in the opposite direction, the router checks the packet to see if it has a multicast or broadcast destination or a unicast destination. If it has a multicast or broadcast destination, the packet is sent as a

spanning-tree explorer. If it has a unicast destination, the router looks up the path to the destination in the RIF cache. If a path is found, it will be used; otherwise, the router will send the packet as a spanning-tree explorer. A simple example of this topology is shown in Figure 1-14.

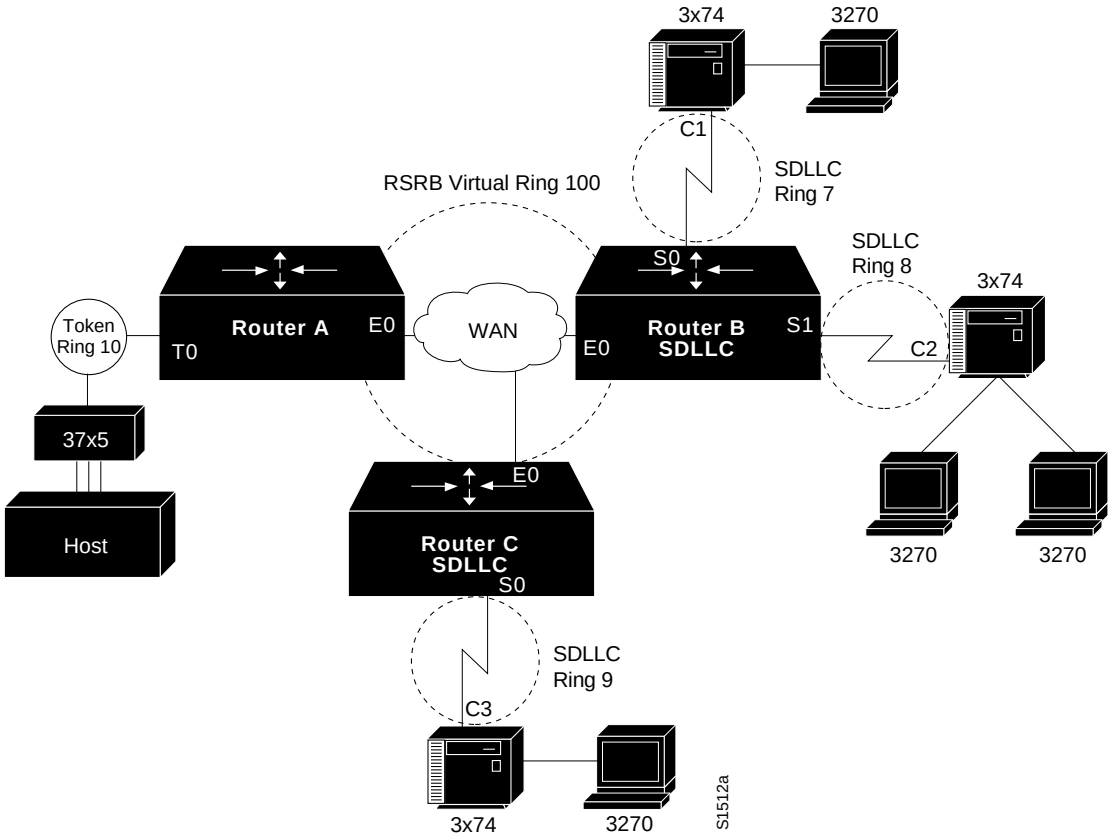
Figure 1-14 Source-Route Translational Bridging Topology



Routers support SRT through implementation of both transparent bridging and SRB algorithms on each SRT interface. If an interface notes the presence of a RIF field, it uses the SRB algorithm; if not, it uses the transparent bridging algorithm.

Translation between serial links running the SDLC protocol and Token Rings running LLC2 is also available. This is referred to as SDLLC frame translation. SDLLC frame translation allows connections between serial lines and Token Rings. This is useful for consolidating traditionally disparate SNA/SDLC networks into a LAN-based, multiprotocol, multimedia backbone network. Using SDLLC, routers terminate SDLC sessions, translate SDLC frames to LLC2 frames, and then forward the LLC2 frames using RSRB over a point-to-point or IP network. Since a router-based IP network can use arbitrary media such as FDDI, Frame Relay, X.25, or leased lines, routers support SDLLC over all such media through IP encapsulation. A complex SDLLC configuration is shown in Figure 1-15.

Figure 1-15 Complex SDLLC Configuration



Evaluating Local-Access Services

The following discussion addresses internetworking features that support local-access services. Local-access service topics outlined here include:

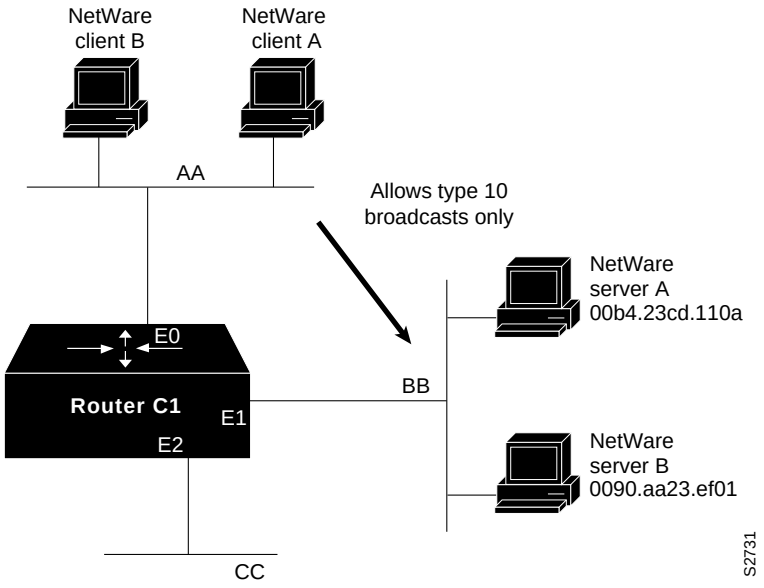
- Value-added network addressing
- Network segmentation
- Broadcast and multicast capabilities
- Naming, proxy, and local cache capabilities
- Media access security
- Router discovery

Value-Added Network Addressing

Address schemes for LAN-based networks such as NetWare and others do not always adapt perfectly to use over multisegment LANs or WANs. One tool routers implement to ensure operation of such protocols is protocol-specific *helper addressing*. Helper addressing is a mechanism to assist the movement of specific traffic through a network when that traffic might not otherwise transit the network.

The use of *helper addressing* is best illustrated with an example. Consider the use of helper addresses in Novell IPX internetworks. Novell clients send broadcast messages when looking for a server. If the server is not local, broadcast traffic must be sent through routers. Helper addresses and access lists can be used together to allow broadcasts from certain nodes on one network to be directed specifically to certain servers on another network. Multiple helper addresses on each interface are supported, so broadcast packets can be forwarded to multiple hosts. Figure 1-16 illustrates the use of NetWare-based helper addressing.

Figure 1-16 Example Network Map Illustrating Helper Address Broadcast Control



NetWare clients on Network AA are allowed to broadcast to any server on Network BB. An applicable access list would specify that broadcasts of type 10 will be permitted from all nodes on Network AA. A configuration-specified helper address identifies the addresses on Network BB to which these broadcasts are directed. No other nodes on Network BB receive the broadcasts. No other broadcasts other than type 10 broadcasts are routed.

Any downstream networks beyond Network AA (for example, some Network AA1) are not allowed to broadcast to Network BB through Router C1, unless the routers partitioning Networks AA and AA1 are configured to forward broadcasts with a series of configuration entries. These entries must be applied to the input interfaces and be set to forward broadcasts between directly connected networks. In this way, traffic is passed along in a directed manner from network to network.

Network Segmentation

The splitting of networks into more manageable pieces is an essential role played by local-access routers. In particular, local-access routers implement local policies and limit unnecessary traffic. Examples of capabilities that allow network designers to use local-access routers to segment networks include IP subnets, DECnet area addressing, and AppleTalk zones.

You can use local-access routers to implement local policies by placing the routers in strategic locations and by configuring specific segmenting policies. For example, you can set up a series of LAN segments with different subnet addresses; routers would be configured with suitable interface addresses and subnet masks. In general, traffic on a given segment is limited to local broadcasts,

traffic intended for a specific end station on that segment, or traffic intended for another specific router. By distributing hosts and clients carefully, you can use this simple method of dividing up a network to reduce overall network congestion.

Broadcast and Multicast Capabilities

Many protocols use *broadcast* and *multicast* capabilities. Broadcasts are messages that are sent out to all network destinations. Multicasts are messages sent to a specific subset of network destinations. Routers inherently reduce broadcast proliferation by default. However, routers can be configured to relay broadcast traffic if necessary. Under certain circumstances, passing along broadcast information is desirable and possibly necessary. The key is that the handling of broadcasts and multicasts can be controlled using routers.

In the IP world, as with many other technologies, broadcast requests are very common. Unless broadcasts are controlled, network bandwidth can be seriously reduced. Routers offer various broadcast-limiting functions that reduce network traffic and minimize broadcast storms. For example, directed broadcasting allows for broadcasts to a specific network or a series of networks, rather than to the entire internetwork. When flooded broadcasts (broadcasts sent through the entire internetwork) are necessary, Cisco routers support a technique by which these broadcasts are sent over a spanning tree of the network. The spanning tree ensures complete coverage without excessive traffic because only one packet is sent over each network segment.

As discussed previously in "Value-Added Network Addressing," broadcast assistance is accommodated with the *helper address* mechanisms. Using helper addresses, you can allow a router or series or routers to relay broadcasts that would otherwise be blocked. For example, you can permit retransmission of SAP broadcasts using helper addresses, thereby notifying clients on different network segments of certain NetWare services available from specific remote servers.

Note For a case study on User Datagram Protocol (UDP) broadcast control, see Topic 4, "UDP Broadcast Flooding" in the Cisco publication *Internetworking Applications: Case Studies*, Vol. 1 No. 2.

The Cisco IP multicast feature allows IP traffic to be propagated from one source to any number of destinations. Rather than sending one packet to each destination, one packet is sent to a multicast group identified by a single IP destination group address. IP multicast provides excellent support for such applications as video and audio conferencing, resource discovery, and stock market traffic distribution.

For full support of IP multicast, IP hosts must run the Internet Group Management Protocol (IGMP). IGMP is used by IP hosts to report their multicast group memberships to an immediately neighboring multicast router. The membership of a multicast group is dynamic. Multicast routers send IGMP query messages on their attached local networks. Host members of a multicast group respond to a query by sending IGMP reports for multicast groups to which they belong. Reports sent by the first host in a multicast group suppress the sending of identical reports from other hosts of the same group.

The multicast router attached to the local network takes responsibility for forwarding multicast datagrams from one multicast group to all other networks that have members in the group. Routers build multicast group distribution trees (routing tables) so that multicast packets have loop-free paths to all multicast group members and so that multicast packets are not duplicated. If no reports are received from a multicast group after a set number of IGMP queries, the multicast routers assume the group has no local members and stop forwarding multicasts intended for that group.

Cisco routers support Protocol Independent Multicast (PIM). PIM allows network administrators to add IP multicast routing to their existing IP network regardless of what unicast routing protocol they are using. PIM has two modes: *dense* and *sparse*. In dense mode, receivers are densely populated and the assumption is that networks will probably use the forwarded data. In sparse mode, receivers are widely distributed and the assumption is that the network segment will not use the information.

The Cisco router dynamically discovers Distance Vector Multicast Routing Protocol (DVMRP) neighbors on the interfaces configured for PIM. Once these neighbors are discovered, the Cisco router treats the interface as if there were a PIM neighbor present (with respect to flooding). The Cisco router sends DVMRP Report messages advertising routes so that sources in the PIM cloud are known to the DVMRP routers. IGMP Reports are sent for all groups known in the PIM cloud so that the DVMRP routers know there are group members present.

The Cisco router can also communicate with DVMRP systems using tunnels. You must configure the IP addresses of the end-points of the tunnel and indicate that the tunnel operates in DVMRP mode. When DMVRP tunnels are configured, the Cisco router caches DVMRP Reports received. This is done so that Reverse Path Forwarding (RPF) checks are relative to sources reached over the tunnel (without having to send unicast packets over the tunnel). DVMRP tunnels are useful to connect PIM clouds to the MBONE (the Internet Multicast backbone). DVMRP can also be used to connect PIM clouds and MOSPF (Multicast OSPF) clouds together.

Naming, Proxy, and Local Cache Capabilities

Three key router capabilities help reduce network traffic and promote efficient internetworking operation: name service support, proxy services, and local caching of network information.

Network applications and connection services provided over segmented internetworks require a rational way to resolve names to addresses. Various facilities accommodate this requirement. Any router you select must support the name services implemented for different end system environments. Examples of supported name services include NetBIOS, IP's Domain Name System (DNS) and IEN-116, and AppleTalk Name Binding Protocol (NBP).

A router can also act as a *proxy* for a name server. The router's support of NetBIOS name caching is one example of this kind of capability. NetBIOS name caching allows the router to maintain a cache of NetBIOS names, which avoids the overhead of transmitting all of the broadcasts between client and server NetBIOS PCs (IBM PCs or PS/2s) in an SRB environment.

When NetBIOS name caching is enabled, the router does the following:

- Notices when any host sends a series of duplicate query frames and limits retransmission to one frame per period. The time period is a configuration parameter.
- Keeps a cache of mappings between NetBIOS server and client names and their MAC addresses. As a result, broadcast requests sent by clients to find servers (and by servers in reply to their clients) can be sent directly to their destinations, rather than being broadcast across the entire bridged network.

When NetBIOS name caching is enabled and default parameters are set on the router, the NetBIOS name server, and the NetBIOS name client, approximately 20 broadcast packets per login are kept on the local ring where they are generated.

In most cases, the NetBIOS name cache is best used in situations where large amounts of NetBIOS broadcast traffic might create bottlenecks on a WAN that connects local internetworks to distant locations.

The router is capable of saving bandwidth (or handling nonconforming name resolution protocols) using a variety of other proxy facilities as well. By using routers to act on behalf of other devices to perform various functions, you can more easily scale networks. Instead of being forced to add

bandwidth when a new workgroup is added to a location, you can use a router to manage address resolution and control message services. Examples of this kind of capability include the proxy explorer feature of SRB and the proxy polling feature of STUN implementations.

Sometimes there are situations in which portions of networks are unable to participate in routing activity or do not implement software conforming to generally implemented address-resolution protocols. Proxy implementations on routers allow network designers to support these networks or hosts without reconfiguring an internetwork. Examples of these kinds of capabilities include proxy ARP address resolution for IP internetworks and NBP proxy in AppleTalk internetworks.

Local caches store previously learned information about the network so that new information requests need not be issued each time the same piece of information is desired. A router's ARP cache stores physical address/network address mappings so that it need not broadcast ARP requests more than once within a given time period for the same address. Address caches are maintained for many other protocols as well, including DECnet, Novell's IPX, and SRB, where RIF information is cached.

Media Access Security

If all corporate information is readily available to all employees, security violations and inappropriate file access can occur. To prevent this, routers must do the following:

- Keep local traffic from inappropriately reaching the backbone
- Keep backbone traffic from exiting the backbone into an inappropriate department or workgroup network

These two functions require packet filtering. Packet filtering capabilities should be tailored to support a variety of corporate policies. Packet filtering methods can reduce traffic levels on a network, thereby allowing a company to continue using its current technology rather than investing in more network hardware. In addition, packet filters can improve security by keeping unauthorized users from accessing information and can minimize network problems caused by excessive congestion.

Routers support a number of filtering schemes designed to provide control over network traffic reaching the backbone. Perhaps the most powerful of these filtering mechanisms is the access list. Each of the following possible local-access services can be provided through access lists:

- You have an Ethernet-to-Internet routing network and you want any host on the Ethernet to be able to form TCP connections to any host on the Internet. However, you do not want Internet hosts to be able to form TCP connections into the Ethernet except to the SMTP port of a dedicated mail host.
- You want to advertise only one network through a RIP routing process.
- You want to prevent packets that originated on any Sun workstation from being bridged on a particular Ethernet segment.
- You want to keep a particular protocol based on Novell IPX from establishing a connection between a source network/source port combination and a destination network/destination port combination.

Access lists physically prevent certain packets from traversing a particular router interface, thereby providing a general tool for implementing network security. In addition to this method, several specific security systems already exist to help increase network security. For example, the U.S. Government has specified the use of an optional field within the IP packet header to implement a hierarchical packet security system called the Internet Protocol Security Option (IPSO).

IPSO support on routers addresses both the basic and extended security options described in a draft of the IPSO circulated by the Defense Communications Agency. This draft document is an early version of RFC 1108. IPSO defines security levels (for example, TOP SECRET, SECRET, and others) on a per-interface basis and accepts or rejects messages based on whether they include adequate authorization.

Some security systems are designed to keep remote users from accessing the network unless they have adequate authorization. For example, the Terminal Access Controller Access Control System (TACACS) is a means of protecting modem access into a network. The Defense Data Network (DDN) developed TACACS to control access to its TAC terminal servers.

The router's TACACS support is patterned after the DDN application. When a user attempts to start an EXEC command interpreter on a password-protected line, TACACS prompts for a password. If the user fails to enter the correct password, access is denied. Router administrators can control various TACACS parameters, such as the number of retries allowed, the timeout interval, and the enabling of TACACS accounting.

The Challenge Handshake Authentication Protocol (CHAP) is another way to keep unauthorized remote users from accessing a network. It is also commonly used to control router-to-router communications. When CHAP is enabled, a remote device (for example, a PC, workstation, router, or communication server) attempting to connect to a local router is "challenged" to provide an appropriate response. If the correct response is not provided, network access is denied.

CHAP is becoming popular because it does not require a secret password to be sent over the network. CHAP is supported on all router serial lines using Point-to-Point Protocol (PPP) encapsulation.

Router Discovery

Hosts must be able to locate routers when they need access to devices external to the local network. When more than one router is attached to a host's local segment, the host must not only be able to locate a router, it must be able to locate the router representing an optimal path to the destination. This process of finding routers is called *router discovery.*

The following are router discovery protocols:

- End System-to-Intermediate System (ES-IS)—This protocol is defined by the ISO OSI protocol suite. It is dedicated to the exchange of information between intermediate systems (routers) and end systems (hosts). ESs send "ES hello" messages to all ISs on the local subnetwork. In turn, "IS hello" messages are sent from all ISs to all ESs on the local subnetwork. Both types of messages convey the subnetwork and network-layer addresses of the systems that generate them. Using this protocol, end systems and intermediate systems can locate one another.
- ICMP Router Discovery Protocol (IRDP)—Although the issue is currently under study, there is currently no single standardized manner for end stations to locate routers in the IP world. In many cases, stations are simply configured manually with the address of a local router. However, Request For Comments (RFC) 1256 outlines a router discovery protocol using the Internet Control Message Protocol (ICMP). This protocol is commonly referred to as IRDP.
- Proxy Address Resolution Protocol (ARP)—ARP uses broadcast messages to determine the MAC-layer address that corresponds to a particular internetwork address. ARP is sufficiently generic to allow use of IP with virtually any type of underlying media-access mechanism. A router that has proxy ARP enabled responds to ARP requests for those hosts for which it has a route, which allows hosts to assume that all other hosts are actually on their network.
- RIP—RIP is a routing protocol that is commonly available on IP hosts. Many hosts use RIP to find the address of the routers on a LAN or, when there are multiple routers, to pick the best router to use for a given internetwork address.

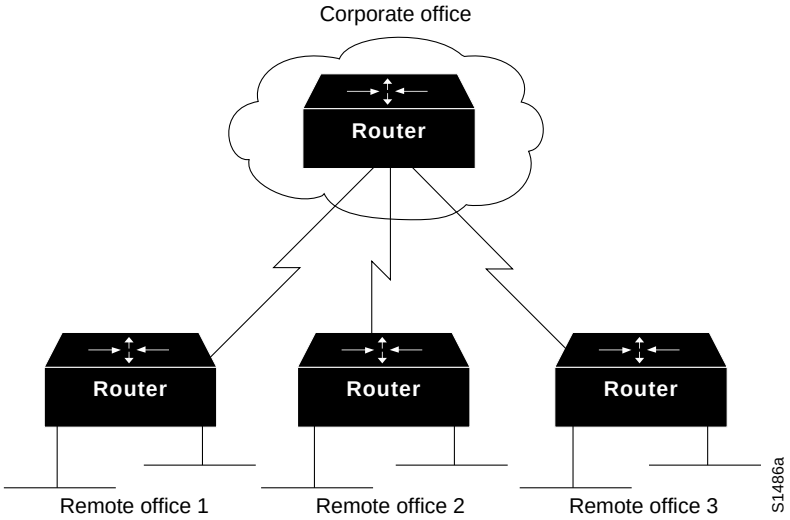
Cisco routers support the router discovery protocols listed above. You can choose the router discovery mechanism that works best in your particular environment.

Choosing Internetworking Reliability Options

The first concern of most network designers is to determine the required level of application availability. In general, this key consideration is balanced against implementation cost. For most organizations, the cost of making a network completely fault tolerant is prohibitive. Determining the appropriate level of fault tolerance to be included in a network, and where redundancy should be used is not trivial.

The nonredundant internetwork design in Figure 1-17 illustrates the considerations involved with increasing levels of internetwork fault tolerance.

Figure 1-17 Typical Nonredundant Internetwork Design



The internetwork shown in Figure 1-17 has two levels of hierarchy: a corporate office and remote offices. Assume the corporate office has 8 Ethernet segments, to which approximately 400 users (an average of 50 per segment) are connected. Each Ethernet segment is connected to a router. In the remote offices, two Ethernet segments are connected to the corporate office through a router. The router in each remote office is connected to the router in the corporate office through a T1 link.

The following sections address various approaches to creating redundant internetworks, provides some context for each approach, and contrasts their relative merits and drawbacks. The following four sections are provided:

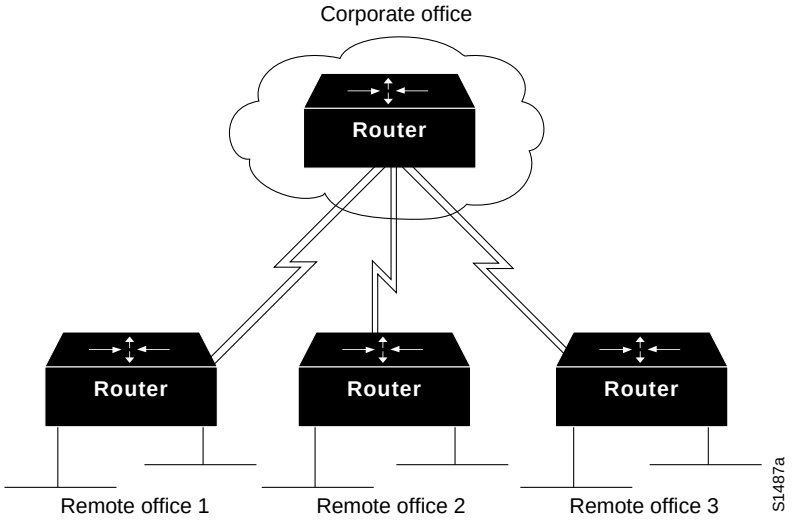
- Redundant links versus meshed topologies
- Redundant power systems
- Fault-tolerant media implementations
- Backup router hardware

Redundant Links versus Meshed Topologies

Typically, WAN links are the least reliable components in an internetwork, usually because of problems in the local loop. In addition to being relatively unreliable, these links are often an order of magnitude slower than the LANs they connect. However, because they are capable of connecting geographically diverse sites, WAN links often make up the backbone network, and are therefore critical to corporate operations. The combination of potentially suspect reliability, lack of speed, and high importance makes the WAN link a good candidate for redundancy.

As a first step in making the example internetwork more fault tolerant, you might add a WAN link between each remote office and the corporate office. This results in the topology shown in Figure 1-18. The new topology has several advantages. First, it provides a backup link that can be used if a primary link connecting any remote office and the corporate office fails. Second, if the routers support load balancing, link bandwidth has now been increased, lowering response times for users and increasing application availability.

Figure 1-18 Internetwork with Dual Links to Remote Offices

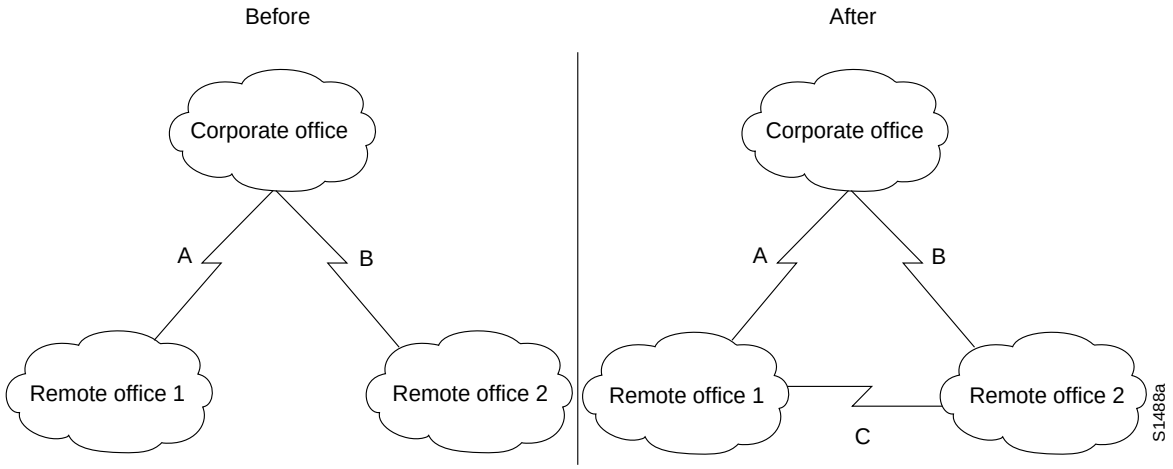


Load balancing in transparent bridging and IGRP environments is another tool for increasing fault tolerance. Routers also support load balancing on either a per-packet or a per-destination basis in all IP environments. Per-packet load balancing is recommended if the WAN links are relatively slow (for example, less than 56 kbps). If WAN links are faster than 56 kbps, enabling fast switching is recommended. When fast switching is enabled, load balancing occurs on a per-destination basis.

Routers can automatically compensate for failed WAN links through routing algorithms of protocols such as IGRP, OSPF, and IS-IS. If one link fails, the routing software recalculates the routing algorithm and begins sending all traffic through another link. This allows applications to proceed in the face of WAN link failure, improving application availability.

The primary disadvantage of duplicating WAN links to each remote office is cost. In the example outlined in Figure 1-18, three new WAN links are required. In large star networks with more remote offices, 10 or 20 new WAN links might be needed, as well as new equipment (including new WAN router interfaces). A lower cost alternative that is becoming increasingly popular links the remote offices using a meshed topology. This topology is illustrated in Figure 1-19.

Figure 1-19 Evolution from a Star to a Meshed Topology



In the "before" portion of Figure 1-19, any failure associated with either Link A or B blocks access to a remote site. The failure might involve the link connection equipment, such as a data service unit (DSU) or a channel service unit (CSU), the router (either the entire router or a single router port), or the link itself. Adding Link C as shown in the "after" portion of the figure, offsets the effect of a failure in any single link. If Link A or B fails, the affected remote site can still access the corporate office through Link C and the other site's link to the corporate office. Note also that if Link C fails, the two remote sites can communicate through their connections to the corporate office.

A meshed topology has three distinct advantages over a redundant star topology:

- A meshed topology is usually slightly less expensive (at least by the cost of one WAN link).
- A meshed topology provides more direct (and, potentially, faster) communication between remote sites, which translates to greater application availability. This can be useful if direct traffic volumes between remote sites are relatively high.
- A meshed topology promotes distributed operation, preventing bottlenecks on the corporate router and further increasing application availability.

A redundant star is a reasonable solution under the following conditions:

- Relatively little traffic must travel between remote offices.
- Traffic moving between corporate and remote offices is delay sensitive and mission critical. The delay and potential reliability problems associated with making an extra hop when a link between a remote office and the corporate office fails might not be tolerable.

Redundant Power Systems

Power faults are common in large-scale networks. Because they can strike across a very local or a very wide scale, power faults are difficult to preempt. Simple power problems include dislodged power cords, tripped circuit breakers, and local power supply failures. More extensive power problems include large-scale outages due to natural phenomena (such as lightning strikes) or brown-outs. Each organization must assess its needs and the probability of each type of power outage before determining which preventative actions to take.

You can take many precautions to try to ensure that problems such as dislodged power cords do not occur frequently. These fall outside the scope of this publication and will not be discussed here. This publication focuses on issues addressable by internetworking devices.

From the standpoint of internetworking devices, dual power systems can prevent otherwise debilitating failures. Imagine a situation where the so-called "backbone-in-a-box" configuration is being used. This configuration calls for the connection of many networks to a router being used as a "connectivity hub." Benefits include a high-speed backbone (essentially the router's backplane) and cost efficiency (less media). Unfortunately, if the router's power system becomes faulty, each network connected to that router loses its ability to communicate with all other networks connected to that router.

Some backbone-in-a-box routers can address this requirement by providing redundant power systems. In addition, many sites connect one power system to the local power grid and the other to an uninterruptable power supply. If router power fails, the router can continue to provide connectivity to each connected network.

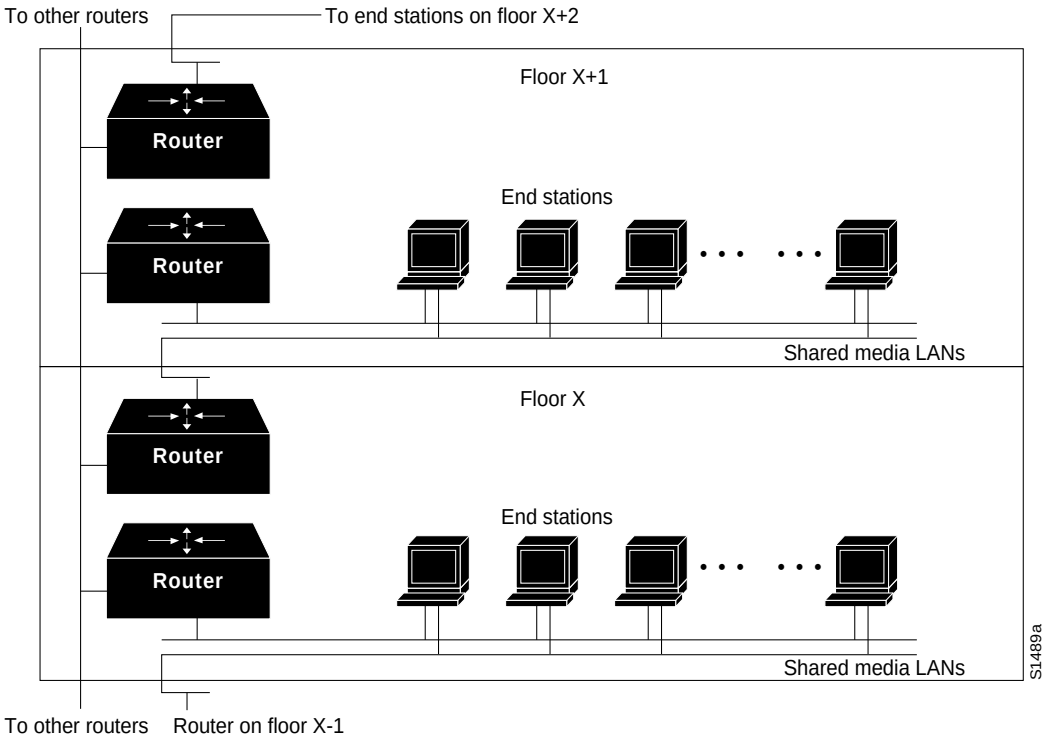
General power outages are usually more common than failures in a router's power system. Consider the effect of a site-wide power failure on redundant star and meshed topologies. If the power fails in the corporate office, the organization might be seriously inconvenienced. Key network applications are likely to be placed at a centralized, corporate location. The organization could easily lose revenue for every minute its network is down. The meshed network configuration is superior in this case, because links between the remote offices would still be able to communicate with each other.

If power fails at a remote site, all connections to that remote site will be terminated, unless otherwise protected. Neither the redundant star nor the meshed topology is superior. In both cases, all other remote offices will still be able to communicate with the corporate office. Generally, power failures in a remote office are more serious when network services are widely distributed.

To protect against local and site-wide power outages, some companies have negotiated an arrangement with local power companies to use multiple power grids within their organization. Failure within one power grid will not affect the network if all critical components have access to multiple power grids. Unfortunately, this arrangement is very expensive and should only be considered by companies with substantial resources, extremely mission-critical operations, and a relatively high likelihood of power failures.

The effect of highly localized power failures can be minimized with prudent network planning. Wherever possible, redundant components should use power supplied by different circuits. Further, these redundant components should not be physically colocated. For example, if redundant routers are employed for all stations on a given floor, these routers can be physically stationed in wiring closets on different floors. This prevents local wiring closet power problems from affecting the ability of all stations on a given floor to communicate. Figure 1-20 shows such a configuration.

Figure 1-20 Redundant Components on Different Floors



For some organizations, the need for fault tolerance is so great that potential power failures are protected against with a duplicate corporate data center. Organizations with these requirements often locate a redundant data center in another city, or in a part of the same city that is some distance from the primary data center. All backend services are duplicated, and transactions coming in from remote offices are sent to both data centers. This configuration requires duplicate WAN links from all remote offices, duplicate network hardware, duplicate servers and server resources, and leasing another building. Because this approach is so costly, it is typically the last step taken by companies desiring the ultimate in fault tolerance.

Partial duplication of the data center is also a possibility. Several key servers and links to those servers can be duplicated. This is a common compromise to the problem presented by power failures.

Fault-Tolerant Media Implementations

Media failure is yet another possible network fault. Included in this category are all problems associated with the media and its link to each individual end station. Under this definition, media components include network interface controller (NIC) failures, lobe or attachment unit interface (AUI) cable failures, transceiver failures, hub failures, and all failures associated with media components (for example, the cable itself, terminators, and other parts). Many media failures are caused by operator negligence and cannot easily be eliminated.

One way to reduce the havoc caused by failed media is to divide existing media into smaller segments and support each segment with different hardware. This minimizes the effect of a failure on a particular segment. For example, if you have 100 stations attached to a single hub, move some

of them to other hubs. This reduces the effect of a hub failure and of certain subnetwork failures. If you place an internetworking device (such as a router) between segments, you protect against additional problems and cut subnetwork traffic.

As shown in Figure 1-20, redundancy can be employed to help minimize media failures. Each station in this figure is attached to two different media segments. NICs, hub ports, and interface cables are all redundant. This approach doubles the cost of network connectivity for each end station as well as the port usage on all internetworking devices, and is therefore only recommended in situations where complete redundancy is required. It also assumes that end station software, including both the network and the application subsystems, can handle and effectively use the redundant components. The application software or the networking software or both must be able to detect network failures and initiate use of the other network.

Certain media access protocols have some fault-tolerant features built in. Token Ring multistation access units (MAUs) can detect certain media connection failures and bypass the failure internally. FDDI dual rings can wrap traffic onto the backup ring to avoid portions of the network with problems. Partially as a result of their fault tolerant characteristics, use of these protocols is increasing within companies with high reliability needs.

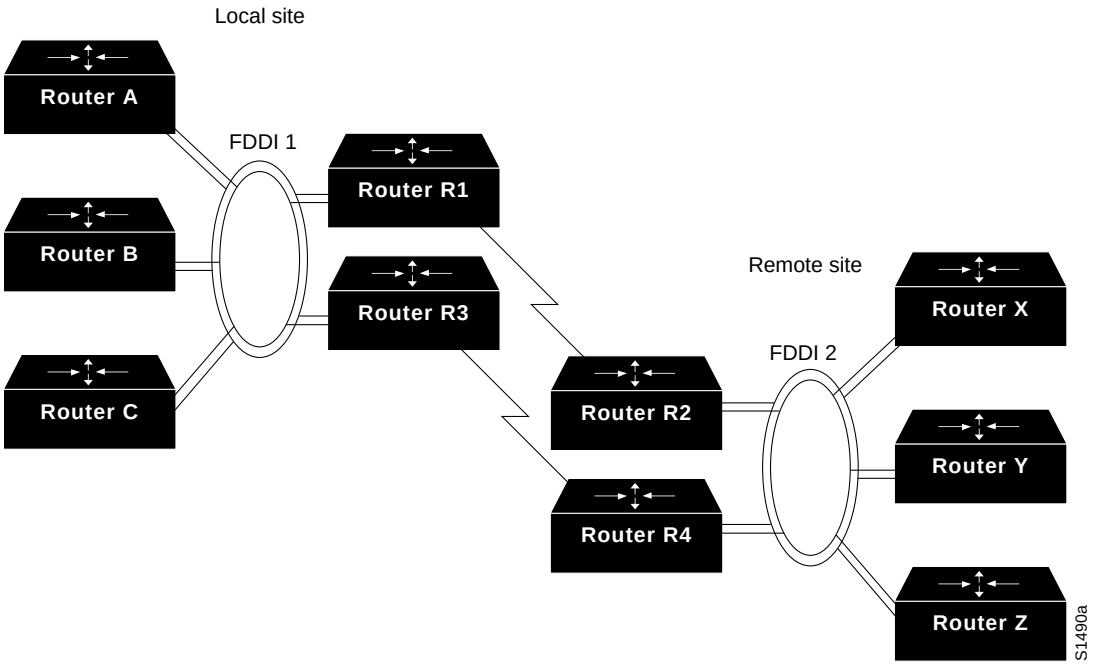
From a router's standpoint, many media failures can be bypassed so long as alternate paths are available. Using various hardware detection techniques, routers can sense certain media-level problems. If routing updates or routing keepalive messages have not been received from devices that would normally be reached through a particular router port, the router will soon declare that route down and will look for alternate routes. Meshed networks provide these alternate paths, allowing the router to compensate for media failures.

Backup Router Hardware

Like all other complex devices, routers and other internetworking devices develop hardware problems. Where serious failures do occur, the use of dual routers can effectively reduce adverse application availability effects. After a router failure, discovery protocols help end stations choose new local routers with which to communicate. If each network connected to the failed router has an alternate path out of the local area, complete connectivity will still be possible.

When backup routers are used, routing metrics can be set to ensure that the backup routers will not be used unless the primary routers are not functioning. Switchover is automatic and rapid. As an example, consider the situation shown in Figure 1-21. In this network, dual routers are used at all sites, with dual WAN links. If Router R1 fails, the routers on FDDI 1 will detect the failure by the absence of messages from Router R1. Using any of a number of dynamic routing protocols, Router A, Router B, and Router C will designate Router R3 as the new next hop on the way to remote resources accessible via Router R4.

Figure 1-21 Redundant FDDI Router Configuration



Many networks are designed with multiple routers connecting particular LANs in order to provide redundancy. In the past, the effectiveness of this design was limited by the speed at which the hosts on those LANs detected a topology update and changed routers. In particular, IP hosts tend to be configured with a default gateway or configured to use Proxy ARP in order to find a router on their LAN. Convincing an IP host to change its router usually required manual intervention in order to clear the ARP cache or to change the default gateway.

The Hot Standby Router Protocol (HSRP) provides a solution which allows network topology changes to be transparent to the host. HSRP will typically allow hosts to reroute in approximately 10 seconds. HSRP is supported on Ethernet, Token Ring, and FDDI. An HSRP group can be defined on each LAN. All members of the group know the standby IP address and the standby MAC address. One member of the group is elected the leader. The lead router services all packets sent to the HSRP group address. The other routers monitor the leader and act as HSRP routers. If the lead router becomes unavailable, the HSRP router elects a new leader who inherits the HSRP MAC address and IP address.

High-end routers (Cisco 7000 and AGS+ families) can support multiple MAC addresses on the same Ethernet or FDDI interface, allowing the routers to simultaneously handle both traffic that is sent to the standby MAC address and the private MAC address.

The commands used to enable HSRP and to configure an HSRP group are **standby ip** and **standby group**, respectively.

CHAPTER 2

Designing Large-Scale IP Internetworks

This chapter focuses on the following design implications with regard to the Enhanced Interior Gateway Routing Protocol (IGRP) and Open Shortest Path First (OSPF) protocols, which are routing protocols for the Internet Protocol (IP):

- Network topology
- Addressing and route summarization
- Route selection
- Convergence
- Network scalability
- Security

Information provided in this chapter is organized around these central topics. An introductory discussion outlines general routing protocol issues; subsequent discussions focus on design guidelines for the specific IP protocols.

Implementing Routing Protocols

The following discussion provides an overview of the key decisions you must make when selecting and deploying routing protocols. This discussion lays the foundation for subsequent discussions regarding specific routing protocols.

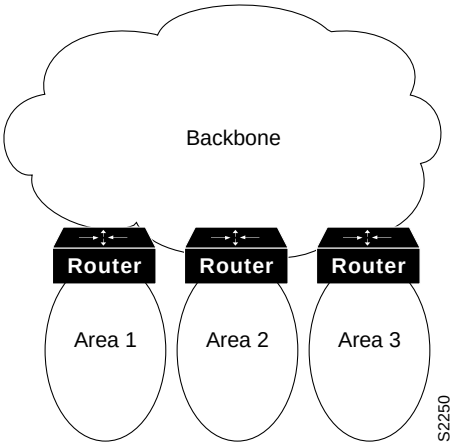
Network Topology

The physical topology of an internetwork is described by the complete set of routers and the networks that connect them. Networks also have a logical topology. Different routing protocols establish the logical topology in different ways.

Some routing protocols do not use a logical hierarchy. Such protocols use addressing to segregate specific areas or domains within a given internetworking environment and to establish a logical topology. For such nonhierarchical, or flat, protocols, no manual topology creation is required.

Other protocols require the creation of an explicit hierarchical topology through establishment of a backbone and logical areas. The OSPF and Intermediate System-to-Intermediate System (IS-IS) protocols are examples of routing protocols that use a hierarchical structure. A general hierarchical network scheme is illustrated in Figure 2-1. The explicit topology in a hierarchical scheme takes precedence over the topology created through addressing.

Figure 2-1 Hierarchical Network



If a hierarchical routing protocol is used, the addressing topology should be assigned to reflect the hierarchy. If a flat routing protocol is used, the addressing implicitly creates the topology.

There are two recommended ways to assign addresses in a hierarchical network. The simplest way is to give each area (including the backbone) a unique network address. An alternative is to assign address ranges to each area.

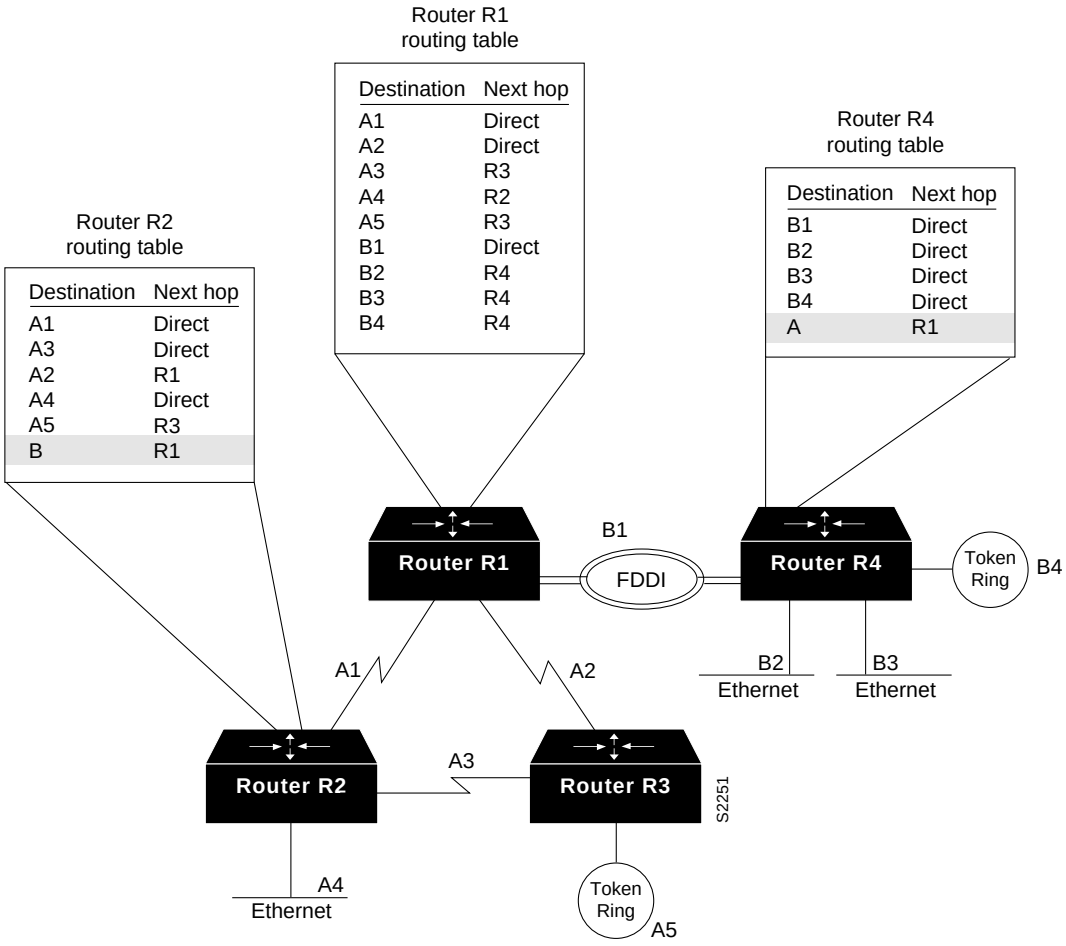
Areas are logical collections of contiguous networks and hosts. Areas also include all the routers having interfaces on any one of the included networks. Each area runs a separate copy of the basic routing algorithm. This means that each area has its own topological database.

Addressing and Route Summarization

Route summarization procedures condense routing information. Without summarization, each router in a network must retain a route to every subnet in the network. With summarization, routers can reduce some sets of routes to a single advertisement, thereby reducing both the load on the router and the perceived complexity of the network. The importance of route summarization grows with network size.

Figure 2-2 illustrates an example of route summarization. In this environment, Router R2 maintains one route for all destination networks beginning with “B” and Router R4 maintains one route for all destination networks beginning with “A.” This is the essence of route summarization. Router R1 tracks all routes because it exists on the boundary between “A” and “B.”

Figure 2-2 Route Summarization Example

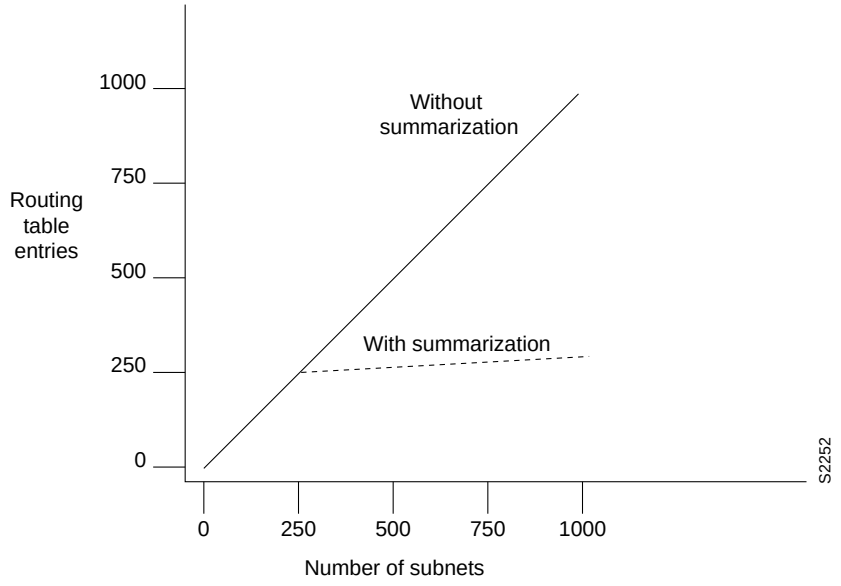


The reduction in route propagation and routing information overhead can be significant. Figure 2-3 illustrates the potential savings. The vertical axis of Figure 2-3 shows the number of routing table entries. The horizontal axis measures the number of subnets. Without summarization, each router in a network with 1000 subnets must contain 1000 routes. With summarization, the picture changes considerably. If you assume a Class B network with 8 bits of subnet address space, each router needs to know all of the routes for each subnet in its network number (250 routes, assuming that 1000 subnets fall into four major networks of 250 routers each) plus one route for each of the other networks (3) for a total of 253 routes. This represents a nearly 75 percent reduction in the size of the routing table.

The preceding example shows the simplest type of route summarization: collapsing all the subnet routes into a single network route. Some routing protocols also support route summarization at any bit boundary (rather than just at major network number boundaries) in a network address. A routing protocol can only summarize on a bit boundary if it supports *variable-length subnet masks* (VLSMs).

Some routing protocols summarize automatically. Other routing protocols require manual configuration to support route summarization.

Figure 2-3 Route Summarization Benefits

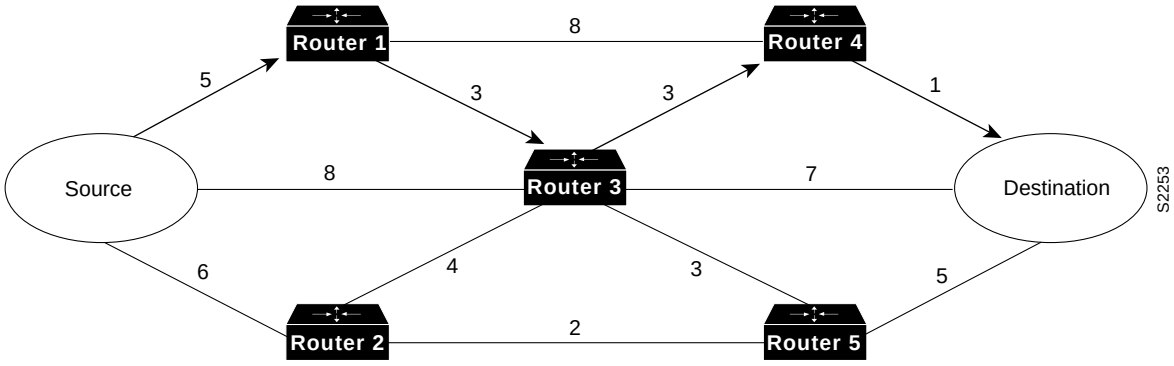


Route Selection

Route selection is trivial when only a single path to the destination exists. However, if any part of that path should fail, there is no way to recover. Therefore, most networks are designed with multiple paths so there are alternatives in case a failure occurs.

Routing protocols compare route metrics to select the best route from a group of possible routes. Route metrics are computed by assigning a characteristic or set of characteristics to each physical network. The metric for the route is an aggregation of the characteristics of each physical network in the route. Figure 2-4 shows a typical meshed network with metrics assigned to each link and the best route from source to destination identified.

Figure 2-4 Routing Metrics and Route Selection



Routing protocols use different techniques for assigning metrics to individual networks. Further, each routing protocol forms a metric aggregation in a different way.

Most routing protocols can use multiple paths if the paths have an equal cost. Some routing protocols can even use multiple paths when paths have an unequal cost. In either case, load balancing can improve overall allocation of network bandwidth.

When multiple paths are used, there are several ways to distribute the packets. The two most common mechanisms are per-packet load balancing and per-destination load balancing. Per-packet load balancing distributes the packets across the possible routes in a manner proportional to the route metrics. With equal-cost routes, this is equivalent to a round-robin scheme. One packet or destination (depending on switching mode) is distributed to each possible path. Per-destination load balancing distributes packets across the possible routes based on destination. Each new destination is assigned the next available route. This technique tends to preserve packet order.

Note Most TCP implementations can accommodate out-of-order packets. However, out-of-order packets may cause performance degradation.

When fast switching is enabled on a router (default condition), route selection is done on a per-destination basis. When fast switching is disabled, route selection is done on a per-packet basis. For line speeds of 56 kbps and faster, fast switching is recommended.

Convergence

When *network* topology changes, network traffic must reroute quickly. The phrase "convergence time" describes the time it takes a router to start using a new route after a topology changes.

Routers must do three things after a topology changes:

- Detect the change.
- Select a new route.
- Propagate the changed route information.

Some changes are immediately detectable. For example, serial line failures that involve carrier loss are immediately detectable by a router. Other failures are harder to detect. For example, if a serial line becomes unreliable but carrier is not lost, the unreliable link is not immediately detectable. In addition, some media (Ethernet, for example) do not provide physical indications such as carrier loss. When a router is reset, other routers do not detect this immediately. In general, failure detection is dependent on the media involved and the routing protocol used.

Once a failure has been detected, the routing protocol must select a new route. The mechanisms used to do this are protocol dependent. All routing protocols must propagate the changed route. The mechanisms used to do this are also protocol dependent.

Network Scalability

The ability to extend your internetwork is determined, in part, by the scaling characteristics of the routing protocols used and the quality of the network design.

Network scalability is limited by two factors: operational issues and technical issues. Typically, operational issues are more significant than technical issues. Operational scaling concerns encourage the use of large areas or protocols that do not require hierarchical structures. When hierarchical protocols are required, technical scaling concerns promote the use of small areas. Finding the right balance is the art of network design.

From a technical standpoint, routing protocols scale well if their resource use grows less than linearly with the growth of the network. Three critical resources are used by routing protocols: memory, central processing unit (CPU), and bandwidth.

Memory

Routing protocols use memory to store routing tables and topology information. Route summarization cuts memory consumption for all routing protocols. Keeping areas small reduces the memory consumption for hierarchical routing protocols.

CPU

CPU usage is protocol dependent. Some protocols use CPU cycles to compare new routes to existing routes. Other protocols use CPU cycles to regenerate routing tables after a topology change. In most cases, the latter technique will use more CPU cycles than the former. For link state protocols, keeping areas small and using summarization reduces CPU requirements by reducing the effect of a topology change and by decreasing the number of routes that must be recomputed after a topology change.

Bandwidth

Bandwidth usage is also protocol dependent. Three key issues determine the amount of bandwidth a routing protocol consumes:

- When routing information is sent—Periodic updates are sent at regular intervals. Flash updates are sent only when a change occurs.
- What routing information is sent—Complete updates contain all routing information. Partial updates contain only changed information.
- Where routing information is sent—Flooded updates are sent to all routers. Bounded updates are sent only to routers that are affected by a change.

Note These three issues also affect CPU usage.

Security

Controlling access to network resources is a primary concern. Some routing protocols provide techniques that can be used as part of a security strategy.

With some routing protocols, you can insert a filter on the routes being advertised so that certain routes are not advertised in some parts of the network.

Some routing protocols have the ability to authenticate routers that are running the same protocol. Authentication mechanisms are protocol specific and generally weak. In spite of this, it is worthwhile to take advantage of the techniques that exist. Authentication mechanisms can increase network stability by preventing unauthorized routers or hosts from participating in the routing protocol, whether those devices are attempting to participate accidentally or deliberately.

Enhanced IGRP Internetwork Design Guidelines

The Enhanced Interior Gateway Routing Protocol (Enhanced IGRP) is a routing protocol developed by Cisco Systems and introduced with Software Release 9.21 and Internetworking Operating System (IOS) 10.0. Enhanced IGRP combines the advantages of distance-vector protocols, such as IGRP, with the advantages of link state protocols, such as Open Shortest Path First (OSPF). Enhanced IGRP uses the Diffusing Update ALgorithm (DUAL) to achieve convergence quickly.

Enhanced IGRP includes support for IP, Novell NetWare, and AppleTalk. The discussion on Enhanced IGRP is divided into the following sections:

- Network topology
- Addressing
- Route summarization
- Route selection
- Convergence
- Network scalability
- Security

Note Although the general discussion covered in this section is applicable to IP, IPX, and AppleTalk Enhanced IGRP, IP issues are highlighted here. For case studies on how to integrate Enhanced IGRP into IP, IPX, and AppleTalk networks, including detailed configuration examples and protocol-specific issues, see Topic 1, "Integrating Enhanced IGRP into Existing Networks" in the Cisco publication *Internetwork Applications: Case Studies, Vol. 1 No. 2*.

Caution If you are using *candidate default route* in IP Enhanced IGRP and have installed multiple releases of Cisco router software within your internetwork that include any versions prior to September 1994, contact your Cisco technical support representative for version compatibility and software upgrade information. Refer to your software release notes for details.

Enhanced IGRP Network Topology

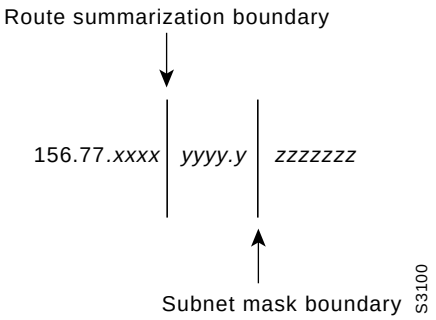
Enhanced IGRP uses a non-hierarchical (or flat) topology by default. Enhanced IGRP automatically summarizes subnet routes of directly connected networks at a network number boundary. This automatic summarization is sufficient for most IP networks. See the section "Enhanced IGRP Route Summarization" later in this chapter for more detail.

Enhanced IGRP Addressing

The first step in designing an Enhanced IGRP network is to decide on how to address the network. In many cases, a company is assigned a single NIC address (such as a Class B network address) to be allocated in a corporate internetwork. Bit-wise subnetting and variable-length subnetwork masks (VLSMs) can be used in combination to save address space. Enhanced IGRP for IP supports the use of VLSMs.

Consider a hypothetical network where a Class B address is divided into subnetworks, and contiguous groups of these subnetworks are summarized by Enhanced IGRP. The Class B network 156.77.0.0 might be subdivided as illustrated in Figure 2-5.

Figure 2-5 Variable-Length Subnet Masks (VLSMs) and Route Summarization Boundaries



In Figure 2-5, the letters x, y, and z represent bits of the last two octets of the Class B network as follows:

- The four *x* bits represent the route summarization boundary.
- The five *y* bits represent up to 32 subnets per summary route.
- The seven *z* bits allow for 126 (128-2) hosts per subnet.

Appendix A, "Subnetting an IP Address Space," provides a complete example illustrating assignment for the Class B address 150.100.0.0.

Enhanced IGRP Route Summarization

With Enhanced IGRP, subnet routes of directly connected networks are automatically summarized at network number boundaries. In addition, a network administrator can configure route summarization at any interface with any bit boundary, allowing ranges of networks to be summarized arbitrarily.

Enhanced IGRP Route Selection

Routing protocols compare route metrics to select the best route from a group of possible routes. The following factors are important to understand when designing an Enhanced IGRP internetwork.

Enhanced IGRP uses the same vector of metrics as IGRP. Separate metric values are assigned for bandwidth, delay, reliability and load. By default, Enhanced IGRP computes the metric for a route by using the minimum bandwidth of each hop in the path and adding a media-specific delay for each hop. The metrics used by Enhanced IGRP are as follows:

- Bandwidth

  Bandwidth is deduced from the interface type. Bandwidth can be modified with the **bandwidth** command.

- Delay

  Each media type has a propagation delay associated with it. Modifying delay is very useful to optimize routing in network with satellite links. Delay can be modified with the **delay** command.

- Reliability

  Reliability is dynamically computed as a rolling weighted average over five seconds.

- Load

  Load is dynamically computed as a rolling weighted average over five seconds.

When Enhanced IGRP summarizes a group of routes, it uses the metric of the best route in the summary as the metric for the summary.

Note For information on Enhanced IGRP load sharing, see the section "IP Routing Protocols with Parallel Links" in Chapter 3, "Designing SRB Internetworks."

Enhanced IGRP Convergence

Enhanced IGRP implements a new convergence algorithm known as DUAL (Diffusing Update ALgorithm). DUAL uses two techniques that allow Enhanced IGRP to converge very quickly. First, each Enhanced IGRP router stores its neighbors routing tables. This allows the router to use a new route to a destination instantly if another *feasible* route is known. If no feasible route is known based upon the routing information previously learned from its neighbors, a router running Enhanced IGRP becomes *active* for that destination and sends a query to each of its neighbors asking for an alternate route to the destination. These queries propagate until an alternate route is found. Routers that are not affected by a topology change remain *passive* and do not need to be involved in the query and response.

A router using Enhanced IGRP receives full routing tables from its neighbors when it first communicates with the neighbors. Thereafter, only *changes* to the routing tables are sent and only to *routers* that are *affected* by the change. A *successor* is a neighboring router that is currently being used for packet forwarding, provides the *least cost* route to the destination, and is not part of a routing loop. Information in the routing table is based on *feasible successors*. Feasible successor routes can be used in case the existing route fails. Feasible successors provide the *next least-cost* path without introducing routing loops.

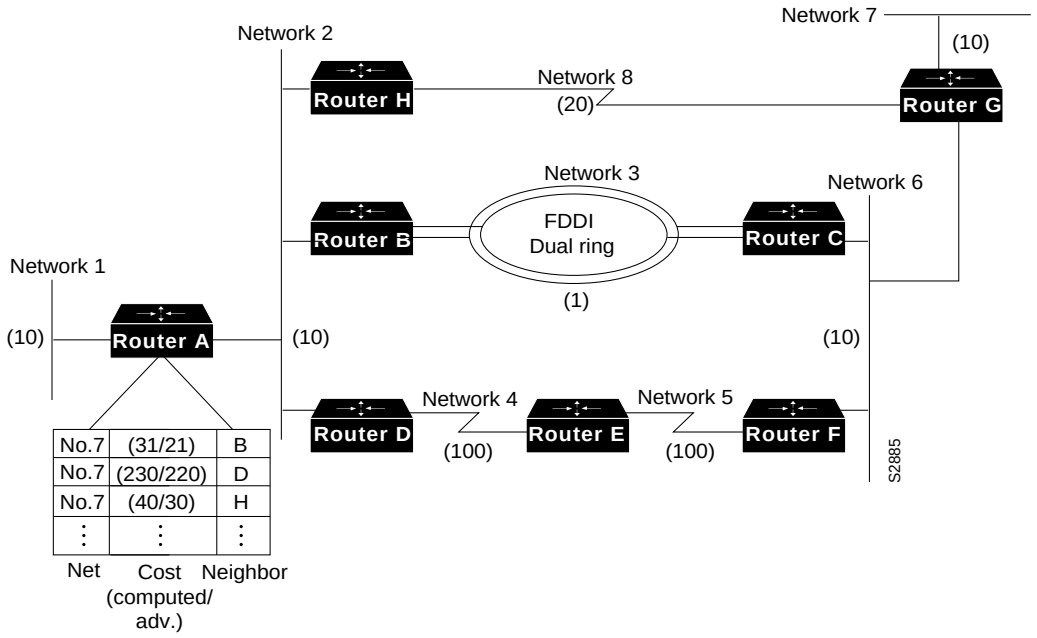
The routing table keeps a list of the computed costs of reaching networks. The topology table keeps a list of all routes advertised by neighbors. For each network, the router keeps the real cost of getting to that network and also keeps the advertised cost from its neighbor. In the event of a failure, convergence is instant if a feasible successor can be found. A neighbor is a feasible successor if it meets the feasibility condition set by DUAL. DUAL finds feasible successors by the performing the following computations:

1. Determines membership of V_1 . V_1 is the set of all neighbors whose advertised distance to network *x* is less than FD. (FD is the feasible distance and is defined as the best metric during an active-to-passive transition.)
2. Calculates $D_{min.}$ D_{min} is the minimum computed cost to network *x*.
3. Determines membership of V_2 . V_2 is the set of neighbors that are in V1 whose computed cost to network *x* equals D_{min}.

The feasibility condition is met when V_2 has one or more members.

The concept of feasible successors is illustrated in Figure 2-6. Consider Router A's topology table entries for network 7. Router B is the *successor* with a computed cost of 31 to reach network 7, compared to the computed costs of Router D (230) and Router H (40).

Figure 2-6 DUAL Feasible Successor



If Router B becomes unavailable, Router A will go through the following 3-step process to find a feasible successor for network 7:

Step 1 Determining which neighbors have an advertised distance to network 7 that is less than Router A's feasible distance (FD) to network 7. The FD is 31 and Router H meets this condition. Therefore, Router H is a member of V_1.

Step 2 Calculating the minimum computed cost to Network 7. Router H provides a cost of 40, and Router D provides a cost of 230. D_{min} is, therefore, 40.

Step 3 Determining the set of neighbors that are in V_1 whose computed cost to network 7 equals D_{min} (40). Router H meets this condition.

The feasible successor is Router H which provides a least cost route of 40 from Router A to network 7.

If Router H, now, also becomes unavailable, Router A performs the following computations:

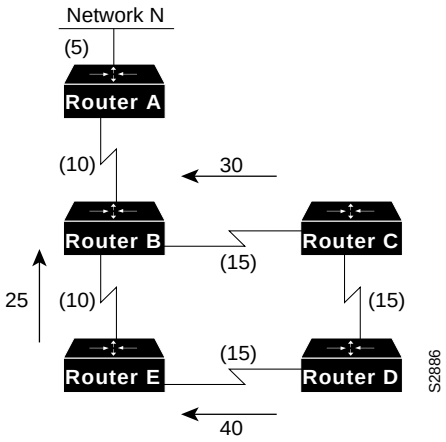
Step 1 Determines which neighbors have an advertised distance to network 7 that is less than the FD for network 7. Because both Router B and H have become unavailable, only Router D remains. However, the advertised cost of Router D to network 7 is 220 which is greater than Router A's FD (31) to network 7. Router D, therefore, cannot be a member of V_1. The FD remains at 31—the FD can only change during an active-to-passive transition, and this did not occur. There was no transition to active state for network 7; this is known as a *local computation*.

Step 2 Because there are no members of V1, there can be no feasible successors. Router A, therefore, transitions from passive to active state for network 7, and queries its neighbors about network 7. There was a transition to active; this is known as a *diffusing computation*.

Note For more details on Enhanced IGRP convergence, see Appendix E, "References and Recommended Reading," for a list of reference papers and materials.

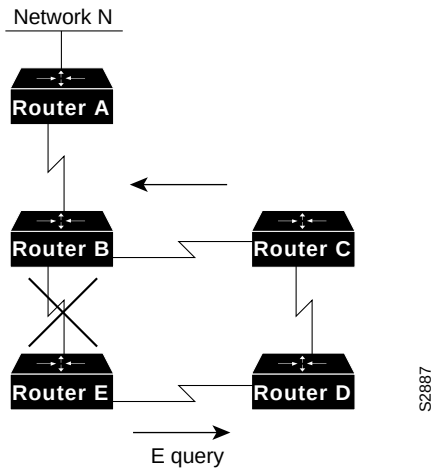
The following example and graphics further illustrate how Enhanced IGRP supports virtually instantaneous convergence in a changing internetwork environment. In Figure 2-7, all routers can access each other and Network N. The computed cost to reach other routers and Network N is shown. For example, the cost from Router E to Router B is 10. The cost from Router E to network N is 25 (cumulative of 10 + 10 + 5 = 25).

Figure 2-7 DUAL Example (part 1): Initial Network Connectivity



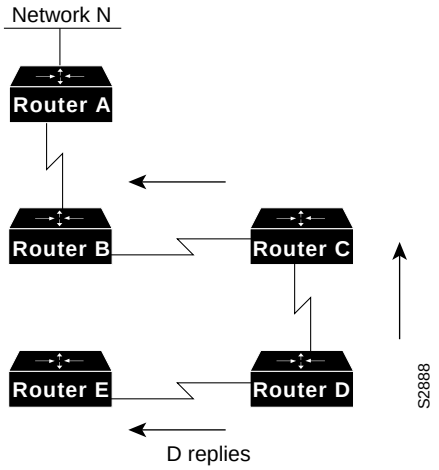
In Figure 2-8, the connection between Router B and Router E fails. Router E sends a multicast query to all of its neighbors and puts network N into an active state.

Figure 2-8 DUAL Example (part 2): Sending Queries



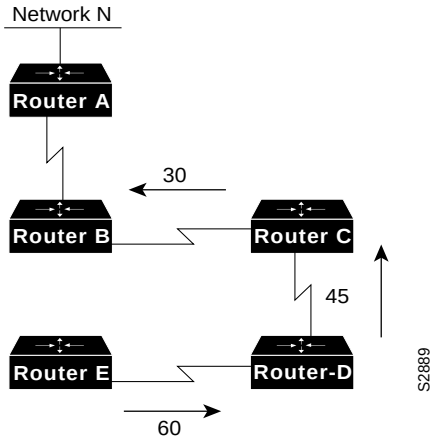
Next, as illustrated in Figure 2-9, Router D determines that it has a feasible successor. It changes its successor from router E to router C, and sends a reply to router E.

Figure 2-9 DUAL Example (part 3): Switching to a Feasible Successor



In Figure 2-10, Router E has received replies from all neighbors and therefore brings network N out of active state. Router E puts network N into its routing table at a distance of 60.

Figure 2-10 DUAL Example (part 4): Final Network Connectivity



Note Router A, Router B, and Rouber C were not involved in route recomputation. Router D recomputed its path to network N, but without first needing to learn new routing information from its downstream neighbors.

Enhanced IGRP Network Scalability

Network scalability is limited by two factors: operational issues and technical issues. Operationally, Enhanced IGRP provides easy configuration and growth. Technically, Enhanced IGRP uses resources at less than a linear rate with the growth of a network.

Memory

A router running Enhanced IGRP stores all routes advertised by neighbors so that it can adapt quickly to alternate routes. The more neighbors a router has, the more memory a router uses. Enhanced IGRP automatic route aggregation bounds the routing table growth naturally. Additional bounding is possible with manual route aggregation.

CPU

Enhanced IGRP uses the DUAL algorithm to provide fast convergence. DUAL recomputes only routes which are affected by a topology change. DUAL is not computationally complex, so it does not require a lot of CPU.

Bandwidth

Enhanced IGRP uses partial updates. Partial updates are generated only when a change occurs; only the changed information is sent, and this changed information is sent only to the routers affected. Because of this, Enhanced IGRP is very efficient in its usage of bandwidth. Some additional bandwidth is used by Enhanced IGRP's HELLO protocol to maintain adjacencies between neighboring routers.

Enhanced IGRP Security

Enhanced IGRP is available only on Cisco routers. This prevents accidental or malicious routing disruption caused by hosts in a network.

In addition, route filters can be set up on any interface to prevent learning or propagating routing information inappropriately.

OSPF Internetwork Design Guidelines

OSPF is an Interior Gateway Protocol (IGP) developed for use in Internet Protocol (IP)-based internetworks. As an IGP, OSPF distributes routing information between routers belonging to a single autonomous system (AS). An AS is a group of routers exchanging routing information via a common routing protocol. The OSPF protocol is based on shortest-path-first, or link state, technology.

The OSPF protocol was developed by the OSPF working group of the Internet Engineering Task Force (IETF). It was designed expressly for the Internet Protocol (IP) environment, including explicit support for IP subnetting and the tagging of externally derived routing information. OSPF Version 2 is documented in Request for Comments (RFC) 1247.

Whether you are building an OSPF internetwork from the ground up or converting your internetwork to OSPF, the following design guidelines provide a foundation from which you can construct a reliable, scalable OSPF-based environment.

Two design activities are critically important to a successful OSPF implementation:

- Definition of area boundaries
- Address assignment

Ensuring that these activities are properly planned and executed will make all the difference in your OSPF implementation. Each is addressed in more detail with the discussions that follow. These discussions are divided into six sections:

- OSPF network topology
- OSPF addressing and route summarization
- OSPF route selection
- OSPF convergence
- OSPF network scalability
- OSPF security

Note For a detailed case study on how to set up and configure RIP and OSPF redistribution, see Topic 1, "RIP and OSPF Redistribution" in the *Internetworking Applications: Case Studies* publication, Vol. 1 No. 1.

OSPF Network Topology

OSPF works best in a hierarchical routing environment. The first and most important decision when designing an OSPF network is to determine which routers and links are to be included in the backbone and which are to be included in each area.

There are several important guidelines to consider when designing an OSPF topology:

- **The number of routers in an area**—OSPF uses a CPU-intensive algorithm. The number of calculations that must be performed given *n* link state packets is proportional to *n log n*. As a result, the larger and more unstable the area, the greater the likelihood for performance problems associated with routing protocol recalculation. Generally, an area should have no more than 50 routers. Areas with unstable links should be smaller.
- **The number of neighbors for any one router**—OSPF floods all link state changes to all routers in an area. Routers with many neighbors have the most work to do when link state changes occur. In general, any one router should have no more than 60 neighbors.
- **The number of areas supported by any one router**—A router must run the link state algorithm for each link state change that occurs for every area in which the router resides. Every area border router is in at least two areas (the backbone and one area). In general, to maximize stability, one router should not be in more than three areas.
- **Designated router selection**—In general, the designated router and backup designated router on a local-area network (LAN) have the most OSPF work to do. It is a good idea to select routers that are not already heavily loaded with CPU intensive activities to be the designated router and backup designated router. In addition, it is generally not a good idea to select the same router to be designated router on many LANs simultaneously.

The discussions that follow address topology issues that are specifically related to the backbone and the areas.

Backbone Considerations

Stability and *redundancy* are the most important criteria for the backbone. Stability is increased by keeping the size of the backbone reasonable. This is caused by the fact that every router in the backbone needs to recompute its routes after every link state change. Keeping the backbone small reduces the likelihood of a change and reduces the amount of CPU cycles required to recompute routes. As a general rule, it is a good idea to have each area (including the backbone) contain no more than 50 routers. If link quality is high and the number of routes is small, the number of routers can be increased.

Redundancy is important in the backbone to prevent partition when a link fails. Good backbones are designed so that no single link failure can cause a partition.

OSPF backbones must be contiguous. All routers in the backbone should be directly connected to other backbone routers. OSPF includes the concept of virtual links. A virtual link creates a path between two area border routers (an area border router is a router connects an area to the backbone) that are not directly connected. A virtual link can be used to heal a partitioned backbone. However, it is not a good idea to design an OSPF network to require the use of virtual links. The stability of a virtual link is determined by the stability of the underlying area. This dependency can make troubleshooting more difficult. In addition, virtual links cannot run across stub areas. See the section "Backbone-to-Area Route Advertisement," later in this chapter for a detailed discussion of stub areas.

Avoid placing hosts (such as workstations, file servers or other shared resources) in the backbone area. Keeping hosts out of the backbone area simplifies internetwork expansion and creates a more stable environment.

Area Considerations

Individual areas must be contiguous. In this context, a contiguous area is one in which a continuous path can be traced from any router in an area to any other router in the same area. This does not mean that all routers must share a common network media. It is not possible to use virtual links to connect a partitioned area. Ideally, areas should be richly connected internally to prevent partitioning.

The two most critical aspects of area design follow:

- Determining how the area is addressed
- Determining how the area is connected to the backbone

Areas should have a contiguous set of network and/or subnet addresses. Without a contiguous address space, it is not possible to implement route summarization. The routers that connect an area to the backbone are called *area border routers*. Areas can have a single area border router or they can have multiple area border routers. In general, it is desirable to have more than one area border router per area to minimize the chance of the area becoming disconnected from the backbone.

When creating large-scale OSPF internetworks, the definition of areas and assignment of resources within areas must be done with a pragmatic view of your internetwork. The following are general rules that will help ensure that your internetwork remains flexible and provides the kind of performance needed to deliver reliable resource access.

- Consider physical proximity when defining areas—If a particular location is densely connected, create an area specifically for nodes at that location.
- Reduce the maximum size of areas if links are unstable—If your internetwork includes unstable links, consider implementing smaller areas to reduce the effects of route flapping. Whenever a route is lost or comes online, each affected area must converge on a new topology. The Dykstra algorithm will run on all the affected routers. By segmenting your internetwork into smaller areas, you can isolate unstable links and deliver more reliable overall service.

OSPF Addressing and Route Summarization

Address assignment and route summarization are inextricably linked when designing OSPF internetworks. To create a scalable OSPF internetwork, you should implement route summarization. To create an environment capable of supporting route summarization, you must implement an effective hierarchical addressing scheme. The addressing structure that you implement can have a profound impact on the performance and scalability of your OSPF internetwork. The following sections discuss OSPF route summarization and three addressing options:

- Separate network numbers for each area
- Network Information Center (NIC)-authorized address areas created using bit-wise subnetting and VLSM
- Private addressing, with a "demilitarized zone" (DMZ) buffer to the official Internet world

---

Note You should keep your addressing scheme as simple as possible, but be wary of oversimplifying your address assignment scheme. Although simplicity in addressing saves time later when operating and troubleshooting your network, taking short cuts can have certain severe consequences. In building a scalable addressing environment, use a structured approach. If necessary, use bit-wise subnetting—but make sure that route summarization can be accomplished at the area border routers.

OSPF Route Summarization

Route summarization is extremely desirable for a reliable and scalable OSPF internetwork. The effectiveness of route summarization, and your OSPF implementation in general, hinges on the addressing scheme that you adopt. Summarization in an OSPF internetwork occurs between each area and the backbone area. Summarization must be configured manually in OSPF.

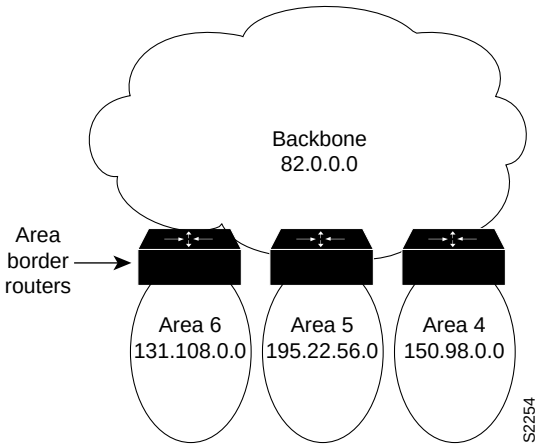
When planning your OSPF internetwork, consider the following issues:

- Be sure that your network addressing scheme is configured so that the range of subnets assigned within an area is contiguous.
- Create an address space that will permit you to split areas easily as your network grows. If possible, assign subnets according to simple octet boundaries. If you cannot assign addresses in an easy-to-remember and easy-to-divide manner, be sure to have a thoroughly defined addressing structure. If you know how your entire address space is assigned (or will be assigned), you can plan for changes more effectively.
- Plan ahead for the addition of new routers to your OSPF environment. Be sure that new routers are inserted appropriately as area, backbone, or border routers. Because the addition of new routers creates a new topology, inserting new routers can cause unexpected routing changes (and possibly performance changes) when your OSPF topology is recomputed.

Separate Address Structures for Each Area

One of the simplest ways to allocate addresses in OSPF is to assign a separate network number for each area. With this scheme, you create a backbone and multiple areas, and assign a separate IP network number to each area. Figure 2-11 illustrates this kind of area allocation.

Figure 2-11 Assignment of NIC Addresses Example



The following are the basic steps for creating such a network:

Step 1 Define your structure (identify areas and allocate nodes to areas).

Step 2 Assign addresses to networks, subnets, and end stations.

In the network illustrated in Figure 2-11, each area has its own unique NIC-assigned address. These can be Class A (the backbone in Figure 2-11), Class B (Areas 4 and 6), or Class C (Area 5).

The following are some clear benefits of assigning separate address structures to each area:

- Address assignment is relatively easy to remember.
- Configuration of routers is relatively easy and mistakes are less likely.
- Network operations are streamlined because each area has a simple, unique network number.

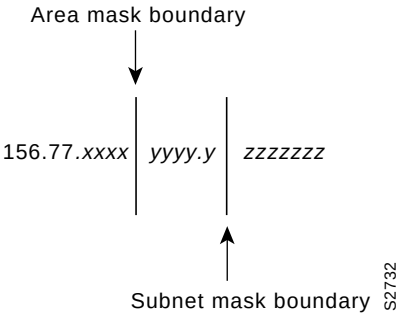
In the example illustrated in Figure 2-11, the route summarization configuration at the area border routers is greatly simplified. Routes from Area 4 injecting into the backbone can be summarized as follows: *all routes starting with 150.98 are found in Area 4.*

The main drawback of this approach to address assignment is that it wastes address space. If you decide to adopt this approach, be sure that area border routers are configured to do route summarization. Summarization must be explicitly set; it is disabled by default in OSPF.

Bit-Wise Subnetting and VLSM

Bit-wise subnetting and variable-length subnetwork masks (VLSMs) can be used in combination to save address space. Consider a hypothetical network where a Class B address is subdivided using an area mask and distributed among 16 areas. The Class B network, 156.77.0.0, might be subdivided as illustrated in Figure 2-12.

Figure 2-12 Areas and Subnet Masking



In Figure 2-12, the letters *x*, *y*, and *z* represent bits of the last two octets of the Class B network as follows:

- The four *x* bits are used to identify 16 areas.
- The five *y* bits represent up to 32 subnets per area.
- The seven *z* bits allow for 126 (128-2) hosts per subnet.

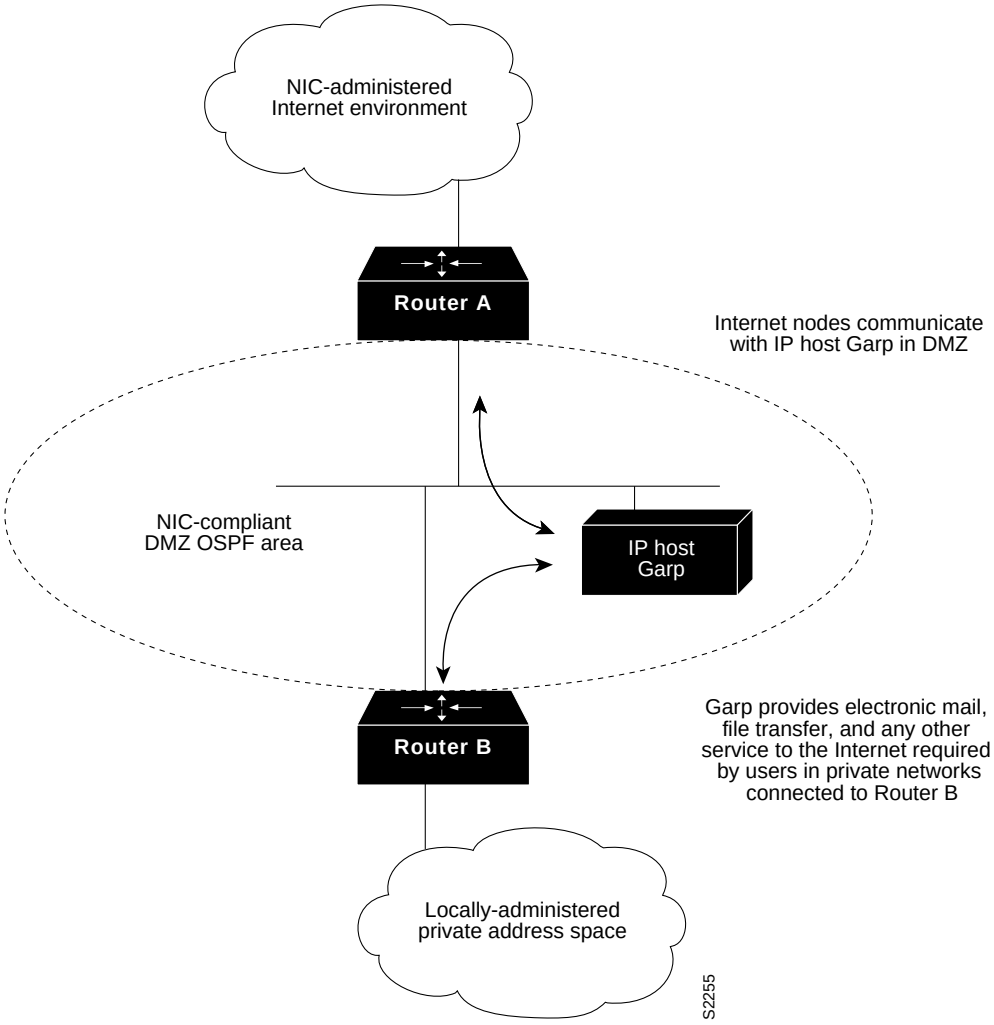
Appendix A, “Subnetting an IP Address Space” provides a complete example illustrating assignment for the Class B address 150.100.0.0. It illustrates both the concept of *area masks* and the breakdown of large subnets into smaller ones using VLSMs.

Private Addressing

Private addressing is another option often cited as simpler than developing an area scheme using bit-wise subnetting. Although private address schemes provide an excellent level of flexibility and do not limit the growth of your OSPF internetwork, they have certain disadvantages. For instance, developing a large-scale internetwork of privately addressed IP nodes limits total access to the Internet, and mandates the implementation of what is referred to as a *demilitarized zone* (DMZ). If you need to connect to the Internet, Figure 2-13 illustrates the way in which a DMZ provides a buffer of valid NIC nodes between a privately addressed network and the Internet.

All nodes (end systems and routers) on the network in the DMZ must have NIC-assigned IP addresses. The NIC might, for example, assign a single Class C network number to you. The DMZ shown in Figure 2-13 has two routers and a single application gateway host (Garp). Router A provides the interface between the DMZ and the Internet, and Router B provides the firewall between the DMZ and the private address environment. All applications that need to run over the Internet must access the Internet through the application gateway.

Figure 2-13 Connecting to the Internet from a Privately Addressed Network



Note For a case study on network security that includes information on how to set up firewall routers and communication servers, see Topic 3, "Increasing Security on IP Networks" in the *Internetwork Applications: Case Studies* publication, Vol. 1 No. 1.

Route Summarization Techniques

Route summarization is particularly important in an OSPF environment because it increases the stability of the network. If route summarization is being used, routes within an area that change do not need to be changed in the backbone or in other areas.

Route summarization addresses two important questions of route information distribution:

- What information does the backbone need to know about each area? The answer to this question focuses attention on area-to-backbone routing information.
- What information does each area need to know about the backbone and other areas? The answer to this question focuses attention on backbone-to-area routing information.

Area-to-Backbone Route Advertisement

There are several key considerations when setting up your OSPF areas for proper summarization:

- OSPF route summarization occurs in the area border routers.
- OSPF supports VLSM, so it is possible to summarize on any bit boundary in a network or subnet address.
- OSPF requires manual summarization. As you design the areas, you need to determine summarization at each area border router.

Backbone-to-Area Route Advertisement

There are four potential types of routing information in an area:

- Default. If an explicit route cannot be found for a given IP network or subnetwork, the router will forward the packet to the destination specified in the default route.
- Intra-area routes. Explicit network or subnet routes must be carried for all networks or subnets inside an area.
- Interarea routes. Areas may carry explicit network or subnet routes for networks or subnets that are in this AS but not in this area.
- External routes. When different ASs exchange routing information, the routes they exchange are referred to as external routes.

In general, it is desirable to restrict routing information in any area to the minimal set that the area needs.

There are three types of areas, and they are defined in accordance with the routing information that is used in them:

- Non-stub areas—Non-stub areas carry a default route, static routes, intra-area routes, interarea routes and external routes. An area must be a nonstub area when it contains a router that uses both OSPF and any other protocol, such as the Routing Information Protocol (RIP). Such a router is known as an autonomous system border router (ASBR). An area must also be a nonstub area when a virtual link is configured across the area. Non-stub areas are the most resource-intensive type of area.
- Stub areas—Stub areas carry a default route, intra-area routes and interarea routes, but they do not carry external routes. Stub areas are recommended for areas that have only one area border router and they are often useful in areas with multiple area border routers. See “Controlling Interarea Traffic,” later in this chapter for a detailed discussion of the design trade-offs in areas with multiple area border routers.There are two restrictions on the use of stub areas: virtual links cannot be configured across them, and they cannot contain an ASBR.
- Stub areas without summaries—Software releases 9.1(11), 9.21(2), and 10.0(1) and later support stub areas without summaries, allowing you to create areas that carry only a default route and intra-area routes. Stub areas without summaries do not carry interarea routes or external routes. This type of area is recommended for simple configurations where a single router connects an area to the backbone.

Table 2-1 shows the different types of areas according to the routing information that they use.

Table 2-1 Routing Information Used in OSPF Areas

Area Type	Default Route	Intra-area Routes	Interarea Routes	External Routes
Nonstub	Yes	Yes	Yes	Yes
Stub	Yes	Yes	Yes	No
Stub without summaries	Yes	Yes	No	No

Note Stub areas are configured using the **area** *area-id* **stub** router configuration command. Routes are summarized using the **area** *area-id* **range** *address mask* router configuration command. Refer to your *Router Products Configuration Guide* and *Router Products Command Reference* publication for more information regarding the use of these commands.

OSPF Route Selection

When designing an OSPF internetwork for efficient route selection, consider three important topics:

- Tuning OSPF metrics
- Controlling interarea traffic
- Load balancing in OSPF internetworks

Tuning OSPF Metrics

The default value for OSPF metrics is based on bandwidth. The following characteristics show how OSPF metrics are generated:

- Each link is given a metric value based on its bandwidth. The metric for a specific link is the inverse of the bandwidth for that link. Link metrics are normalized to give FDDI a metric of 1. The metric for a route is the sum of the metrics for all the links in the route.

Note In some cases, your network might implement a media type that is faster than the fastest default media configurable for OSPF (FDDI). An example of a faster media is ATM. By default, a faster media will be assigned a cost equal to the cost of an FDDI link—a link state metric cost of 1. Given an environment with both FDDI and a faster media type, you must manually configure link costs to configure the faster link with a lower metric. Configure any FDDI link with a cost greater than 1, and the faster link with a cost less than the assigned FDDI link cost. Use the **ip ospf cost** interface configuration command to modify link state cost.

- When route summarization is enabled, OSPF uses the metric of the best route in the summary.
- There are two forms of external metrics: type 1 and type 2. Using an external type 1 metric results in routes adding the internal OSPF metric to the external route metric. External type 2 metrics do not add the internal metric to external routes. The external type 1 metric is generally preferred. If you have more than one external connection, either metric can affect how multiple paths are used.

Controlling Interarea Traffic

When an area has only a single area border router, all traffic that does not belong in the area will be sent to the area border router.

In areas that have multiple area border routers, two choices are available for traffic that needs to leave the area:

- Use the area border router closest to the originator of the traffic. (Traffic leaves the area as soon as possible.)
- Use the area border router closest to the destination of the traffic. (Traffic leaves the area as late as possible.)

If the area border routers inject only the default route, the traffic goes to the area border router that is closest to the source of the traffic. Generally, this behavior is desirable because the backbone typically has higher bandwidth lines available. However, if you want the traffic to use the area border router that is nearest the destination (so that traffic leaves the area as late as possible), the area border routers should inject summaries into the area instead of just injecting the default route.

Most network designers prefer to avoid asymmetric routing (that is, using a different path for packets that are going from A to B than for those packets that are going from B to A.) It is important to understand how routing occurs between areas to avoid asymmetric routing.

Load Balancing in OSPF Internetworks

Internetwork topologies are typically designed to provide redundant routes in order to prevent a partitioned network. Redundancy is also useful to provide additional bandwidth for high traffic areas. If equal-cost paths between nodes exist, Cisco routers automatically load balance in an OSPF environment.

Cisco routers can use up to four equal-cost paths for a given destination. Packets might be distributed either on a per-destination (when fast switching) or a per-packet basis. Per-destination load balancing is the default behavior. Per-packet load balancing can be enabled by turning off fast switching using the **no ip route-cache** interface configuration command. For line speeds of 56 kbps and faster, it is recommended that you enable fast switching.

OSPF Convergence

One of the most attractive features about OSPF is the ability to quickly adapt to topology changes.

There are two components to routing convergence:

- Detection of topology changes—OSPF uses two mechanisms to detect topology changes. Interface status changes (such as carrier failure on a serial link) is the first mechanism. The second mechanism is failure of OSPF to receive a hello packet from its neighbor within a timing window called a *dead timer*. Once this timer expires, the router assumes the neighbor is down. The dead timer is configured using the **ip ospf dead-interval** interface configuration command. The default value of the dead timer is four times the value of the Hello interval. That results in a dead timer default of 40 seconds for broadcast networks and 2 minutes for nonbroadcast networks.
- Recalculation of routes—Once a failure has been detected, the router that detected the failure sends a link state packet with the change information to all routers in the area. All the routers recalculate all of their routes using the Dykstra (or SPF) algorithm. The time required to run the algorithm depends on a combination of the size of the area and the number of routes in the database.

OSPF Network Scalability

Your ability to scale an OSPF internetwork depends on your overall network structure and addressing scheme. As outlined in the preceding discussions concerning network topology and route summarization, adopting a hierarchical addressing environment and a structured address assignment will be the most important factors in determining the scalability of your internetwork.

Network scalability is affected by operational and technical considerations:

- Operationally, OSPF networks should be designed so that areas do not need to be split to accommodate growth. Address space should be reserved to permit the addition of new areas.
- Technically, scaling is determined by the utilization of three resources: memory, CPU, and bandwidth.

Memory

An OSPF router stores all of the link states for all of the areas that it is in. In addition, it can store summaries and externals. Careful use of summarization and stub areas can reduce memory use substantially.

CPU

An OSPF router uses CPU cycles whenever a link state change occurs. Keeping areas small and using summarization will dramatically reduce CPU use and will create a more stable environment for OSPF.

Bandwidth

OSPF sends partial updates when a link state change occurs. The updates are flooded to all routers in the area. In a quiet network, OSPF is a quiet protocol. In a network with substantial topology changes, OSPF minimizes the amount of bandwidth used.

OSPF Security

Two kinds of security are applicable to routing protocols:

- Controlling the routers that participate in an OSPF network

  OSPF contains an optional authentication field. All routers within an area must agree on the value of the authentication field. Because OSPF is a standard protocol available on many platforms, including some hosts, using the authentication field prevents the inadvertent startup of OSPF in an uncontrolled platform on your network and reduces the potential for instability.
- Controlling the routing information that routers exchange

All routers must have the same data within an OSPF area. As a result, it is not possible to use route filters in an OSPF network to provide security.

CHAPTER 3

Designing SRB Internetworks

This chapter discusses source-route bridging (SRB) and remote source-route bridging (RSRB). SRB is evaluated within two contexts: Systems Network Architecture (SNA) and NetBIOS.

The challenge for any SRB internetwork occurs when the scale exceeds what was originally intended by IBM. When SRB technology was invented in the mid-eighties, it was viewed as a local technology that would interconnect a few rings and terminate at a remote 3745. This technology encountered problems when non-IBM protocols were required to coexist with native Token Ring traffic. Source-route bridges were intended to be the primary internetworking tool for creating a corporate-wide Token Ring internetwork. These bridges were never meant to scale to the level that many customers require. This chapter addresses the challenges of this environment and aims to help network designers successfully implement SRB within a large, multiprotocol topology.

This chapter is grouped into the following primary topics:

- SRB technology and implementation overview
- Internet Protocol (IP) routing protocol selection and implementation
- SRB network design recommendations and guidelines

Note For information concerning IBM serial line connections, refer to Appendix B, "IBM Serial Link Implementation Notes."

SRB Technology Overview and Implementation Issues

The following discussions address SRB-related technology, features provided to support SRB requirements, and implementation issues that can affect large-scale, router-based SRB networks. Specific topics include:

- Typical SRB Environments
- Multiport Bridging
- Explorer Packets and Propagation
- NetBIOS Broadcast Handling
- LAN Framing
- WAN Framing
- WAN Parallelism

- WAN Frame Sizes
- SNA Host Configuration Considerations for SRB

Note If you have eight or fewer routers operating as SRBs, you can skip this chapter. You probably do not need to tune your network.

Typical SRB Environments

SRB is used in three types of user environments:

- Many end stations to few end stations (hierarchical)—In a hierarchical SNA network, end users from multiple access sites need connectivity to a host site through a limited number of front end processors (FEPs).
- Many end stations to several end stations (distributed)—Many users need to access a limited number of servers or a limited number of devices, such as an AS/400.
- Any to any (flat)—End users at any site need to access end stations at any other site.

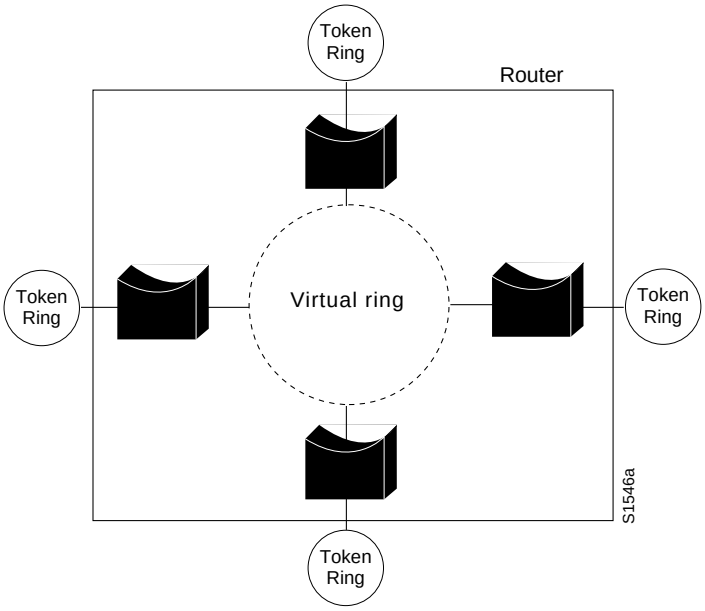
The following discussions evaluate SRB environment design issues in the context of these user environments.

Multiport Bridging

The fundamental design of an SRB as initially created by IBM was a two-port, ring-to-bridge-to-ring combination. IBM also created a half-bridge configuration that consisted of a ring-to-wide-area-network (WAN) combination followed by a second WAN-to-ring half-bridge combination.

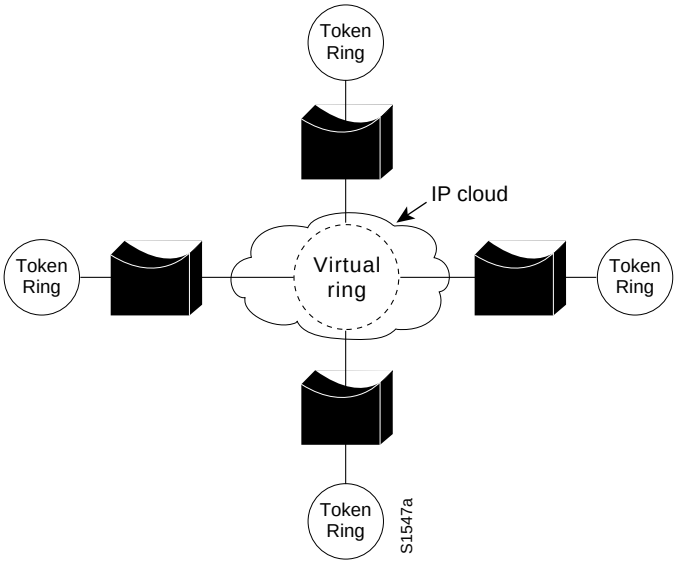
To allow more than two rings, multiport routers adopt an implementation that allows SRBs to include multiple rings on a single internetworking node. This is accomplished via the *virtual ring* capability. A virtual ring is a conceptual entity that connects two or more physical rings together either locally or remotely. Figure 3-1 illustrates the concept of multiport bridges and a virtual ring.

Figure 3-1 Multiport Bridge Using Virtual Ring Concept to Permit Multiple Ring Interconnection



The concept of virtual rings can be expanded across router boundaries. A large virtual ring can connect several access points to a central router with an FEP. Figure 3-2 illustrates this expansion.

Figure 3-2 Virtual Rings Expanded across an IP Cloud

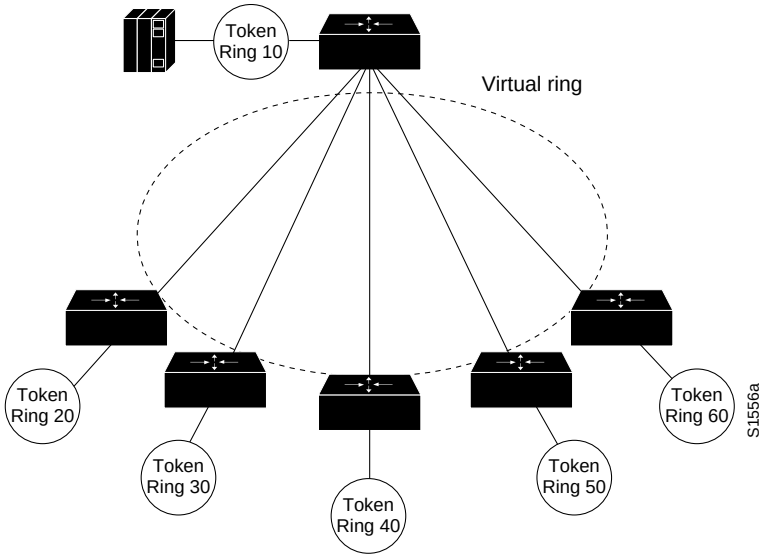


Routers support simple bridging, multiport bridging, and connections to both local and remote virtual rings. A virtual ring configuration is required to communicate with remote rings. The half-bridge configuration is not supported. The IBM half bridge does not use the concept of virtual rings; two IBM half bridges use two rings. The virtual ring advantage is in a topology that features many SRBs. In such an arrangement only a single unit is required at a central site.

Remote virtual rings have a property not found in physical ring topologies: the logical connectivity is determined by the network administrator. Two options are available: partially meshed topologies (sometimes called *redundant star topologies*) or fully meshed topologies. In a partially meshed topology, a single central location (such as an FEP Token Ring), is connected to all access locations. Each access location is logically connected to the central FEP rings and is not connected to any other ring. Partially meshed topologies using virtual rings do not permit *direct* communication between remote rings. However, communication is allowed from the central ring to the remote rings, which also allows communication among remote rings through the central ring.

In a fully meshed virtual ring topology, any ring can communicate with any other ring. Figure 3-3 and Figure 3-4 illustrate partially meshed and fully meshed topologies. In the partially meshed topology depicted in Figure 3-3, all rings are logically bridged to Token Ring 10. The access rings are not bridged together. In the fully meshed topology illustrated in Figure 3-4, all rings are bridged to all other rings.

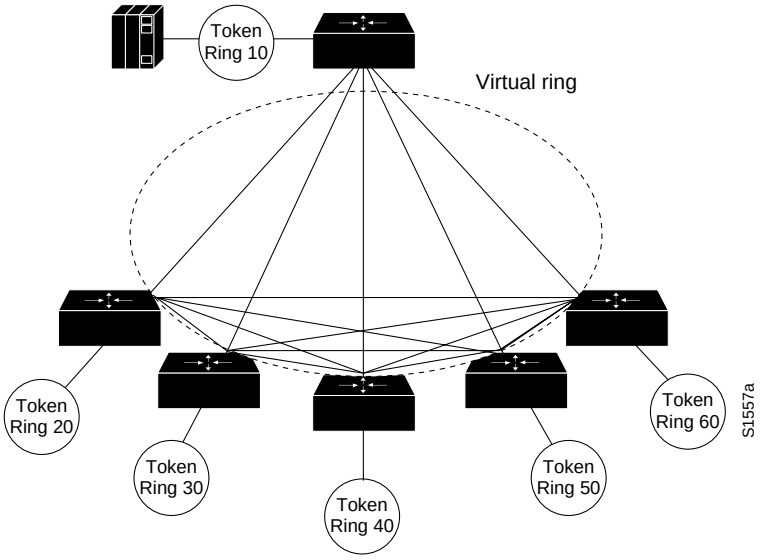
Figure 3-3 Typical Hierarchical Topology



In the topology illustrated in Figure 3-3, each of the access routers is a peer to the FEP router. They are not peers to each other. Thus, SRB is enabled between all rings and Token Ring 10 and is not enabled between Token Rings 20, 30, 40, 50 and Token Ring 60.

Assuming this is only a hierarchical SNA environment, users connected to these rings do not have SRB connectivity. Broadcasts are not forwarded across the lower layer rings (Token Rings 20 through 60); broadcasts only are sent from Token Ring 10 to or from the other rings.

Figure 3-4 Typical Fully Meshed (Flat) Topology



In the topology illustrated in Figure 3-4, each router is a peer to each other router. All rings are logically bridged to all other rings. The actual physical topology is less important than the logical topology. In Figure 3-4, the same logical topology can exist even if there are no physical connections between the access routers.

Explorer Packets and Propagation

Once you build a network of ring and bridge combinations, you must have a method for the end stations to find other end stations in the network.

An IBM bridge uses a system of *explorer packet marking* to propagate routing information through an SRB internetwork. The explorer packet is produced by the source end station and marked (updated) by each bridge that it traverses. The marked field is called the Routing Information Field (RIF). Two important transactions occur in the explorer packet handling exchange: the transmission of the explorer packet and the reply by the end station to the explorer packets that it receives.

In this environment, the source end stations must know the Token Ring Media Access Control (MAC) address of the destination end stations. Once the MAC address is understood, the source end station produces an explorer packet.

The source-route bridge updates the explorer packet to include its bridge-ring combination in the explorer packet's RIF in the MAC frame. By accumulating this information, the explorer packet gathers a hop-by-hop description of a path through the SRB network. In addition, the bridge forwards the explorer to each destination ring it encounters, therefore creating a complete topological map for each end station trying to find its way through the network.

Explorer Packet Types

There are three types of explorer packets: *local explorer packets, spanning explorer packets,* and *all-routes explorer packets.*

Note All-routes explorer packets are also known as *all-rings explorer packet*s, and spanning explorer packets are also known as *single-route* and *limited-route explorer packets*.

A local explorer packet is generated by some end systems (either NetBIOS or SNA) to find a host connected to the local ring. Once this event has occurred without finding a local host, the end station produces either a spanning explorer or an all-routes explorer packet. This behavior depends on the type of end station. SNA end stations generally produce an all-routes explorer packet. NetBIOS end stations produce a spanning explorer packet.

Note As of IOS Release 10.2, Cisco supports auto spanning tree (AST) for SRB. The implementation of AST in IOS Release10.2 is based on the IEEE 802.1 standard and is fully compatible with IBM PC bridging. New global and interface configuration commands are required to configure a router for AST. Once configured, AST can be enabled and disabled through LAN Network Manager (LNM). The following discussion of spanning tree explorer packets applies to the manual spanning tree functionality available in software releases prior to IOS 10.2.

To pass a spanning explorer packet on a router, the configuration for the router's Token Ring interface must have the **source-bridge spanning** interface configuration command for the specific ring. If this interface command is not included, spanning explorer packets are discarded.

In contrast, an all-routes explorer packet can find any valid SRB ring. No specific router configuration other than specification of SRB is required to pass all-routes explorer packets.

Explorer packet processing works as illustrated in Figure 3-5. If End station X sends an all-routes explorer packet, bridge B1 and bridge B2 both forward the explorer packet. End station Y receives two all-routes explorer packets in this configuration. End station Y responds to each of the all-routes explorer packets by sending a directed, nonbroadcast packet. In the example illustrated in Figure 3-5, four packets are generated:

- 2 all-routes explorer packets inbound (to end station Y)
- 2 nonbroadcast packets outbound (from end station Y)

Figure 3-5 Explorer Packet Processing (All-Routes Broadcast)

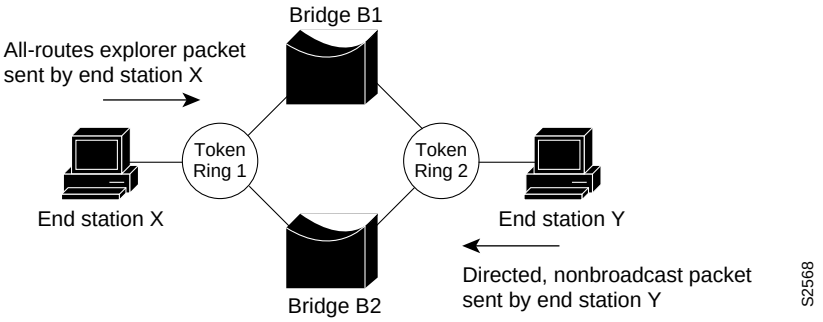
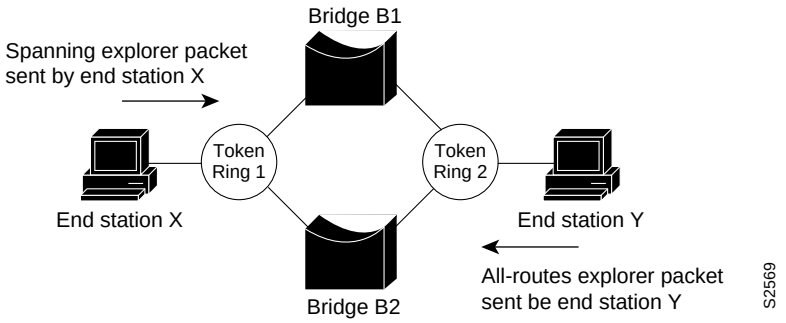


Figure 3-6 illustrates an end station sending a spanning explorer packet. Bridge B1 and bridge B2 make their respective forwarding decisions based on whether or not spanning is enabled. Assume bridge B1 has spanning enabled and bridge B2 does not have spanning enabled. Bridge B1 forwards the spanning explorer packet, and bridge B2 does not. End station Y receives one spanning explorer packet and returns an all-routes explorer packet for each single route received. As before, bridge B1 and bridge B2 forward the all-routes explorer packet. In this example, the following packets are generated:

- 1 spanning explorer packet inbound (to end station Y)
- 2 all-routes explorer packets outbound (to end station X)

Figure 3-6 Explorer Packet Processing (Spanning Explorer Broadcast)



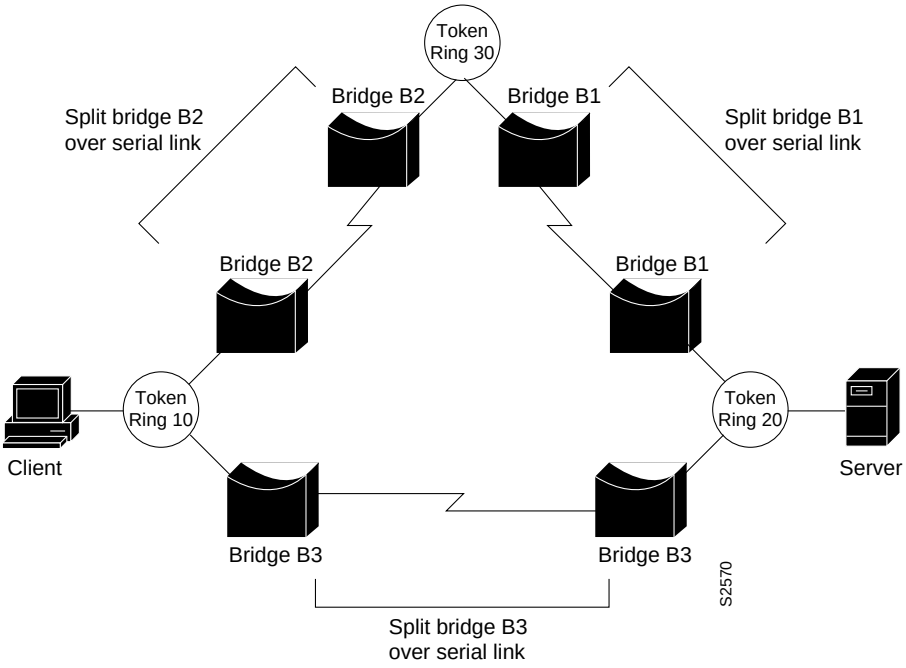
If spanning were enabled on bridge B2, it would also forward the spanning explorer packet. The following packets would be generated:

- 2 spanning explorer packets inbound (to end station Y)
- 4 all-routes explorer packets outbound (to end station X)

Note In general, there should be only a single path through the network for spanning explorer packets. If redundancy is required, a trade-off should be made between automatic redundancy and tolerance for additional explorer packet traffic. Where redundancy is required, AST should be used.

Redundancy can be achieved in many instances within the router-based cloud as a result of encapsulation in either TCP or IP, the latter called *Fast Sequenced Transport* (FST). To contrast redundancy provided by a pure SRB environment and an internetwork combining routing capabilities with SRBs, consider the networks illustrated in Figure 3-7, Figure 3-8, and Figure 3-9. Figure 3-7 illustrates a pure bridged network. Figure 3-8 and Figure 3-9 illustrate an SRB network running over routers.

Figure 3-7 Redundancy in a Pure SRB Network



In Figure 3-7, two SRB paths exist between Token Ring 10 and Token Ring 20:

- Token Ring 10 to split bridge B3 to Token Ring 20
- Token Ring 10 to split bridge B2 to Token Ring 30 to split bridge B1 to Token Ring 20

If spanning is enabled on both paths, the traffic resulting from a spanning explorer broadcast from the server is as follows:

- 2 spanning explorer packets inbound (to server)
- 4 all-routes explorer packets outbound (to client)

In router-based networks, the same type of redundancy is achieved in a different, more efficient manner as illustrated in Figure 3-8 and Figure 3-9.

Figure 3-8 Redundancy in a Router-Based SRB Network (Physical Router Connectivity)

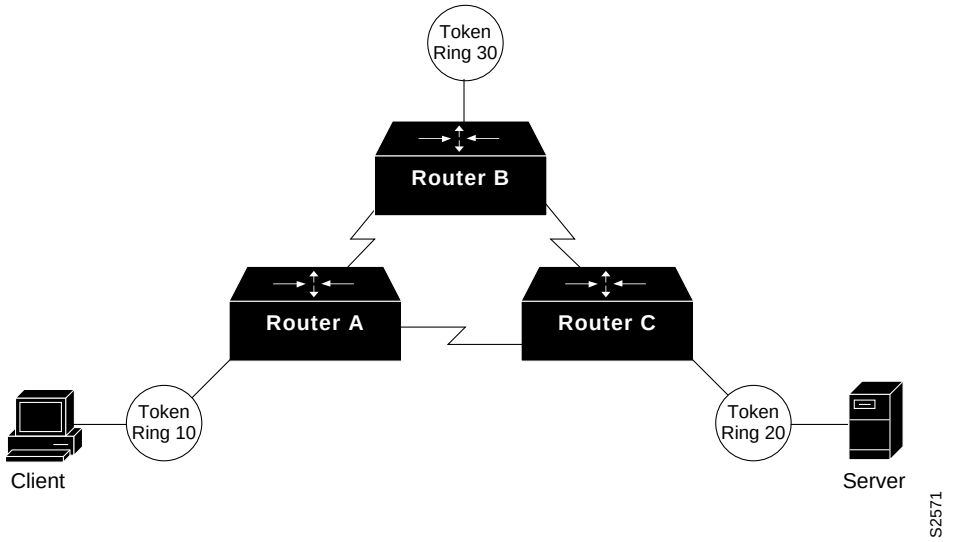
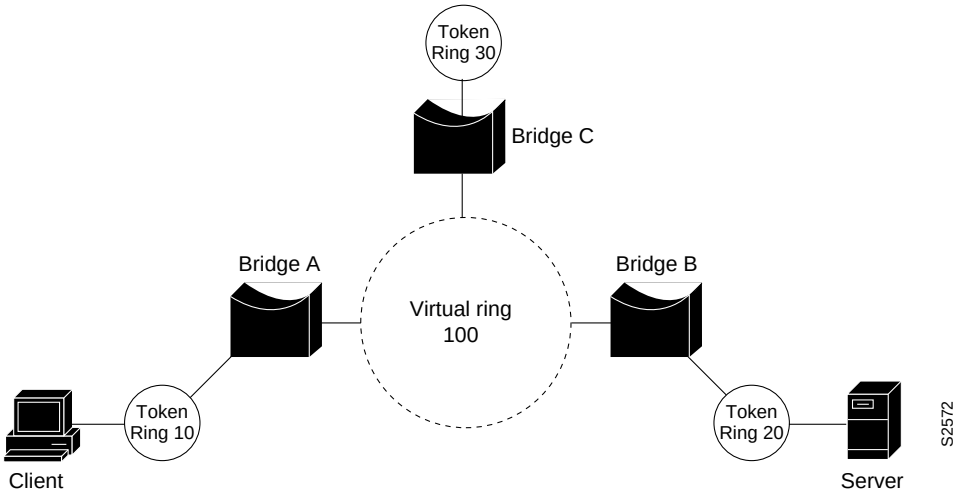


Figure 3-9 Redundancy in a Router-Based SRB Network (Logical SRB Connectivity)



With the network illustrated in Figure 3-9, there is only one SRB path between Token Ring 10 and Token Ring 20. The path is Token Ring 10 to bridge A to virtual ring 100 to bridge B to Token Ring 20. When the client sends a spanning explorer packet, the following occurs:

- 1 spanning explorer packet inbound (to server)
- 2 all-routes broadcasts outbound; one to the client on Token Ring 10 and one to Token Ring 30

These broadcast rules are valid even when spanning is enabled on all the routers. In this example, spanning does not affect the traffic. The redundancy is a result of router-to-router traffic handling.

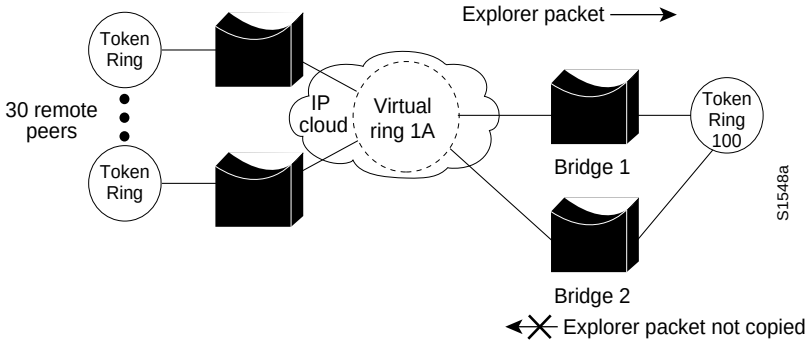
Each explorer packet is modified and copied at each destination ring when a multiring bridge is connected to more than two rings or to a virtual ring with multiple remote destinations. The virtual ring in these cases operates indistinguishably from a physical ring. The RIFs are modified exactly

as if the virtual ring were a physical ring. Because all source-route bridges are designed to forward packets, frame copying can be a limiting factor in both large-scale bridges and topologies with many Token Rings. In these topologies, your most important job as a network designer is to prevent excessive forwarding of explorer packets, which can disable an entire network.

Most source-route bridges do not propagate an explorer packet onto a ring from which it has just arrived. As a result, explorer packets are not copied from a virtual ring back to the same virtual ring even in the presence of valid remote peers.

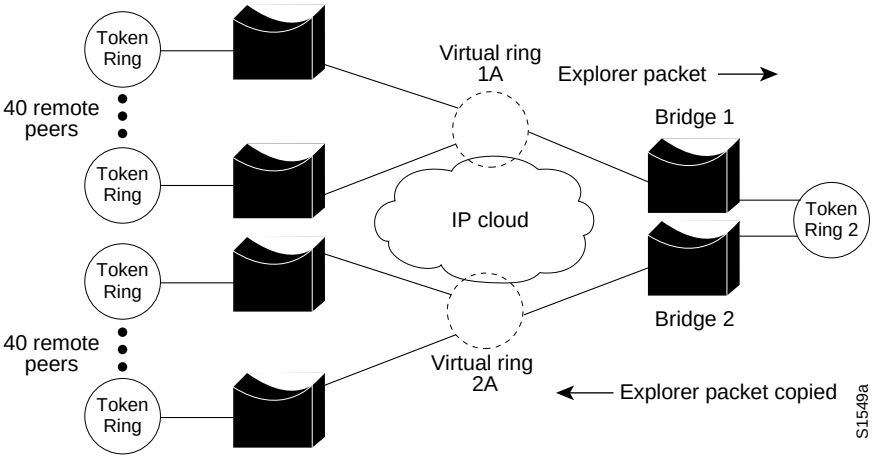
Figure 3-10 illustrates a situation where incoming explorer packets arriving on virtual ring 1A are transmitted to bridge 1, but are not copied back to virtual ring 1A even in the presence of multiple remote peer statements pointing to virtual ring 1A. This is desirable behavior. Bridge 2 does not forward frames that originated from bridge 1 because the frame has been on virtual ring 1A.

Figure 3-10 Virtual Ring and Explorer Packet Behavior



In contrast, Figure 3-11 illustrates a topology that can result in a storm of explorer packets. In this topology, two virtual rings are separated by physical Token Ring 2.

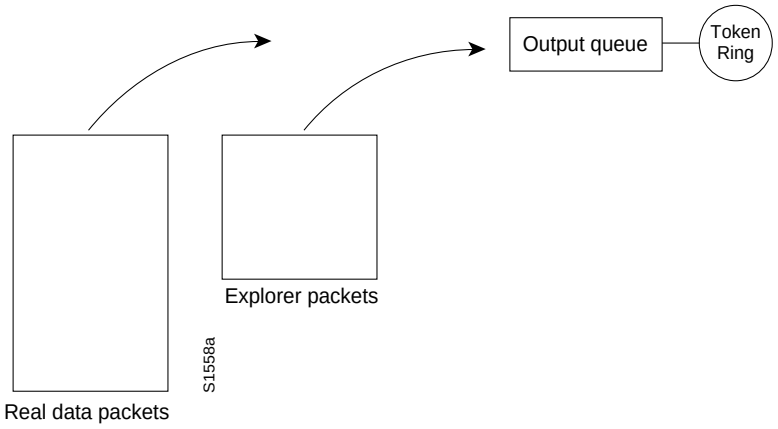
Figure 3-11 Virtual Ring Topology Resulting in Explorer Packet Storms



An incoming explorer packet arriving on virtual ring 1A is propagated to physical Token Ring 2 through bridge 1. This explorer packet is then propagated into bridge 2 and copied 40 times for each remote peer statement. Because the SRB protocol does not scale effectively, it results in this kind of explorer packet explosion that is the cause of much of the performance problems in Token Ring environments. The bridge must modify and copy the explorer packet in the CPU, causing inefficient use of the CPU and system bus for copying and modifying each explorer packet bound for a new destination.

You can reduce the number of forwarded explorer packets by enabling the explorer packet processing queue. The queue is used to divide traffic into data frames and explorer packets, as illustrated in Figure 3-12.

Figure 3-12 Queuing Process Resulting in the Division of Frames between Real Data and Explorer Packets



Reducing the number of forwarded explorer packets improves overall efficiency by allowing the CPU to spend more cycles transmitting frames for routing and bridging and less time copying, modifying, and forwarding explorer packets. To enable the explorer packet processing queue, use the following global configuration command (available with Software Release 9.1.8(5) and subsequent releases):

source-bridge explorerq-depth *number*

The value of *number* specifies the queue depth. The default value of *number* is 30 queue entries.

The disadvantage of enabling the explorer packet processing queue is the potential for suboptimal paths. For most SRB networks that are plagued by excessive explorer packet, this potential is an acceptable trade-off.

Limiting the copying of explorer packets is an important factor in designing SRB networks. Poorly designed SRB networks can collapse under high explorer packet copying loads and the resulting volume of explorer packet traffic. Although good internetwork design, such as a single unified virtual ring, can eliminate large-scale explorer packet copying, this solution does not scale infinitely. For very large internetworks, contact your technical support representative for more information about specific limitations. Also, refer to “SRB Network Design” later in this chapter for more information about how different topologies scale.

Proxy Explorer

Another way of limiting explorer packet traffic is use the *proxy explorer* feature. The function of the *proxy explorer* feature is to create an explorer packet reply cache, the entries of which are reused when subsequent explorer packets need to find the same host. The proxy explorer feature allows the SRB network designer to minimize exploding explorer packet traffic throughout the network. Routers cache the explorer packet reply and reuse it for subsequent explorer packets that are searching for the same MAC address.

Proxy explorer functionality is very useful in traditional SNA configurations because most explorer packets are destined for a single FEP on a single ring. However, if the host to be reached is an FEP on two rings (with a single locally administered address duplicated on both rings), this feature will select a single path without the possibility of redundant paths from a single router. Different routers can use different paths.

If your configuration does not involve duplicate FEPs with the same locally administered address, you can use the proxy explorer function in any SNA environment. Use the following interface configuration command:

source-bridge proxy-explorer

NetBIOS Broadcast Handling

NetBIOS stations issue broadcasts for several reasons: to verify at startup that a station's name is unique in the network, to find the route to a particular server, and to provide a heartbeat function to maintain connectivity between servers and requesters. These broadcasts are addressed either to a specific name, or to the NetBIOS functional address (such as C000 0000 0080). Station requests, such as a NAME QUERY frame, are sent as a spanning explorer broadcast to a unique NetBIOS name, and the corresponding response is returned as a broadcast of all-routes explorer packets.

NetBIOS is a broadcast-intensive protocol that can quickly consume lower bandwidth bridge paths. To address this problem, the router provides four different methods of preventing single and all-routes broadcast traffic from consuming your network:

- NetBIOS name caching
- NetBIOS datagram broadcast handling
- NetBIOS broadcast throttling
- NetBIOS broadcast damping

NetBIOS Name Caching

NetBIOS name caching allows the router to maintain a cache of NetBIOS names that it uses to avoid the high overhead of transmitting many of the broadcasts used between client and server PCs in an SRB environment.

Name caching allows the router to detect when any host sends a series of duplicate query frames and to limit the host to one frame per configurable time period. The name cache includes a cache of mappings between NetBIOS server and client names and their MAC addresses. The name cache allows the router to send broadcast requests from clients to find servers and from servers in reply to their clients directly to their destinations, rather than having to send the broadcast across the entire bridged network.

In most cases, the NetBIOS name cache is best used in situations where large amounts of broadcast traffic creates bottlenecks on the WAN media. The traffic savings of NetBIOS name caching is probably not worth the router processor overhead, however, when two local-area network (LAN) segments are interconnected.

As NetBIOS broadcasts traverse the router, the router caches the NetBIOS name contained in NAME-QUERY and NAME-RECOGNIZED broadcast frames along with the station MAC address, RIF, and the physical port from which the broadcast was received. Because the router has the NetBIOS name as well as the route to the station, it can respond locally to broadcasts and eliminate the overhead of propagating broadcast frames throughout the network.

NetBIOS name caching can be enabled on each interface by using the following interface configuration commands:

source-bridge proxy-explorer

netbios enable-name-cache

The **source-bridge proxy-explorer** command is a prerequisite for NetBIOS name caching.

To limit proxy-explorer to NetBIOS only, use the following configuration command:

source-bridge proxy-netbios-only

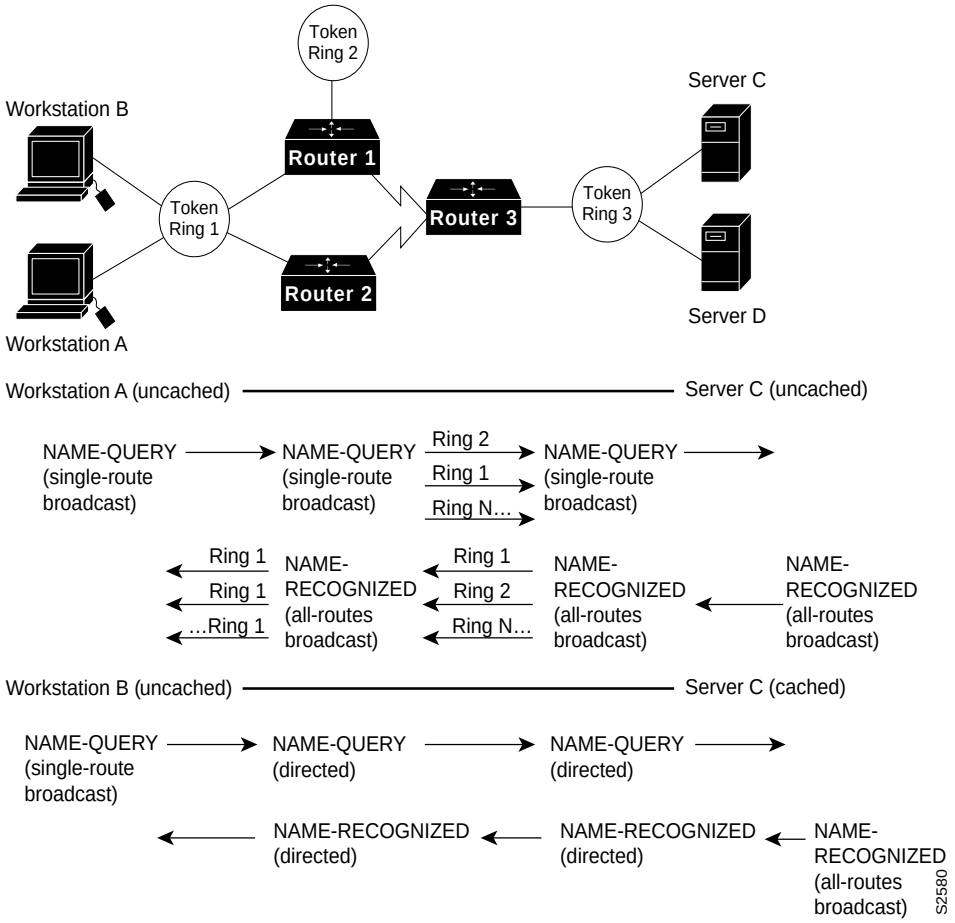
NetBIOS Name Caching Operation

Figure 3-13 illustrates the NetBIOS name caching process. Workstation A issues a NAME-QUERY frame looking for server C. The single-route broadcast is propagated to all rings in the network and server C responds with a NAME-RECOGNIZED response as a broadcast of all-routes explorer packets. The all-routes broadcast propagates throughout the network, and generates two duplicate NAME-RECOGNIZED responses to workstation A, each with different routes reflected in the MAC header. Workstation A and server C are now cached in routers 1, 2, and 3.

Workstation B now broadcasts a NAME-QUERY frame also looking for server C. The broadcast is received by router 1, which finds server C in its cache. To verify that server C and the cached route are still available, the router converts the broadcast frame to a directed frame using the cached RIF information, forwards the NAME-QUERY frame, and starts the RIF validate-age timer. When server C receives the NAME-QUERY frame, it responds with a NAME-RECOGNIZED (all-routes) broadcast. If the router receives server C's response before the validate-age timer expires, it keeps the RIF information; if not, the router deletes the RIF information from the cache.

Router 3 copies the NAME-RECOGNIZED broadcast and checks its cache for workstation B. If an entry exists, the all-routes broadcast is converted to a directed frame and is forwarded to workstation B. This example demonstrates that once a station's name has been broadcast into the network and its name is cached, no further broadcasts traverse the network. Without name caching, the broadcast activity in a network with 100 fully meshed ring segments can become a serious issue. The use of NetBIOS name caching significantly reduces the bandwidth consumed by nonproductive broadcast traffic.

Figure 3-13 NetBIOS Name Caching Process



Each NetBIOS name cache entry is aged out of the table if activity has not occurred for a given name within a configurable period of time. Aging ensures the information in the cache is current and that the cache is kept to a minimum size to maintain optimal performance.

The following global configuration command controls the name caching age timer:

netbios name-cache timeout *minutes*

The default is 15 minutes.

NetBIOS Datagram Broadcast Handling

The router also checks the NetBIOS name cache when it receives NetBIOS datagram broadcasts (addressed to unique names), which allows the router to handle NetBIOS datagram broadcasts locally in a way that is similar to NAME-QUERY and NAME-RECOGNIZED broadcast handling. The difference is that datagram broadcasts are generally one-way flows with no corresponding reply. If datagram broadcasts represent a small percentage of overall broadcast traffic, you can disable datagram handling and avoid expending additional router overhead for relatively minor effect. This decision can only be made with an understanding of your broadcast traffic patterns.

NetBIOS Broadcast Throttling

NetBIOS applications broadcast by issuing multiple successive copies of the broadcast frame into the network. For example, IBM's OS/2 LAN Requester sends six successive copies of a NAME-QUERY frame, with a pause of a half second between each repeated transmission. Some applications allow you to tune this behavior, but tuning NetBIOS broadcasts is difficult to maintain if the number of NetBIOS workstations in your network is high.

As illustrated in Figure 3-14, when NetBIOS name caching is enabled, the router forwards the first of these six broadcasts, and drops the duplicate five broadcasts. The duplicate broadcasts (which originated from the same station), continue to be dropped until the dead timer expires. Two global configuration commands control relevant timers:

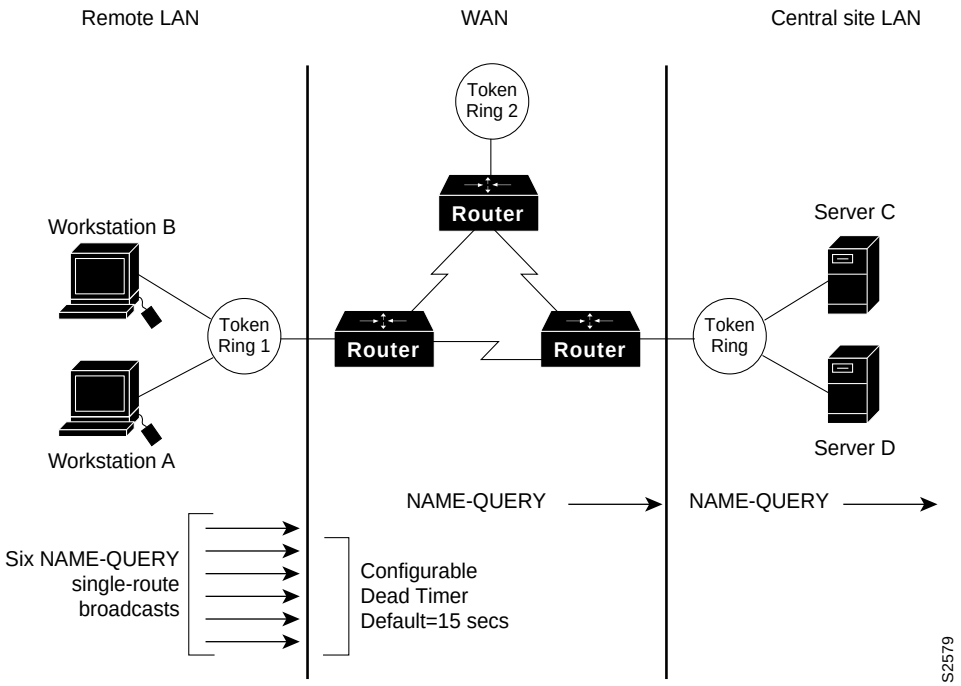
netbios name-cache query-timeout *seconds*

The default is 6 seconds.

netbios name-cache recognized-timeout *seconds*

The default is 1 second.

Figure 3-14 Throttling NetBIOS Broadcasts

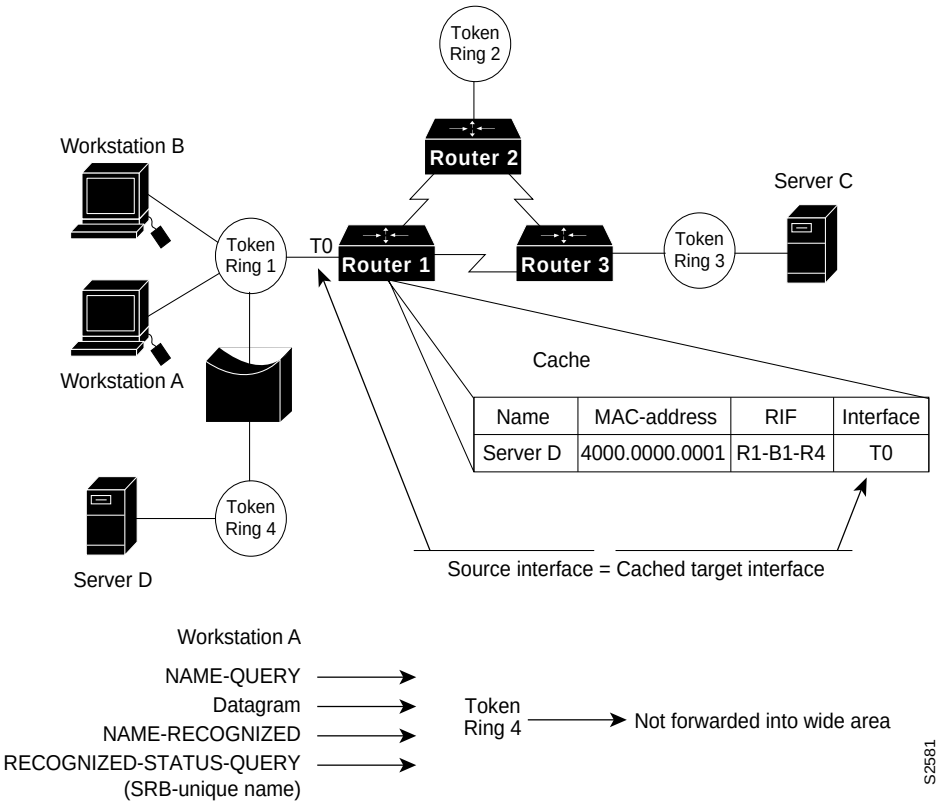


NetBIOS Broadcast Damping

The router remembers the physical port from which a NetBIOS station's route was cached. As a result, the router can remember where a cached station resides relative to the router. If the router receives a broadcast frame that is addressed to a cached NetBIOS name and if the router knows the route to that station exists off of the same interface, the router does not need to forward the broadcast to find the target station. Instead, the router drops the broadcast and prevents unnecessary broadcast traffic from traversing the network.

As illustrated in Figure 3-15, a NetBIOS broadcast addressed to server D is received by router 1 on interface T0. Router 1 finds a cached entry for server D which indicates that the route to server D is via interface T0. Because the broadcast was received on T0 and because the route to server D is via T0, the broadcast is prevented from continuing on in the network, and the requester finds server D via the local SRB topology.

Figure 3-15 NetBIOS Broadcast Damping

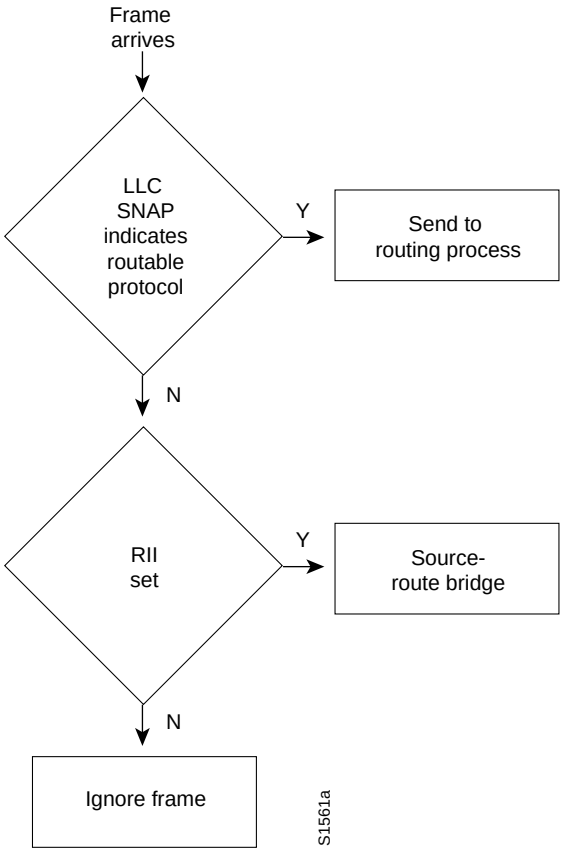


LAN Framing

Framing for SRB networks is straightforward. Using a basic IEEE 802.5 frame with Logical Link Control, type 2 (LLC2) 802.2 framing, a RIF field follows the source MAC field in the IEEE 802.5 frame. The presence of a RIF field is indicated by setting the Routing Information Identifier (RII), which is the high-order bit of the source MAC field.

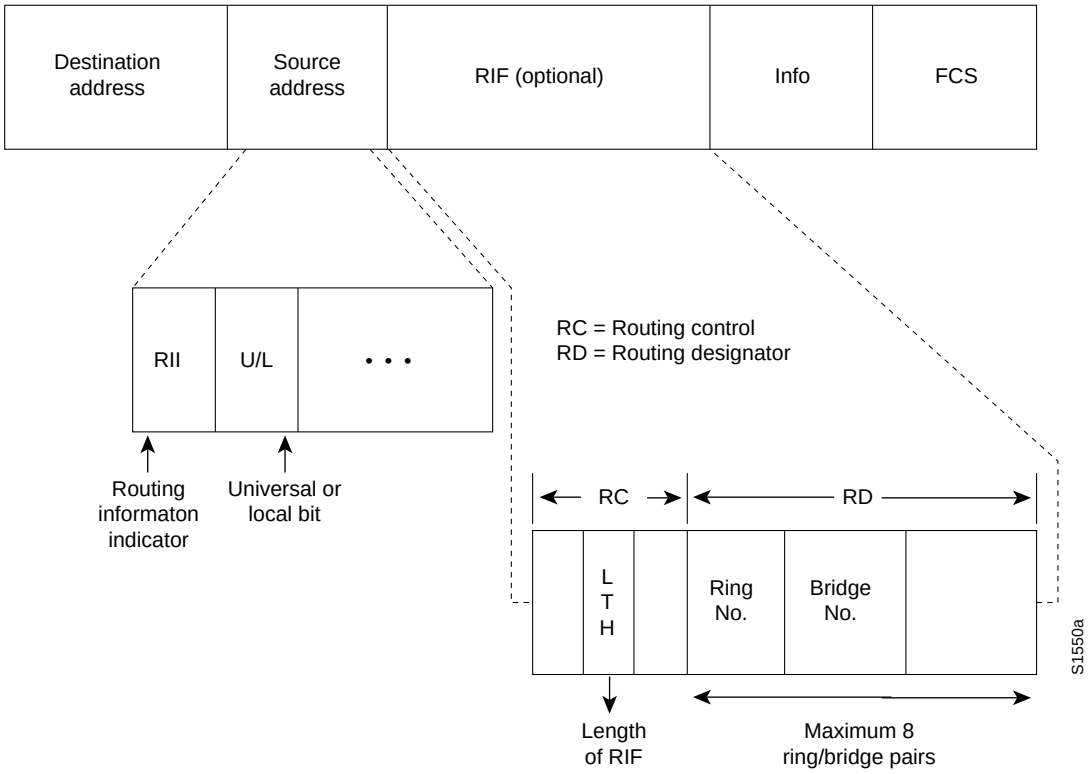
A router with SRB configured evaluates incoming frames based on IEEE 802.5 values, which is mostly a Subnetwork Access Protocol (SNAP) evaluation. Once the router determines whether the packet is to be routed, it evaluates whether to use SRB based on the value of the RII bit. If the bit is set and the router is not configured to route a specific protocol, the router sends the frame using SRB. Figure 3-16 illustrates this decision process.

Figure 3-16 Decision Process for Identifying Routable versus SRB Packets



A RIF frame has certain peculiarities that can be a challenge to decode. For example, one of the bits in the RIF indicates whether it is read front-to-back or back-to-front. Figure 3-17 outlines basic RIF content.

Figure 3-17 RIF Format



When mixing SRB networks with multiprotocol routers, routers provide a feature called *multiring* that provides several benefits. The first benefit is realized when connecting a multiprotocol router to an existing pure SRB network to support routable protocols (such as Novell's IPX). In this case, the multiring feature allows you to connect IPX and SRB networks seamlessly by routing IPX even in the presence of SRB framing. IPX stations can be linked via SRB networks or locally connected to a Token Ring with SRB framing. A router will route to the IPX station by first exploring for the station and then framing each Token Ring frame with RII and a RIF.

The second benefit of multiring is that all outgoing packets for a specific routable protocol are framed in an SRB frame. The router creates a valid SRB frame by transmitting an explorer packet to create a valid RIF entry for the SRB frame of a routable network packet.

The third benefit of multiring is that it allows a smooth transition from a previously framed SRB network to a routed network. For instance, a locally connected Token Ring can either use an IPX frame or an SRB frame depending on what is currently in use. To leverage existing IPX servers with SRB drivers, you just need to configure multiring for that specific Token Ring.

A typical **multiring** interface configuration example might be as follows:

```
interface tokenring 0
source-bridge 10 1 100
multiring ipx spanning
```

WAN Framing

Routers recognize two forms of SRB. The first is *local* SRB, which is characterized by either the standard single ring-to-bridge-to-ring combination, or a flexible form using a multiple ring-to-bridge-to-virtual ring arrangement. The second form of SRB involves WAN connections and is called *remote* SRB (RSRB).

The framing that occurs to support WAN activities is twofold. First, the SRB frame is encapsulated in one of three ways: Transmission Control Protocol/Internet Protocol (TCP/IP) encapsulation, Fast Sequence Transport (FST) encapsulation, or direct High-Level Data Link Control (HDLC) encapsulation. Next, the frame is placed in the WAN frame for the appropriate WAN media, such as HDLC, Frame Relay, or Switched Multimegabit Data Service (SMDS).

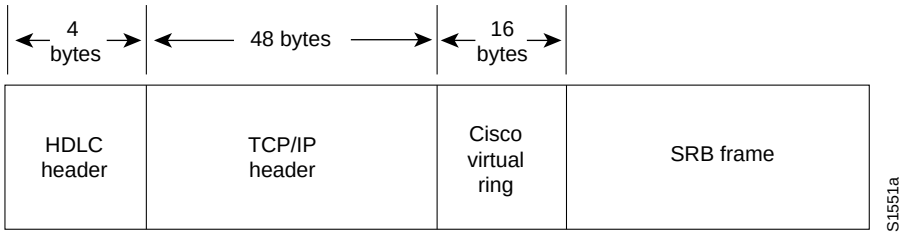
If you select direct encapsulation for a WAN serial like, you avoid the overhead of encapsulating into either IP or TCP. The datagram is framed directly into HDLC. Direct encapsulation for WAN frames only works for HDLC. Over a multiaccess media, such as Ethernet or Fiber Distributed Data Interface (FDDI), direct encapsulation can be used to transmit data from one router to another.

Selection of encapsulation is critical to the performance of the underlying network and affects the degree to which the topology can scale to very large networks of Token Rings. Each encapsulation form is addressed in the following sections.

TCP/IP Encapsulation

TCP/IP encapsulation is the most common encapsulation format. Figure 3-18 illustrates a TCP/IP-encapsulated SRB frame. The chief benefit of TCP/IP encapsulation is a robust set of capabilities to ensure reliable transport.

Figure 3-18 SRB Frame Encapsulated in TCP/IP with HDLC Header



Because many tasks are involved in TCP/IP encapsulation, such as packet reordering, running timers for retransmission, and sending acknowledgments, TCP/IP encapsulation imposes a high cost in terms of CPU overhead. For both LANs and WANs, TCP/IP encapsulation incurs additional CPU overhead because all framing occurs in the CPU and the resulting IP frame is then process-switched, which incurs additional overhead. (Process switching and its associated costs are discussed in "Process Switching," later in this chapter.)

Because of the high overhead associated with TCP/IP encapsulation, there is a significant upper boundary to maximum traffic forwarding. Performance is not the only constraint for using TCP/IP; fewer connections to other SRB rings can be supported using TCP/IP than any other encapsulation because of processor overhead required to maintain the TCP structure. In general, you should limit the maximum number of remote peers connected to a single Cisco CSC/4 or RP card using TCP/IP encapsulation. Issues that can affect the acceptable number of remote peers include link speed, traffic load, number of supported protocols, routing platform implemented, and the level of other non-SRB activity occurring in the router.

Consult with your router technical support representative when implementing TCP/IP encapsulation in environments that might be sensitive to performance or involve large numbers of remote peers.

Fast Sequenced Transport (FST) Encapsulation

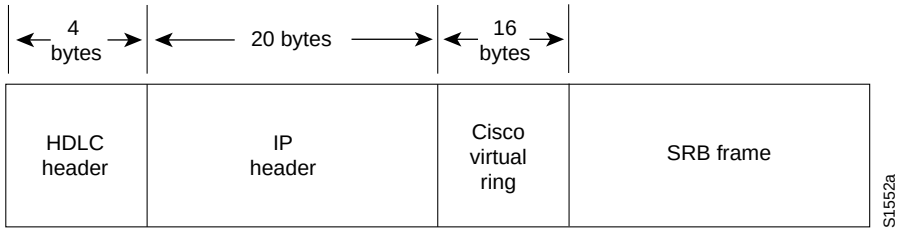
Fast Sequenced Transport (FST) encapsulation is an alternative to TCP/IP encapsulation. FST encapsulation creates an IP frame with a sequence number; this frame is transmitted to an IP destination. At arrival, FST encapsulation strips the IP frame. If the sequence number of the arriving frame is greater than the sequence number of the last frame that arrived, FST encapsulation places the frame on the destination ring. If the sequence number of the arriving frame is less than the last frame transmitted by FST encapsulation, the frame is discarded, and the router relies on the transport mechanism of LLC2 to request the discarded or out-of-order frames to be retransmitted.

FST encapsulation is configured on a per-remote-ring basis. A typical example of using the **fst** keyword with the **source-bridge remote-peer** global configuration command follows:

```
source-bridge remote-peer 10 fst 131.108.3.2
```

The benefit of FST encapsulation is sustained end-to-end performance across multiple hops. FST is fast because the IP encapsulation happens on the interface card (AGS+, Cisco 7000, MGS, and CGS) or the system memory (IGS, Cisco 2000, Cisco 2500, Cisco 3000, and Cisco 4000) while the processor is in interrupt mode. For WAN transmissions, once the framing occurs, you can select an IP switching mechanism, either process switching or fast switching depending on the desired result. Figure 3-19 illustrates the frame format of FST encapsulation.

Figure 3-19 SRB Frame Encapsulated in FST with HDLC Header



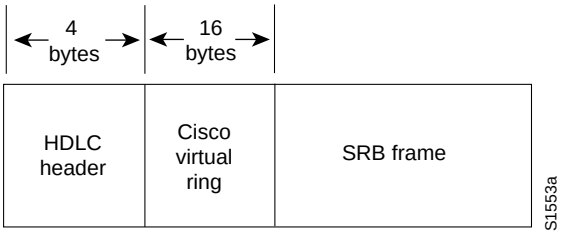
There is a cost to implementing FST encapsulation. Because the packet discard feature associated with FST encapsulation does not guarantee delivery, FST encapsulation cannot be used in conjunction with the router's local acknowledgment feature.

Direct HDLC Encapsulation

Direct HDLC encapsulation is the fastest SRB encapsulation, but has the most restrictions. Direct HDLC encapsulation allows the network designer to configure two Token Rings separated by a single Ethernet, FDDI ring, Token Ring, or serial link.

For multiaccess media such as Ethernet or FDDI, you must know the destination MAC address of the neighbor. For HDLC on a WAN link, you only need to know the serial interface over which you intend to transmit traffic. As with FST, direct HDLC encapsulation occurs at processor interrupt level and is very fast. Figure 3-20 illustrates the format.

Figure 3-20 SRB Frame Encapsulated in Direct HDLC



The following is an example of a global configuration command that configures direct HDLC encapsulation on a serial interface:

```
source-bridge remote-peer 10 interface Serial0
```

The following is an example of a global configuration command that configures direct HDLC encapsulation on an FDDI interface:

```
source-bridge remote-peer 10 interface Fddi0 00c0.3456.768a
```

When connected to parallel WAN links, direct HDLC encapsulation can operate over only one of the links. Contact your router technical support representative for specific information regarding likely performance characteristics given your specific network configuration and selection of encapsulation type.

WAN Parallelism

Parallelism implies multiple paths exist between two points that are in parallel to each other. These paths might be of equal or unequal cost. Parallel links present a number of potential problems to network designers. Parallelism is not specifically a WAN issue. However, because WAN links are expensive, parallelism becomes an important design factor. For that reason, this chapter explores some of the considerations for implementing parallel links.

Problems with parallel links in an SRB environment result from the tandem objectives of minimizing session loss when links fail and of maximizing traffic across a WAN infrastructure. Pure SRB networks maximize the WAN infrastructure but cause session losses at each link failure. IP-routed SRB networks minimize session loss, but leave the challenge of maximizing WAN links to network designers. The goal of this section is to explore the issues that affect your efforts to balance these objectives.

Setting up parallel links between either two routers (Figure 3-21) or several routers (Figure 3-22) can pose challenges in an SRB environment. First, consider environments running NetBIOS and SNA over SRB environments. When an SNA or NetBIOS frame is delivered out of sequence, the end station might declare a protocol violation and terminate the session. Session loss is probably the worst possible outcome from the point of view of a user or a network administrator.

Figure 3-21 Parallel Paths between Two WAN Routers

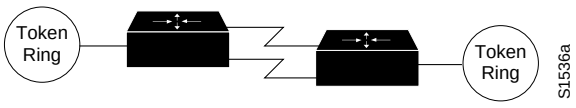
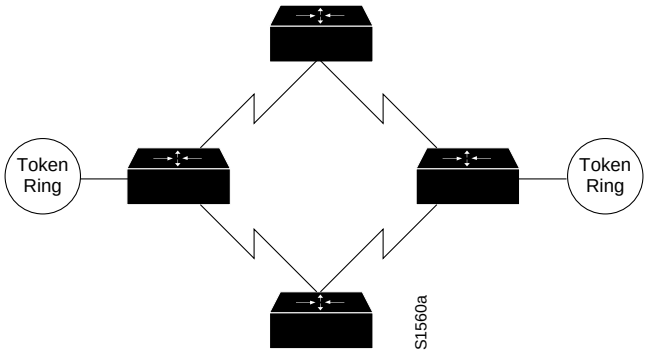


Figure 3-22 Parallel WAN Connections among Several Routers



Delivering frames in sequence is the key objective of any SRB delivery mechanism. In general, when you create parallel WAN links, you expect parallel delivery. In an SRB universe this might not be achievable because timing differences on WAN links alone can cause packet resequencing. If the router uses parallel links and starts one frame header ahead of a second frame header, there is no guarantee that the frames will arrive with the same sequencing. The second frame might arrive before the first frame because of WAN link delays. This is particularly true of packet-switched WANs.

When selecting or applying an encapsulation strategy with parallel WAN links, other factors influence which encapsulation can be used. These factors include the WAN switching technology and the IP routing protocols implemented. Each is addressed in discussions that follow. Some choices are predetermined. For example, direct HDLC encapsulation voids all parallel connections across a single virtual ring. In a multiprotocol environment, you can place SRB traffic on a single parallel link, while other protocols are load balanced on parallel links. As an alternative, you can configure the second link exclusively for multiprotocol (non-SRB) traffic.

WAN technologies can use two primary switching types: *process switching* and *fast switching*. Process switching provides full route evaluation and per-packet load balancing across parallel WAN links. Fast switching associates an IP host destination to a single interface to avoid out of order frames. The fact that the destination of a remote peer is a single IP destination can impact SRB decisions.

Process- and fast-switching techniques provide different features and different performance characteristics. Each technique must be applied in situations that can optimize their respective capabilities. These switching strategies are addressed in detail in the following sections, “Process Switching” and “Fast Switching.” Later in this chapter, “IP Routing Protocols with Parallel Links” addresses routing and switching together in the context of SRB framing options.

Process Switching

Process switching is the most expensive switching operation that the CPU can perform. Process switching involves transmitting frames in their entirety to the router CPU. Frames are then repackaged for delivery to or from a WAN interface, and the router makes a route selection for each packet. TCP/IP framing must be process switched because switching must occur when the rings are encapsulating or unencapsulating data, which occurs at processor level. FST framing can be process switched or fast switched.

Process switching begins when a frame arrives on a Token Ring interface and causes an interrupt to be transmitted to the CPU. The CPU then determines that the frame must be process switched and schedules the switch in noninterrupt mode. The frame is then transferred to the CPU and placed on

an input queue, whose depth is viewable with the **show interfaces** EXEC command. Once the entire frame has been transferred across the system bus, the frame is reworked for appropriate TCP headers and header compression.

Next, the IP route for the destination is examined. If multiple paths exist, the frame pointer is updated to use the next path for the next frame that arrives. After a route is selected, the frame is transmitted across the system bus to the output interface queue of the specific interface card from which the frame will exit. The queue entry is placed on the specific exit interface and the SCI card dequeues and transmits the frame down the WAN link.

Consult with your technical support representative for details regarding process switching performance specifics for your internetwork's particular design.

Fast Switching

Fast switching maximizes the volume of traffic that the router can handle by streamlining the router's queuing mechanisms. Fast switching deals with incoming frames in *processor interrupt mode* and minimizes the number of decisions that must be applied.

Fast switching precaches routes. Once an IP destination has been process switched, its route is cached and associated with a specific interface. When an IP destination is precached, it is tied to a specific path. For either FST or TCP/IP encapsulations, a single IP destination carries all of the SRB traffic to an FEP destination. With multiple IP paths, if fast switching is used, a single path exists for each ring destination. You must use process switching to load balance traffic across multiple paths.

Two of the SRB framing techniques are capable of being fast switched: direct HDLC encapsulation and FST encapsulation. Direct HDLC encapsulation is by definition fast switched; it cannot be process switched. FST can be fast switched or process switched.

Two IBM SRB WAN options do not allow fast switching of a frame: TCP header compression and priority or custom output queuing. If either of these features is invoked, the frame cannot be fast switched. The reason for these caveats is that certain frame components are modified when using fast switching in AGS+, MGS, CGS, or Cisco 7000 interface memory and not in the CPU memory. If extensive modification of the frame is needed, it must be done in CPU system buffers and not in buffers associated with individual interface cards.

In addition, fast switching uses only interface buffers that are not generally reported using monitoring EXEC commands such as **show interfaces**. The buffers reported in the **show interfaces** EXEC command are CPU buffers for input and output that are only using during process switching. Fast switching uses preconfigured interface buffers. You can view the allocation of buffers using the **show controllers** EXEC command.

When using SRB for a 4-port SCI card, the router normally allocates eight 2000 byte buffers for input and output combined. The Cisco 3000, Cisco 4000, Cisco 4500, and IGS/TR can transmit a *huge* buffer—generally about 15 KB. The **show buffers** EXEC command displays the exact value of a huge buffer.

For direct HDLC encapsulation, SRB frames are directly linked to an output serial port (such as interface serial 0). When an SRB frame enters the 2R or CTR card, an interrupt is transmitted to the CPU. The CPU verifies that the frame is an SRB frame and the buffer on either the 2R card or the ciscoBus controller is modified to create an HDLC header. The new frame is transmitted two bytes at a time through the CPU from either the 2R card or the ciscoBus controller across the system bus to the SCI card or SIP card.

A similar process occurs for FST encapsulation; however, SRB frames are directly linked to a destination IP address. When an SRB frame enters the 2R or CTR card, an interrupt is transmitted to the CPU. The CPU verifies that the frame is an SRB frame and the buffer on either the 2R card

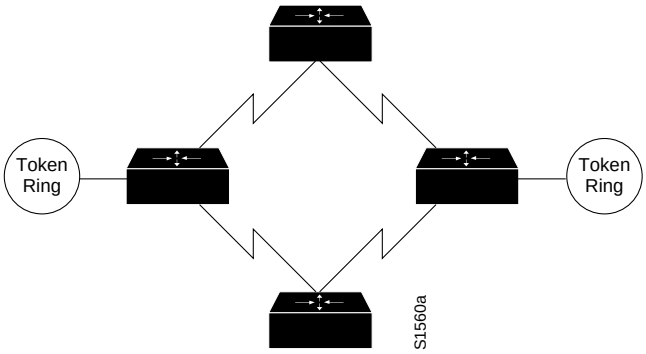
or the ciscoBus controller is modified to create an IP datagram with appropriate WAN framing for the destination. The new frame is transmitted two bytes at a time through the CPU from either the 2R card or the ciscoBus controller across the system bus to the SCI card or SIP card

To determine whether an IP destination is being fast switched, use the EXEC command **show ip route cache** to look in the IP route cache table. This command lists IP destinations as well as the relevant MAC frame and destination interface that the specific IP address will use. If the entry is fast switched, the destination IP address will be present. If the destination IP address is not present, the router is using process switching to reach the destination. By default, HDLC WAN links are fast switched for IP. If the router is configured for direct HDLC encapsulation, the only status indication is the output for the **show source-bridge** EXEC command. The bridge will indicate it is using a direct serial interface and not an IP address.

IP Routing Protocols with Parallel Links

IP routing protocols play a part in the parallel SRB WAN decisions because they can create wider parallelism than two routers with parallel links. Figure 3-23 illustrates an applicable case. With the parallel links characterized in this figure, load balancing makes three critical assumptions: equal-cost paths; routing protocol support for equal-cost load balancing; and process switching for a single IP destination.

Figure 3-23 Parallel WAN Paths



Process switching is discussed extensively in the section, “Process Switching,” earlier in this chapter. Issues relating to equal-cost path IP routing and unequal-cost path routing are discussed in the sections that follow, “IP Routing over Equal-Cost Paths” and “IP Routing over Unequal-Cost Paths Using Variance.”

IP Routing over Equal-Cost Paths

An *equal-cost path* is one that is *metrically* equivalent with some other parallel path between two points. In RIP, equivalence is parallel WAN paths of equal hops. In Interior Gateway Routing Protocol (IGRP) and Open Shortest Path First (OSPF), metric equivalence translates into WAN paths of equal bandwidth, where the bandwidth is declared by the network administrator. IGRP also adds the concept of delay to determine metrically equivalent links. To create parallel links for equal-cost paths and to actively use these paths, the router must use process switching, because all frames sent from one ring to another have the same IP destination.

The following list outlines the ability of supported IP routing technologies to create equal-cost paths:

- Static routes—For Cisco software releases predating 9.1, static routes cannot be created in parallel; only a single path can be selected. As of Software Release 9.1, static routes can be created in parallel.
- IGRP and Enhanced Interior Gateway Routing Protocol (Enhanced IGRP)—Can use up to four equal-cost paths in parallel. Ensure that the bandwidth command is correctly configured on all links.
- OSPF—If paths are of equal declared metrics, OSPF can use up to four equal-cost paths in parallel.
- RIP—RIP can use four equal-cost paths in parallel. Remember that this will not take into account anything but hops, so even unequal bandwidth links will be evaluated as having equivalent cost.

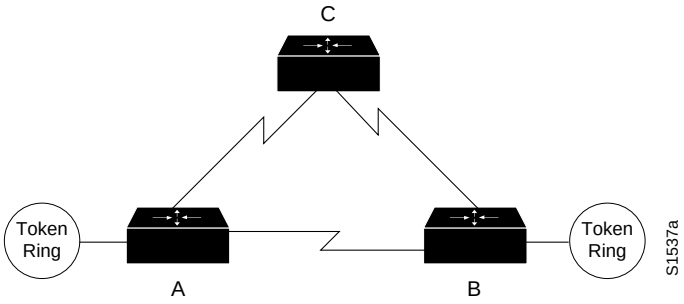
IGRP, Enhanced IGRP, and OSPF can route traffic across equal-cost paths and split SRB traffic across equal-cost links if the router is process switching. RIP will route across equal-cost paths and it will assume that all WAN links are the same speed regardless of reality. Static routes allow parallel paths and are a tool for the advanced network designer.

A router's ability to use parallel paths is determined in part by the encapsulation method used. If TCP/IP encapsulation is used, parallel paths are used. If FST encapsulation is used under normal operational conditions, all traffic must use only one of the parallel links. This is because all the RSRB traffic sent to another FST peer goes to a single IP destination address. When using fast switching, the router might alternate some traffic across parallel links based on destination address. However, because all traffic to a peer router uses only one destination IP address, all RSRB traffic flows across one link.

IP Routing over Unequal-Cost Paths Using Variance

The only routing protocols that can handle intentional unequal-cost path balancing are IGRP and Enhanced IGRP. Using a feature called *variance*, the router can load balance over unequal-cost paths. Figure 3-24 illustrates one such configuration from A to B. In this figure, load balancing the link from C to B is assumed to be faster than the link from A to B.

Figure 3-24 Unequal-Cost Load Balancing with IGRP

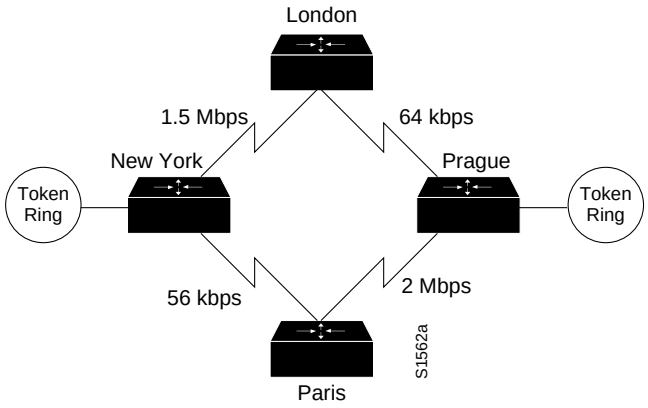


Variance has two rules that apply in this or any unequal-cost load balancing situation:

- Rule 1—Parallelism must exist in the topology.
- Rule 2—Packets must make *forward progress* on any parallel link toward an intended destination. In other words, a router will not forward traffic to another router that has the same (or greater) relative distance metric to a destination. This rule prevents loops. The rule of forward progress is straightforward. If the next-hop router is closer to the destination (than some other router) a path through it will be used as a valid alternate path.

If these rules are satisfied and the network administrator adds variance to the IGRP configuration, the router will load balance over parallel paths for a single IP destination when it is process switching. Figure 3-25 illustrates a case where variance might be used.

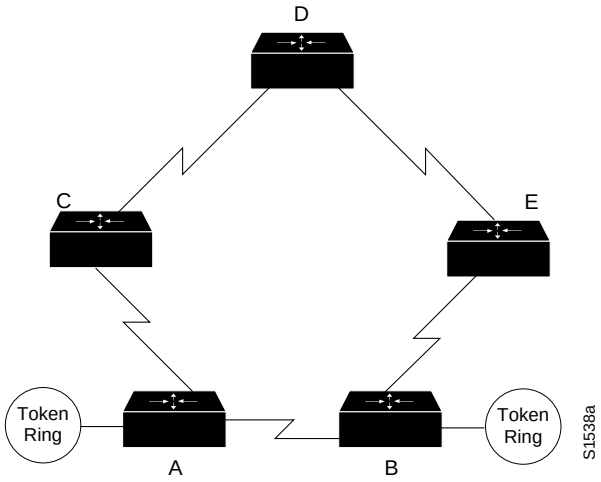
Figure 3-25 Environment Illustrating Variance Applications



Consider a set of routers connected via WAN links in a circle, where each of the WAN links is the same speed, as illustrated in Figure 3-26. Assume that a data center is at location A and that all the link speeds are the same. Consider parallelism from B to A. A parallel link exists from A to B and A to C to D to E to B; however, routing protocols are not intuitive. This topology satisfies the first rule because parallelism clearly exists; however this topology fails the forward progress rule.

The way to evaluate the forward progress rule is to examine the obvious short path separately from the long variant path, subtracting the first hop. Is C to D to E to B a better path than A to B? The answer is *no*; variance will have no effect in this topology for the problem as described.

Figure 3-26 Unequal-Cost Path and Variance Implementation Example



Now evaluate the problem from the perspective of A to E. Using the forward progress rule, compare A to B to E with C to D to E. In this topology, these paths are equal and they fail the forward progress rule. If these paths are to pass data in parallel, router A must have two paths: one to C and one to B. If C had variance configured, it would have two paths: one to A and one to D. This leaves the possibility of C routing to A and A routing to C in a loop. Thus, the variance rule is that the metric of the next-hop router must be less than the metric through the shortest path. In a five router topology with equal cost WAN links, no parallelism can be achieved.

By default, variance is not configured. If it is configured, it must be configured as an integer multiple of the allowable metric variance. Consider the following use of the **variance** router configuration command:

```
router igrp 1343
variance 2
```

Using this particular instance of the **variance** command results in a load-balanced topology with a 2:1 ratio of bandwidth. For all practical topologies, this should be an upper maximum because you should not load balance an overly high variance of WAN links such as E1 and 64 kbps.

Use variance carefully. Because IP fast switching will link an IP destination to an interface or next hop, it is possible for a single IP destination to be stuck on a 64-kbps link while most other IP destinations are wired to a fast link such as an E1. This situation will cause users to call their network administrators to determine transmission is slow one day when it was fast the day before. If SRB is fast switched, all users of a destination ring can be linked to the slower path using variance.

Variance has another major benefit: if a link fails for any reason, the router immediately switches all traffic to the parallel link without any convergence overhead. The router can do this because the parallel link is a known valid path and the router does not need to wait for the routing protocols to converge.

Local Acknowledgment Recommendations

The following recommendations apply to implementation of local acknowledgment. Use of local acknowledgment is suggested under the following conditions:

- When the WAN implementation must accommodate long network delays
- When the internetwork includes slow links, heavily used links, or poor quality links

- When the internetwork requires that sessions remain active during router convergence
- When WAN traffic must be minimized
- When the amount of LLC traffic on backbone needs to be reduced (when more than 50 percent of packets are LLC packets)
- When WAN costs must be reduced
- When network integrity must be improved, assuming TCP/IP encapsulation is used
- When unreliable WAN links exist that are causing frequent session loss
- When end station timer or retry modifications are difficult or costly
- When bandwidth constraints require the elimination of acknowledgment traffic

Parallel Link Recommendations

The following recommendations apply to parallel WAN link configuration:

- Do not combine multiple CTR cards with multiple WAN links; create a separate router with primarily WAN links. For example, do not create an 8-T1/E1 process-switched WAN solution on top of a 75-kilopackets-per-second (kpps) Token Ring engine. You will run out of CPU bandwidth.
- Use FST encapsulation whenever possible.
- Use TCP/IP encapsulation when local acknowledgment or prioritization is required.
- Maximize fast switching.

When link speeds are primarily 64 kbps, and slower and local acknowledgment or prioritization is a requirement, follow this recommendation:

- Use TCP/IP encapsulation with IGRP variance in meshed topologies when the topology can take advantage of these features.

When link speeds are primarily greater than 64 kbps and local acknowledgment is a requirement, follow this recommendation:

- Use TCP/IP encapsulation only on those links that have a history of session loss (local acknowledgment).
- Use FST encapsulation on the remaining links.

WAN Frame Sizes

The following routers have a hardware frame size limitation of 4 KB for transmitting RSRB frames: AGS+, MGS, and CGS. The 4 KB limitation results from the constraints of the SCI card to process large frames; the constraining factor is the size of a *huge* buffer. When configuring beyond 2-KB frame sizes, be aware of a decrease in buffer allocation across the interfaces. No such limitation occurs with the IGS, Cisco 2000, Cisco 2500, Cisco 3000, Cisco 4000, and Cisco 7000.

Limitations are not apparent when using TCP/IP encapsulation because the router fragments the frame into 1492-byte units that can be transmitted across the serial link. The router reassembles the frame on the other side of the WAN link. TCP/IP also assembles smaller packets into a 1492-byte frame and disassembles it on the other side of the WAN link. Assembly of smaller packets is triggered by the presence of eligible packets in the TCP input queue and is done on a per LLC2 session basis.

Use the following interface configuration command to limit maximum transmission unit (MTU) size:

mtu *bytes*

For example, the following command limits SNA frames transmitted through interface serial 0 to 1500-bytes per frame:

```
interface serial 0
mtu 1500
```

SNA Host Configuration Considerations for SRB

When designing SRB-based internets featuring routers and IBM SNA entities, you must carefully consider the configuration of SNA nodes, as well as routing nodes. Tables in Appendix C, "SNA Host Configuration for SRB Networks," provide examples of SNA host configurations that focus on three specific SNA devices:

- Front-end processors
- VTAM-switched major nodes
- 3174 cluster controllers

IP Routing Protocol Selection for SRB Networks

When designing large SRB network, the goal is to optimize the underlying IP network so it can carry SNA and NetBIOS traffic more efficiently. To do this, select your IP routing protocol carefully. You should consider the following parameters when selecting your routing protocol:

- Time to converge
- Maintainability of the internetwork routing environment

If you select a protocol using only one criterion, you might build a network that cannot be expanded and that might eventually break.

Three interior gateway routing protocols work best in an SRB environment: IGRP, Enhanced IGRP, and OSPF. In general, IGRP, Enhanced IGRP, and OSPF are the only options for building an IP SNA network. You can also consider RIP. However, because RIP does not provide any consistent WAN bandwidth sensitivity, it is a bad choice when redundancy or a meshed topology is involved.

The following discussion focuses on network topology and convergence considerations.

Convergence Considerations

Convergence is the time it takes a router to start using a new route when an active link fails in a network where alternate routes are available.

Rapid convergence is critical for SNA environments, particularly when local acknowledgment is not used. Consider a 3174 failure recovery scenario: an SNA device can lose its session with the host in 13 seconds. The result is session loss and route rediscovery for all affected units. If the 3174 had not just sent data to the host, the session would not be lost for somewhere between 13 and 42 seconds depending on what the value of the T1 Timer Inactivity parameter when the link failed. If local acknowledgment is used, the SNA session does not fail while the routers are converging.

Convergence becomes an issue when installing large meshed networks with multiple valid alternative paths. Distance vector protocols such as RIP or IGRP cannot determine the source of a learned route. A route could be learned by Router A from a neighbor that had originally learned the route from Router A. If Router A and its neighbor both use this route, they create a *routing loop*. Routing loops imply *broadcast storms* and, as a result, are widely viewed as undesirable events.

Enhanced IGRP provides superior convergence properties and operating efficiencies. It uses a convergence algorithm that eliminates routing loops throughout a route computation. More importantly, convergence time with Enhanced IGRP is reduced to a level below the threshold for session loss.

OSPF was also designed to minimize convergence time. It is good at convergence, but has side effects that will be discussed in the next section.

For routers, total convergence time has two primary components:

- Link failure detection time
- IP routing protocol convergence time

Link failure detection time is the minimum, maximum, and average time it takes the router to detect that no frames are crossing the link. IP routing protocol convergence time is the time it takes the routing protocol to detect that a failure that has occurred and to switch to alternative links.

Link Failure Effects on Convergence

Links fail in a hierarchical order of occurrence. Serial links are the most unreliable of the media. FDDI networks, Token Ring networks, and Ethernet networks are about equal in reliability and fail infrequently.

The following sections describe the significance of media-failure detection mechanisms with respect to recovery from media failure and the effects of different media failures on convergence in an IBM internetwork.

Keepalives and Convergence

Routers institute *keepalives* to verify the stability of a link. A router transmits a packet every 10 seconds by default, and when three keepalives sequentially fail to cross the link, the router declares the link to be down. To recover, the router retransmits the packet every few seconds.

For IBM IP networks, keepalives should only be active on serial and Ethernet links. Ethernet link keepalives are acceptable because the failure rate is low, but serial links (especially those faster than 64 kbps) should be set to 3 seconds. Use the **keepalive** interface configuration command to adjust the keepalive timer for a specific interface. For example:

```
interface serial 0
keepalive 3
```

This configuration reduces the maximum failure detection for the serial interface from 30 seconds to 9 seconds. (The interface is declared down after three consecutive update intervals pass with no keepalives detected.) Media-related keepalive specifics are provided in the sections that follow.

Enhanced IGRP uses small hello packets to verify link stability. Hello packets are transmitted by default every 5 seconds. When three hello packets fail to cross the link, the router immediately converges. Hello packets originate from the network layer and are protocol dependent. Use the **ip hello-interval eigrp** interface configuration command to configure a different hello packet interval for IP. For example:

```
ip hello-interval eigrp 109 3
```

This example configures a hello packet interval of 3 seconds for the IP protocol on Enhanced IGRP autonomous system number 109.

Serial Link Failure Effects

Serial links are inherently unreliable because they usually extend over long distances and because they are subject to a variety of failures.

In general, if a router detects loss of the carrier signal, it immediately disables the link. Unfortunately, carrier loss is not a guaranteed way of detecting a failed link, so the router must also use keepalives or hello packets to determine whether an interface is connected to an operational medium.

When the carrier signal is lost, the router detects the failure immediately. For any other serial failure, given the default keepalive timer of 10 seconds and the rule that three keepalives must be missed before the router declares that the interface is down, failure detection takes at least 21 seconds and could take as long as 30 seconds, with an average detection time of 25.5 seconds. When the keepalive timer is 3 seconds, the failure is detected within 7 to 9 seconds.

Token Ring Failure Effects

Token Ring media, whether twisted pair or IBM media attachment units (MAUs), rarely encounter failures. When media failures occur, the Token Ring protocol fails, causing the ring to transition, beacon, and reinitialize.

Token Ring has built-in reliability that allows the interface to determine whether the ring is up or down: the returning token indicates an active ring. Keepalives, which provide a fail-safe mechanism in case the Token Ring protocol itself fails, are also available but can be disabled in most networks to prevent unnecessary network traffic. Any keepalive failure usually indicates that the Token Ring interface is under tremendous load or may have already failed.

The failure detection time for Token Ring is immediate.

FDDI Failure Effects

Like Token Ring, FDDI rings are reliable media. The most common cause of failure for dual-attached FDDI networks is users who turn their devices off, which causes the FDDI ring to "wrap." Keepalives are available, but are not particularly useful. Enabling keepalives with FDDI can cause problems in high-load environments because the keepalives add to the traffic. Because the router disables the interface when it is experiencing intolerably heavy traffic loads, detection of a keepalive loss is usually a false error indication.

The failure detection time for FDDI rings is immediate.

Ethernet Failure Effects

Ethernet media is generally reliable but lacks a failure-detection protocol. Therefore, keepalives play a critical role in determining the availability of the media. The keepalive must fail three times for the router to disable the interface. There is no indication of the location or source of the failure, whether it is from router to MAU or across the physical media.

Given the default keepalive timer of 10 seconds and the rule that three keepalives must be missed before the router declares that the interface is down, failure detection takes at least 21 seconds and could take as long as 30 seconds, with an average detection time of 25.5 seconds.

Routing Protocol Convergence

When analyzing routing convergence, it is assumed that a link has failed or router keepalives have not been detected. The router waits for a link failure detection period to expire. After this waiting period passes, the router incurs a *routing protocol convergence time*. The following discussions address convergence for IGRP, Enhanced IGRP, and OSPF.

IGRP Convergence

IGRP convergence is controlled by a single factor: whether *holddown* has been configured. This discussion focuses on determining when it is appropriate to disable holddown and when it should be enabled (the default).

Because a router learns about routes from its neighbors, a distance vector routing protocol never actually understands the topologies to which it is connected; instead, it approximates the topologies.When enabled, holddown, which is a property of distance vector routing protocols, specifies that alternate paths are not used until the paths in question are determined to be actual alternate routes. When a failure occurs and alternate paths exists, the router holds down any routing protocol changes until the holddown timer expires to determine that the network is now completely known.

IGRP allows you to configure the protocol so that it will *not* hold down a link. The danger of administratively disabling holddown is that the routers might loop packets to each other for networks that are unreachable, which would cause the receipt of high volumes of errant traffic that could dominate low-bandwidth links. Any errant datagram would loop up to 254 times before the "counting to infinity" process causes the datagram to be dropped. The length of time associated with "counting to infinity" can be modified using the **metric maximum-hops** *hops* router configuration command. The default *hops* value is 100; the maximum is 255.

Generally, meshed WAN bandwidth that consists of fractional T1/E1 or greater can converge faster than parallel WAN bandwidth that is 64 kbps. Network topologies with high WAN bandwidth can support disabling holddown, so you can safely disable holddown on all routers in any network that have high WAN bandwidth.

If convergence time is worth trading off against potential bandwidth for sites with lower-speed links, you can disable holddown on these sites. However, if a loop occurs when a link is lost, the network performance for end systems connected to affected sites might be poor until "counting to infinity" ends. If you require faster convergence and can live with congestion for a brief period, you can disable holddown in any case.

To disable holddown, enter the following router configuration commands for all routers in the network:

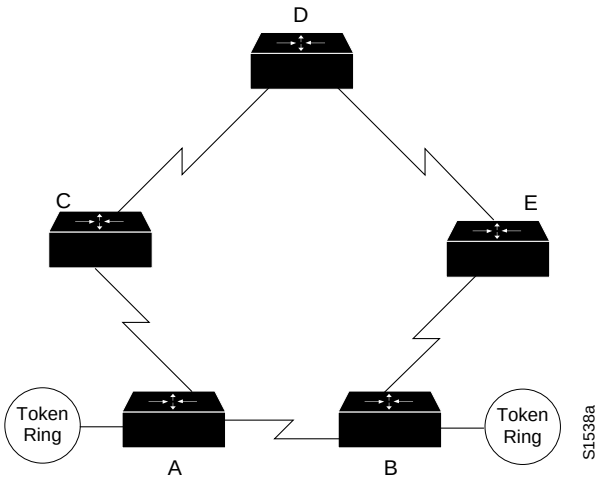
router igrp *autonomous-system*

network *network-number*

no metric holddown

Including the **no metric holddown** router configuration command changes the convergence of IP to 50 percent of neighbor update time (on average) assuming a neighbor is using this other valid route. Consider a topology as illustrated in Figure 3-27.

Figure 3-27 Convergence Topology



Assume that all links illustrated in Figure 3-27 are of equal speed and that the link from A to B fails. If C is using A to get to B, the IGRP Flash update tells C that its route to B is probably down. When D sends the next IGRP update, C uses D to get to B. A knows its route to B is down, and waits for two updates from C (on average) to get a new route to B. Most topologies converge with a single neighbor update.

If variance is active and there are two separate paths to a destination network, the network converges immediately to the remaining path when the router receives a Flash update.

Also note that the default values for IGRP timers are appropriate for general IP networks, not for IBM IP networks. It is necessary to change a few timer defaults for an IBM IP environment. The basic neighbor update timer is set to 90 seconds. For IBM IP networks, use 20 seconds, which results in an average IBM IP convergence for IGRP of 10 seconds with a Flash update. To make this change, modify the IGRP configuration of each router. The router configuration commands are as follows:

router igrp *autonomous-system*

network *network-number*

timers basic *update invalid holddown flush* [*sleeptime*]

Consider the following configuration for the **timers basic** router configuration command:

```
timers basic 20 60 60 120
```

These values optimize IGRP convergence in an IBM IP environment. If holddown is enabled, the worst-case convergence is three update periods of 20 seconds each, for a total of 60 seconds. Although these values optimize convergence, the worst-case convergence time can break IBM sessions. Try using local acknowledgment to keep sessions up while IGRP converges.

Enhanced IGRP Convergence

Enhanced IGRP is an advanced version of IGRP. The same distance vector technology found in IGRP is used in Enhanced IGRP, and the underlying distance information remains unchanged. Enhanced IGRP implements a new convergence algorithm that permits loop-free operation throughout a route computation, which improves Enhanced IGRP convergence properties and operating efficiency. Enhanced IGRP allows all routers involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in the recomputation. The result is very fast convergence time.

OSPF Convergence

OSPF uses two mechanisms for detecting failure. The first mechanism consists of interface status changes, such as carrier loss on a serial link or keepalive loss. The second mechanism is failure of OSPF to transmit and receive a hello packet within a timing window called a *dead timer.* Once the dead timer expires, the router assumes the link is dead. Once a router running OSPF assumes a link is dead, it produces an area-wide broadcast that causes all nodes to recompute their topology maps.

When OSPF receives an active multicast with link down information, the convergence time is less than 1 second. Suboptimal OSPF convergence occurs when a link is down but the router receives no forward indication. In this failure situation, the router must wait for the dead timer to expire. By default, the OSPF dead timer is set to 40 seconds. In general IP networks, you can set the dead timer equal to at least three OSPF hello packets.

In the IBM IP environment, the default values of the OSPF timers are too high for the session layer convergence that SNA and NetBIOS require, so you should change the dead timer to 18 seconds and the hello timer to 6 seconds for each interface in your network. For example:

```
interface tokenring 0
ip ospf dead-interval 18
ip ospf hello-interval 6
```

Convergence Summary

If you followed all the recommendations in this section, the behavior of the two routing protocols is as follows.

- IGRP provides instant convergence with carrier loss and active variant paths. Assuming *serial keepalive* = 3 and *update* = 20, convergence delays are as follows:
  - 7- to 9-second convergence (8-second convergence on average) with serial keepalive loss and active variant paths. This assumes that 3 keepalives have expired, no holddown, and no routing update was required.
  - 1- to 20-second convergence (10.5-second convergence on average) with carrier loss, Flash update, no holddown. This assumes failure detection is immediate; therefore time for routing update is the only factor.

  - 10- to 29-second convergence with keepalive loss, Flash update, and no holddown. This assumes a 9-second keepalive loss, the Flash update was immediate, and 1 to 20 seconds for the routing update.
  - 69-second convergence worst case scenario (9 second keepalive loss, 3 updates of 20 seconds each).
- Enhanced IGRP provides instant convergence with carrier loss and presence of a feasible successor. Convergence delays are as follows:
  - 11- to 15-second convergence by default for Hello packet loss in all cases.
- OSPF provides instant convergence with carrier loss and active broadcasts. Convergence delays are as follows:
  - 9-second convergence with serial keepalive loss and active broadcasts.
  - 18-second convergence worst-case scenario.

Note Unless you have configured OSPF with unrealistically low timer settings, the total convergence time is the sum of the time it takes the interface to change its state from up to down, combined with the time it takes the routing protocol to converge.

Routing Protocol Design and Maintenance Issues

You must consider two key design and maintenance factors when creating networks based on IGRP, Enhanced IGRP, or OSPF for primarily SNA traffic:

- Routing protocol network design
- Routing protocol scalability

Routing Protocol Network Design

Some routing protocols do not require an additional topological structure to build a successful internetwork. Other routing protocols require a separate topological structure outside of the existing addressing structure that must be maintained and well understood. IGRP, Enhanced IGRP, and OSPF show how different routing protocols deal with network design issues.

IGRP Routing Protocol Network Design

IGRP has no implicit network design requirements. IGRP networks can scale as nonhierarchical topologies to thousands of networks and hundreds of routers.

However, implementing hundreds of IGRP routers in the same autonomous system results in the transmission of an extremely large routing update every 90 seconds (by default). The impact of routing update transmission is dampened by a feature of IGRP called *route summarization,* which summarizes unconnected network numbers into a single routing table entry.

For example, if 1000 subnets of TCP/IP are distributed evenly across 10 IP networks, a single router with route summarization would see the 100 subnets of its locally connected network and nine summary routes to all other networks. Route summarization reduces the routing table of large networks, but can result in suboptimal routing at the border points.

Enhanced IGRP Routing Protocol Network Design

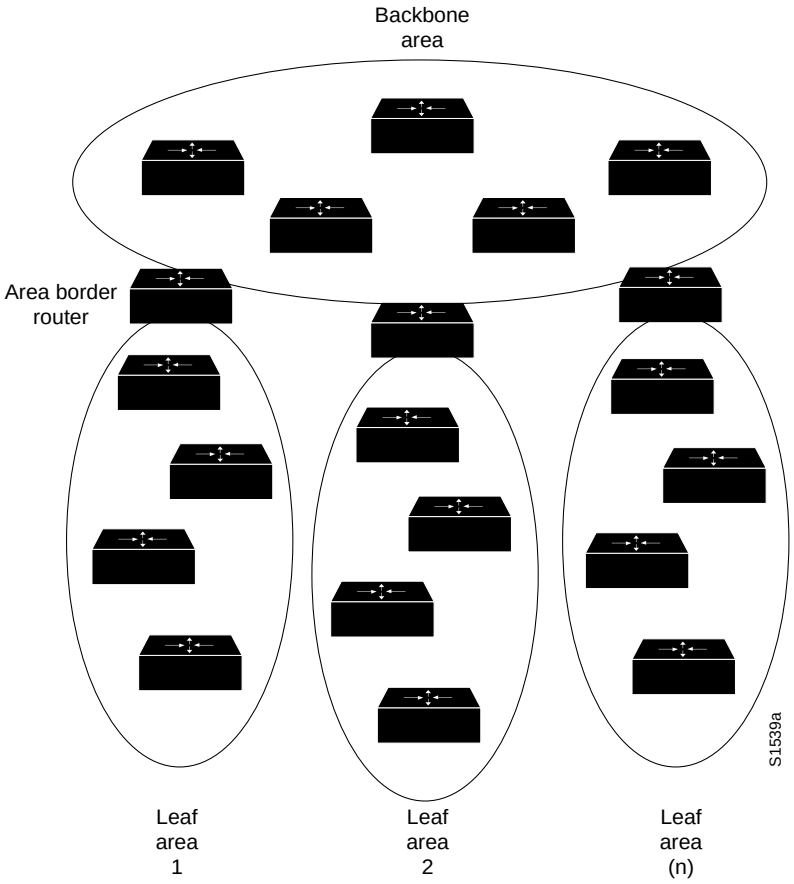
Enhanced IGRP, like IGRP, has no implicit network design requirements. Unlike IGRP, Enhanced IGRP does not make large routing table updates in a single large autonomous system, which makes the use of Enhanced IGRP even more scalable. Only routers that are directly involved in a topology change are involved in route recomputation, which saves processor overhead and results in minimal bandwidth utilization for routing updates.

Enhanced IGRP uses an automatic redistribution mechanism so IGRP routes are imported into Enhanced IGRP and vice versa, for compatibility and seamless interoperation with IGRP routers. The compatibility feature allows you to take advantage of the benefits of both protocols while migrating from IGRP to Enhanced IGRP and allows Enhanced IGRP to be enabled in strategic locations carefully without disrupting IGRP performance. By default, IGRP routes take precedence over Enhanced IGRP routes, but a configuration command that does not require routing processes to restart can change the default.

OSPF Routing Protocol Network Design

OSPF has a network structure that must be maintained separately from the addressing structure of IP. The concept in OSPF is that a single backbone of routers will communicate with several leaf areas. Consider the general environment illustrated in Figure 3-28.

Figure 3-28 OSPF Backbone Communicating with Several Leaf Areas



Communication between areas occurs through the backbone only. There is no interarea communication except through the backbone, which is not an overwhelming constraint for most SNA networks because they are already hierarchical in nature. However, NetBIOS networks are not hierarchical in nature and this design can pose a challenge.

A hierarchical structure limits the extent of link-state broadcasts that indicate link failures. OSPF builds in each router a full area topological database that describes each router and each link. When any link changes state, each router within an area recomputes the entire database and builds a new routing table. Traffic associated with this recomputation process occurs across all links in the area. With a typical large installation (for example, 400 routers), you might expect several link updates per second. However, link updates can occur more often, flooding the network and forcing the routers to spend all active cycles maintaining routing tables instead of forwarding traffic.

To avoid these problems, create a *structure* of leaf areas and a unique backbone. To create this structure, take the square root of the number of routers and subtract 1 for the backbone. For example, 100 routers would optimally be allocated with 10 routers in the backbone and 9 areas each with 10 routers. Each area must touch the backbone, so the selection of the backbone routers is critical.

Modifying an existing topology to add an additional ten routers to a geographically remote location poses a greater challenge. You must decide whether to create an unbalanced area connecting the remote location to the backbone, or to rebalance the topology by adding an OSPF backbone router at the remote location.

Once you have created the topology, you must impose IP addressing on it. If you do not assign a separate network to each leaf area, the boundaries between the leaf areas and the backbone are meaningless and link status changes will propagate through the entire network. Each backbone router that bounds an area (called an *area border router*) must summarize the routes imported to and from the backbone. Route summarization does not occur by default, so for most IP networks you must include a common set of commands at each area border router. The following is a typical configuration for area border routers:

```
router ospf 10
network 192.30.0.0 0.0.0.255 area 0
network 131.108.0.0 0.0.255.255 area 0.0.0.1
area 0.0.0.1 range 131.108.0.0 255.255.0.0
```

In this example, the importation of routes into the backbone of network 131.108.0.0 is limited. Unfortunately, it only specifies a single point of entry for network 131.108.0.0. If several area border routers are connected to leaf area 1 using network 131.108.0.0, the router uses the nearest area border router with connectivity to 131.108.0.0.

The techniques used for addressing an OSPF using multiple areas are discussed in "Addressing and Route Summarization," in Chapter 2, "Designing Large-Scale IP Internetworks."

Routing Protocol Scalability

Only one significant design challenge exists for large scalable IBM networks using IGRP as the routing protocol: low-speed links individually connected as leaf networks where IGRP transmits large routing tables. To prevent potential problems, configure the router to transmit IGRP information in a single direction—toward the backbone. The leaf router uses default routing to find the backbone and all other valid routes. The leaf router will transmit IGRP information about its routes to the backbone. The backbone router does not transmit any IGRP information to the leaf.

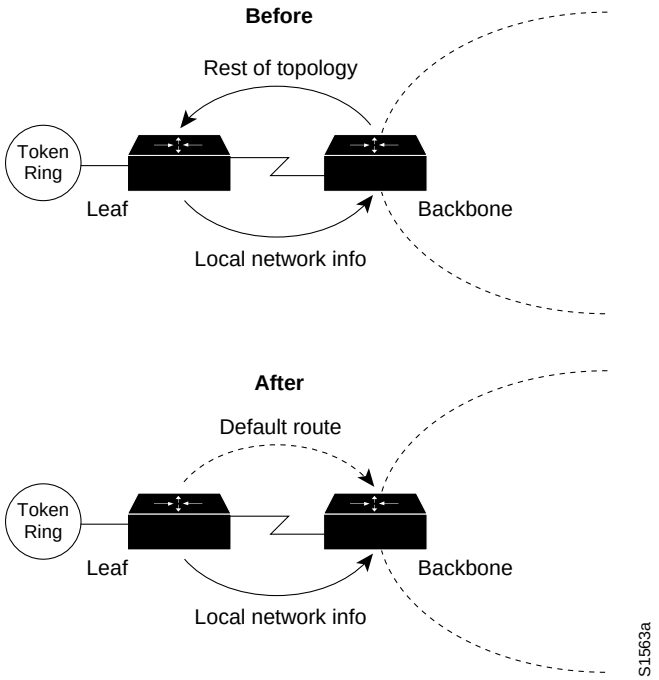
The following examples illustrate configurations for leaf and backbone routers:

```
! Leaf router configuration
router igrp 109
network 131.108.0.0
ip route 0.0.0.0 Serial0
ip route 131.108.0.0 Serial0

! Backbone router configuration
router igrp 109
network 131.108.0.0
passive-interface Serial0
```

Figure 3-29 illustrates what happens when the preceding leaf and backbone router configurations are used. This configuration does not send routing information to the lower-speed leaf routers, while the backbone retains all valid routes in the network.

Figure 3-29 Effects of Using the Passive-Interface Router Configuration Command



Note When designing large branch networks based on OSPF routing, OSPF has a natural limit. When link instability raises broadcast traffic and route recomputation to an unacceptable level, the network is at its limit. Always contact your router technical support representative when designing large OSPF-based internetworks.

SRB Network Design

The key to building predictable and scalable SRB networks is to design the network properly and within the framework of this guide. Ultimately, there is a limit to the maximum diameter of a single meshed virtual ring, so before you begin designing a network, consider four critical questions:

- How many routers are required?
- Are there any T1/T3, E1/E3, fractional T1/T3, or fractional E1/E3 links?
- Is the design for an SNA network, a NetBIOS network, or both?
- Is the network a multiprotocol environment?

Answering these questions will help you assess the available options. The first question assesses the ability to build a simple SRB network. If you are implementing a large internetwork, contact your technical support representative for specific information about virtual ring limitations.

The second question relates to the existence of SRB WAN traffic that would reduce a meshed topology to a smaller radius. If you are using T1/T3 or E1/E3 technology, you can take advantage of their increased bandwidth capabilities by increasing traffic loads to and from the rings, which allows you to reduce the number of routers.

The third question helps you determine whether a partially meshed topology can be used when an FEP-connected ring is a peer of each Token Ring in the network. The remote Token Rings are only allowed to be a peer of the FEP rings, not of each other. This topology is called a partially meshed network because certain points can connect only to certain points. Partially meshed SRB networks are much more scalable than fully meshed networks, in which all rings can reach all rings. Fully meshed topologies are often required in NetBIOS environments.

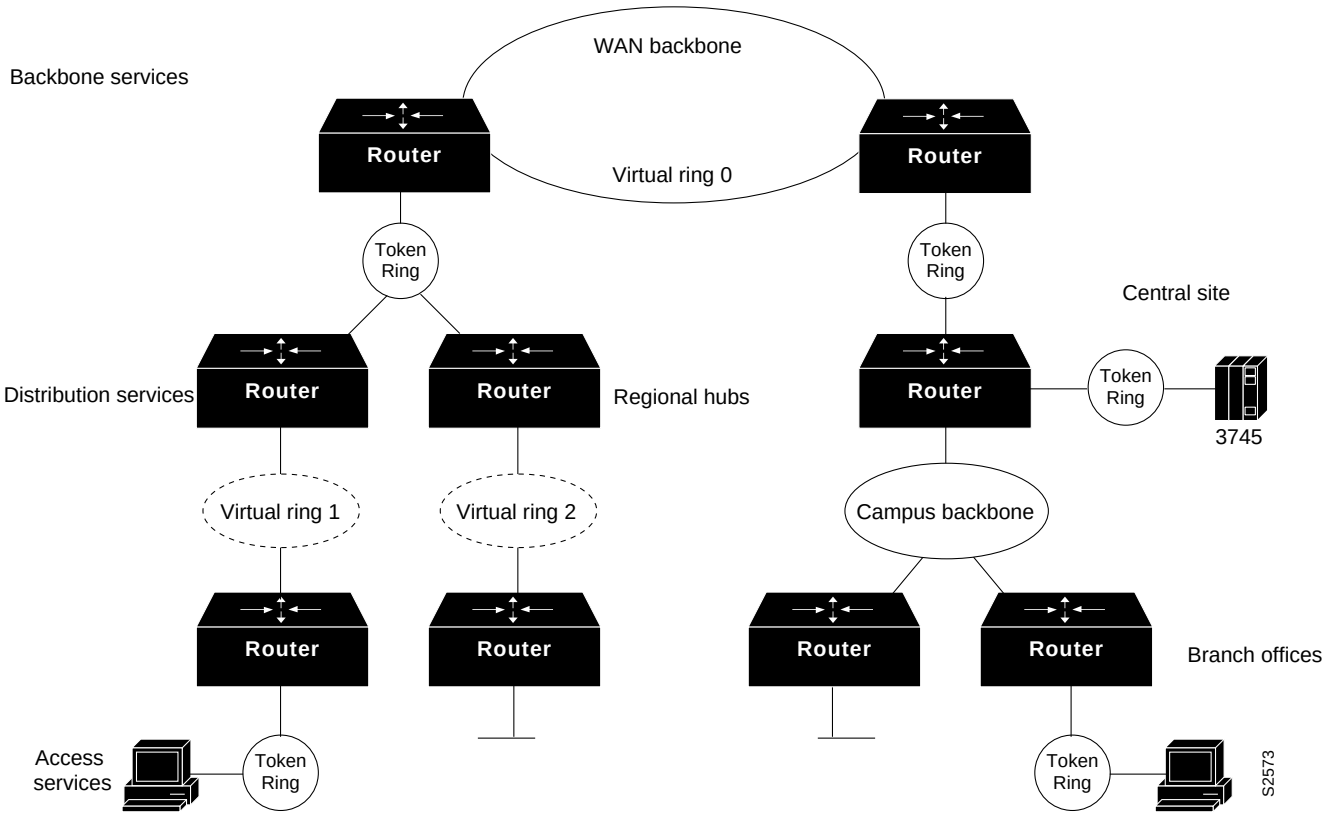
The last question implicitly raises the topic of *prioritization*. When dealing with a multiprotocol internetwork, you must consider your options for implementing some kind of traffic priority to ensure acceptable response time for interactive traffic, while maintaining adequate internetworking resources to handle other types of traffic, such as file transfers.

In general, it is best to design a router network in a hierarchical fashion; there are typically three logical service layers: the backbone (or core) service layer, the distribution service layer, and the access service layer. Figure 3-30 illustrates these basic service layers. A key consideration in designing a router network for SRB is determining how to design the virtual rings.

Two key considerations guide the design of virtual rings:

- The type of SRB connectivity required (hierarchical, distributed, or flat)
- The corporate organizational structure

Figure 3-30 Backbone, Distribution, and Access Service Layers in an SRB Environment



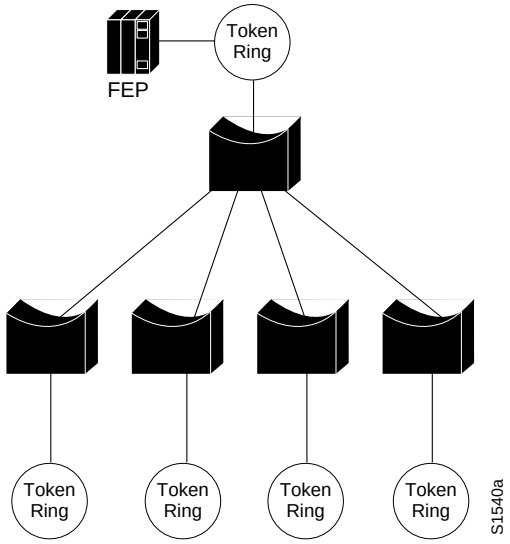
The remainder of this section focuses on network design approaches that help create scalable networks. The following topics are discussed:

- Hierarchical design for SNA environments
- Hierarchical design for NetBIOS environments
- Queuing and prioritization schemes

Hierarchical Design for SNA Environments

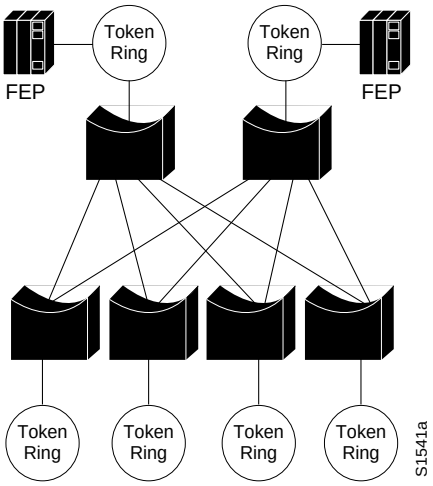
In SNA-only networks, all processing is hierarchical, where a single FEP or a few FEPs (one primary and one secondary) are the target of all remote rings. The SRB topology is focused from all remote rings to a single or a few redundant rings. A topology featuring a single FEP is illustrated in Figure 3-31.

Figure 3-31 Hierarchical Topology Featuring a Single FEP



A topology featuring duplicate FEPs on duplicate rings is illustrated in Figure 3-32.

Figure 3-32 Duplicate FEPs on Duplicate Rings



The topology in Figure 3-32 is a partially meshed topology because the remote nodes cannot reach each other; they can only reach the core of the network where the FEPs are located.

When you are designing a partially meshed topology, several options are available. SNA traffic can be generalized as having few explorer packets and having the requirement to connect many remote sites. The suggested topology for a partially meshed topology depends on whether the link speed to the core is greater the 64 kbps. Contact your technical support representative for specific limitations and capabilities regarding the maximum number of peers for the various encapsulation implementations and your specific network attributes.

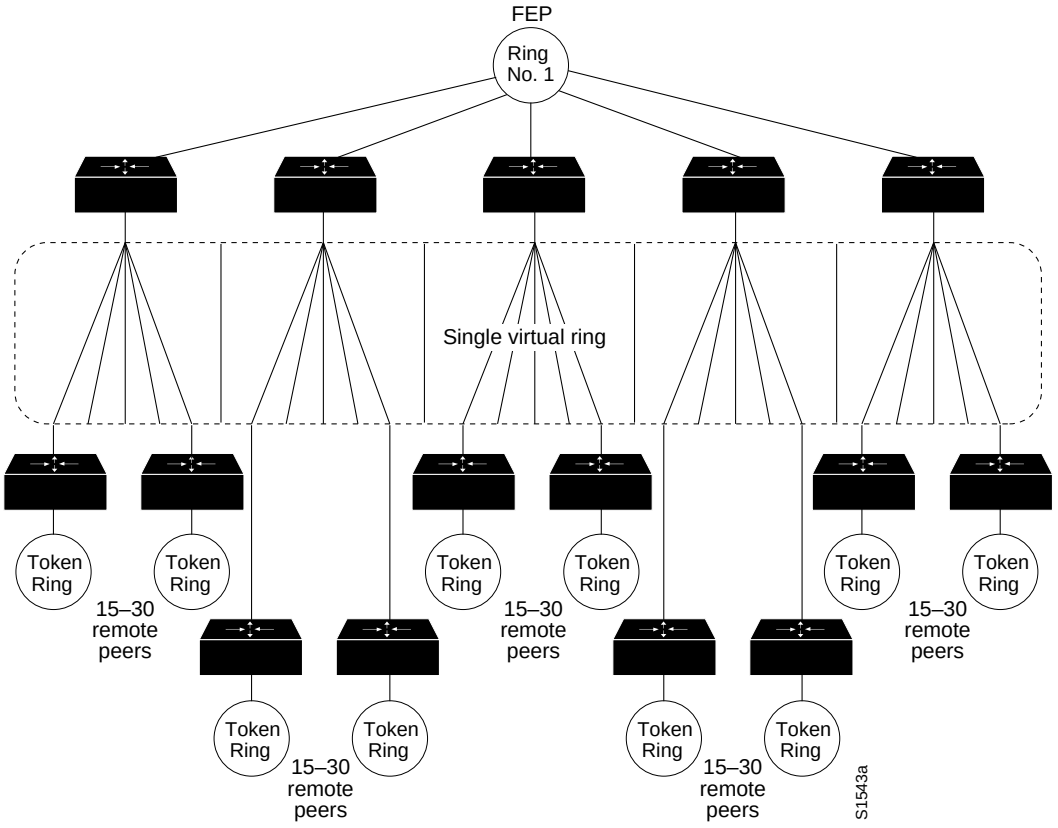
To scale a partially meshed network to diameters of greater than 15 to 100 remote rings, you can take two approaches: build a hierarchical structure of virtual rings or build a scalable partially meshed structure using a single virtual ring.

Proceed with caution to avoid uncontrolled growth in virtual rings, especially in parallel, because parallel virtual rings replicate explorer packets, which causes unnecessary explorer packet traffic.

Scalable Partially Meshed Rings

With a partially meshed ring topology, the objective is to leverage the advantage of a network that does not require "any-to-any" connectivity. Using a single virtual ring, you can connect a series of routers at the FEP sites. For each additional 15 to 100 remote peers, you must add a router to the central site. Figure 3-33 illustrates this kind of environment. Contact your technical support representative for more information about specific limitations and recommendations that might apply to your network specifications.

Figure 3-33 Virtual Ring Environment Interconnecting Multiple Remote Peers

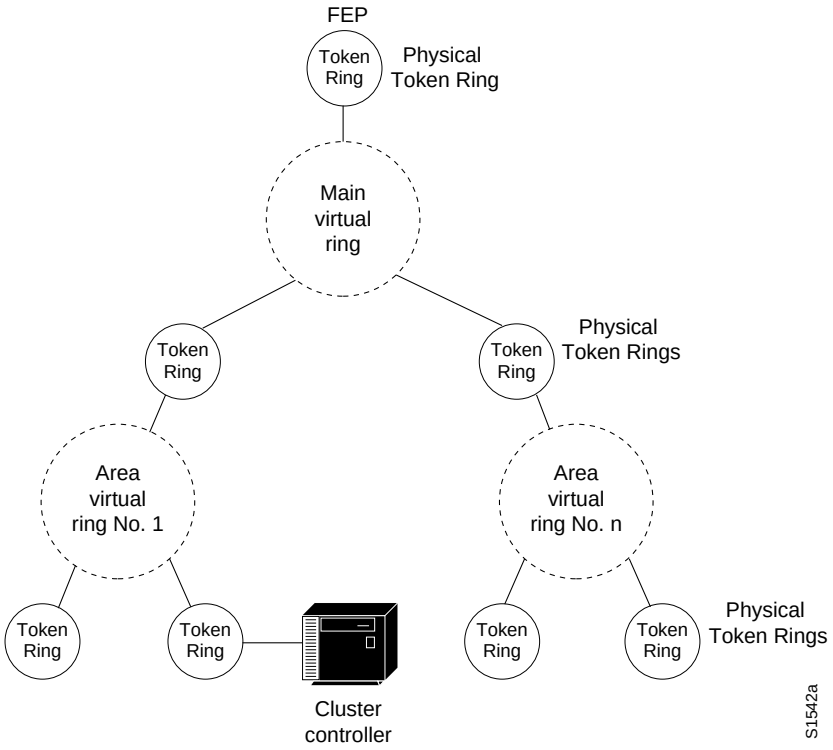


The network illustrated in Figure 3-33 will work for local acknowledgment because all traffic exists on a single virtual ring. A potential problem with this topology is that the number of routers is related to the number of virtual rings, not to the LAN or WAN connectivity at the FEP site. A site with two WAN links and a Token Ring could require several routers if it is the main FEP site.

Hierarchical Virtual Rings

Using *hierarchical* virtual rings, you can use physical Token Rings and virtual rings to create a hierarchy of virtual rings. Figure 3-34 illustrates such a hierarchy.

Figure 3-34 Hierarchical Virtual Ring Topology



Because the destination of all explorer packets is to the core virtual ring, you can use filters to eliminate explorer packet traffic crossing between local-access virtual rings at the point where rings meet. The filters would also filter out FEP traffic. As an alternative, you can use the same virtual ring number for each virtual ring to filter out FEP traffic that might otherwise traverse the local-access virtual rings.

One problem with this design is that the hop count might limit Token Ring SRB connectivity. Because the connectivity from access point to FEP uses four hops, additional local bridges either at the access points or at the central site might not be reachable from the entire network.

Combined Designs

Networks can be built out of both hierarchical designs and scalable partially meshed designs as long as you prevent explorer packet traffic from reexploring access points. To fulfill this requirement, write access lists to prevent explorer packet traffic from entering a ring if that traffic did not originate from the ring that has an FEP.

Hierarchical Design for NetBIOS Environments

The challenge of NetBIOS is that applications might require unrestricted ring-to-ring connectivity. The SRB protocol was not designed to scale, and network designers often demand that routers help scale beyond the original design of the protocol.

Limitations on the maximum number of peers mandates that your only topological option for a NetBIOS SRB network is the hierarchical environment illustrated in Figure 3-34. This design poses certain challenges because of increased explorer packet traffic. It is imperative that you create a single-stream, spanning connection through the network to minimize explorer packets.

To succeed, a hierarchical NetBIOS network needs three elements:

- Proxy explorer
- Aggressive explorer packet caching
- NetBIOS name caching

These features allow switching of valid NetBIOS traffic even under the worst conditions of high explorer packet load. You might not be able to use the partially meshed network design if you have to maintain unrestricted ring-to-ring connectivity. Contact your technical support representative to determine any specific limitations for your network topology and design implementation. Refer to "Explorer Packets and Propagation" and "NetBIOS Broadcast Handling" earlier in this chapter for additional details concerning these topics.

Queuing and Prioritization Schemes

The following information focuses on current prioritization mechanisms. Prioritization tools discussed include:

- Priority queuing
- Custom output queuing
- Service access point (SAP) prioritization
- Logical unit (LU) address prioritization
- SAP filtering

Note The queuing and prioritization schemes described in this section rely on process switching. If the router is configured for fast switching or for autonomous switching, the configuration of a queuing or prioritization scheme will increase processor utilization. However, increased processor utilization is usually not a problem when the router is sending traffic over low speed WAN links.

Priority Queuing (Software Release 9.1)

Priority queuing provides a mechanism to prioritize packets transmitted on an interface.When priority queuing is enabled on an interface, the router maintains four output queues for that interface. During congestion, the packets are queued on one of the four queues according to their priority. The router services all packets on the highest priority queue before moving on to the next highest priority queue. In other words, the queuing delay of a packet on a lower priority queue is nondeterministic: an RSRB session set to normal priority might time out if, for example, IPX packet traffic is heavy and is configured for the highest priority queue.

This scheme introduces a fairness problem in that packets configured for lower priority queues might not be serviced in a timely manner, or at all, depending on the bandwidth used by packets sent from the higher priority queues. Priority queuing does not provide bandwidth allocation.

Priority queuing can be used when there is sufficient bandwidth to accommodate all packets destined for a particular interface, but where packets from certain protocols such as file transfers cause other protocols like Telnet sessions to suffer from poor response.

If there is insufficient bandwidth on an output interface to pass data from various sources, priority queuing cannot solve the limited bandwidth condition. If there is not enough bandwidth to pass all of the data destined for an interface, protocols assigned to the lower priority queues will suffer packet loss.

Priority queuing introduces processor overhead that might be acceptable for slow interfaces, but might be unacceptable for higher speed interfaces such as Ethernet, Token Ring, or FDDI. If you are currently fast switching packets, be aware that priority queuing requires that these packets be process switched, which would negatively impact performance.

Use the **priority-list** global configuration command to define priority lists and the **priority-group** interface command to assign a list to an interface. Priority queuing can be configured instead of, but not in addition to, custom output queuing.

Note Priority queuing does not operate over X.25.

Custom Output Queuing (Software Release 9.21)

With custom output queuing, a *weighted-fair queuing* strategy is implemented for the processing of interface output queues. For each interface, you can control the percentage available bandwidth used by a particular kind of traffic when the available bandwidth is unable to accommodate the aggregate traffic queued.

When custom output queuing is enabled on an interface, the router maintains eleven output queues (numbered from 0 to 10) for that interface that can be used to modify queuing behavior. The router cycles through queue numbers 1 to 10 in a sequential fashion, delivering packets in the current queue before moving on to the next. Associated with each output queue is a configurable byte count, which specifies how many bytes of data should be delivered from the current queue by the router before the router moves on to the next queue.When a particular queue is being processed, packets are sent until the number of bytes sent exceed the queue byte count or the queue is empty.

Queue number 0 is a system queue; its queue is emptied before any of the queues numbered 1 to 10 are processed. The router queues high priority packets to this queue, such as interface keepalive packets. Routing protocol packets are not automatically placed in the system queue.

The custom output queuing implementation should not impact the performance of existing packet queuing code. The queuing algorithm implementation is time critical because it affects packet delivery time when custom output queuing is in use.

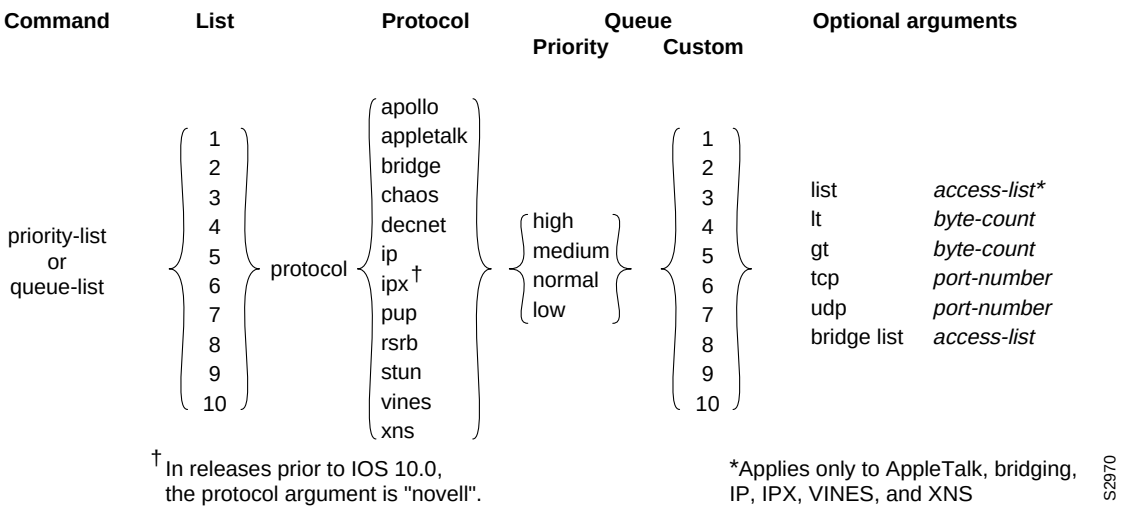
When custom output queuing (or priority queuing) is enabled, it should take much longer for the router to switch packets because each packet has to be classified by the processor card.

Use the **queue-list** global configuration command to define custom queue lists and the **custom-queue-list** interface configuration command to assign a custom queue list to an interface. Custom output queuing can be configured instead of, but not in addition to, priority queuing.

Figure 3-35 describes the syntax of the **priority-list** and **queue-list** commands.

Note Custom output queuing does not operate over X.25.

Figure 3-35 Priority and Custom Output Queuing Command Syntax



SAP Prioritization

The purpose of the SAP prioritization feature is to allow you to specify the priority (precedence and bandwidth) of a protocol over another protocol across the RSRB/SDLLC WAN. The prioritization is based on the destination service access point (DSAP) address and source service access point (SSAP) address.

SAP prioritization can be built based on priority queuing or on custom output queuing. The actual SAP classification code can be developed regardless of the underlying prioritization mechanism. The priority queuing mechanism addresses only the *precedence* criteria. The custom output queuing mechanism provides *precedence* and guarantees *bandwidth*. This section describes SAP prioritization using priority queuing.

To provide a fine granularity in the prioritization of packets, the **sap priority-list** global configuration command (available in Software Release 9.1(9)) allows you to specify any combination of DSAP, SSAP, destination MAC (DMAC) address, and source MAC (SMAC) address.

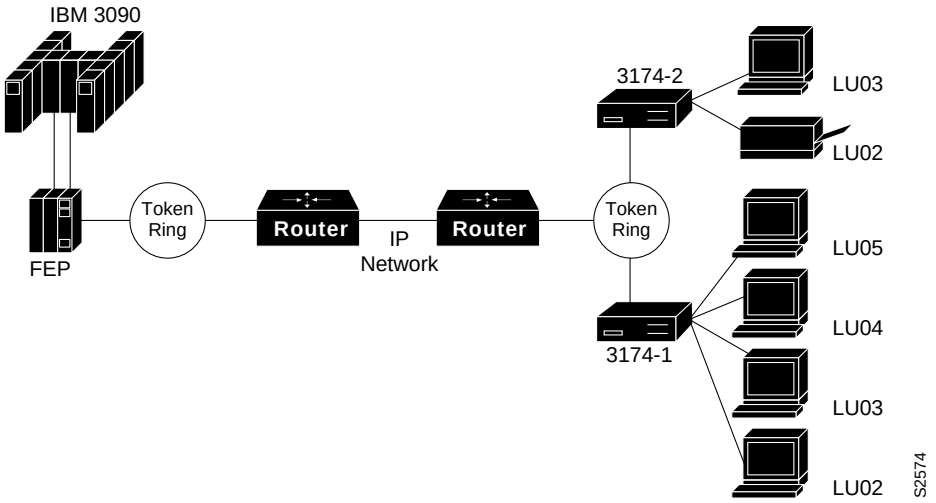
For example, if you want to prioritize all SNA traffic (SAP 04) over NetBIOS traffic (SAP F0), only the DSAP or SSAP must be configured. In contrast, if you want to give precedence to traffic on a particular LLC2 session, you must specify four parameters: DSAP address, SSAP address, DMAC address, and SMAC address. Use the **sap-priority** *list* interface configuration command (available in Software Release 9.1(9)) to tie the priority list to a particular input interface.

You must also specify the **priority** option in the **source-bridge remote-peer** global configuration command to enable SAP prioritization. In addition, you must configure the **priority-list** global configuration command for the appropriate interfaces and using the **priority-group** interface configuration command on the output interface.

Enhanced LU Address Prioritization

The enhanced logical unit (LU) address-prioritization feature allows you to specify the physical unit (PU) on which an LU resides. This is important because multiple PUs on a Token Ring or on a multidropped SDLC line might have LUs with the same LU address. For example, Figure 3-36 illustrates a situation where LU02 on 3174-2 is a 3287 printer, and LU02 on 3174-1 is a 3278 terminal. It is undesirable to assign the same priority to the printer and the terminal.

Figure 3-36 LU Prioritization for RSRB



As of Software Release 9.1(9), the LU address prioritization for both RSRB and serial tunneling (STUN) allow you to prioritize addresses as follows:

- In the RSRB case, in addition to the LU address, you can specify the MAC and SAP address, which together uniquely identify a PU.
- In the STUN case, in addition to the LU address, you can specify the SDLC address to identify a PU on a multidropped SDLC line.

SAP Filters on WAN Links

SAP filters, which are currently available for serial, Token Ring, and Ethernet interfaces, can be used to prevent local NetBIOS and other broadcasts from traversing the RSRB/SDLLC WAN. To implement SAP filter logic on the RSRB/SDLLC WAN interface, it is desirable to place the code at the RSRB level independent from the encapsulation type on the interface. The same filter code should work for direct HDLC encapsulation, TCP/IP encapsulation, and FST encapsulation. In addition to filtering by SAP address, SAP filters can also be used to filter packets by NetBIOS name.

The commands, which are available in Software Release 9.1.(9) are the same as those used for SRB on the Token Ring interface:

- The **access-list** *list* global configuration command builds the access list.
- The **rsrb remote-peer** *ring-group* interface configuration command filters by Local Service Access Point (LSAP) address or by NetBIOS station name on the RSRB WAN interface.
- The **netbios access-list** *host* global configuration command builds a NetBIOS access filter by host name.

SRB Design Checklist

Before implementing a source-route bridging (SRB) network, be sure to familiarize yourself with the technical reference material in the *Router Products Configuration Guide* and the *Router Products Command Reference* publication that deal with SRB internetworking.

Next, read "Multiport Bridging" through "WAN Framing" earlier in this chapter. Depending on your implementation, you might also need to review "IP Routing Protocol Selection for SRB Networks" and "SRB Network Design" earlier in this chapter.

If you require more than eight routers continue as follows:

Step 1 Evaluate the following requirements:

Determine which protocols are in use or are to be used. Relevant options are hierarchical Systems Network Architecture (SNA) and NetBIOS. If you are running hierarchical SNA, determine the link speeds to the core front end processor (FEP) sites.

Determine whether parallel paths exist in the network either between individual routers or in the general network. If they do, refer to "WAN Parallelism" earlier in this chapter.

Determine whether the network requires greater than 2-kilobyte frames to be sent across WAN links. If so, refer to "WAN Frame Sizes" earlier in this chapter.

Step 2 If the access ring and the FEP-connected sites exceed 15 Token Rings, you must address the following topics:

- Determine whether local acknowledgment is a requirement (refer to "Local Acknowledgment Recommendations" earlier in this chapter).
- Select an encapsulation method. (Refer to "WAN Framing.")
- Design a network topology incorporating the rules outlined in "SRB Network Design."
- Select a routing protocol described in "WAN Parallelism" and "IP Routing Protocol Selection for SRB Networks."

Step 3 If performance is an important concern for your internetwork, review "IP Routing Protocol Selection for SRB Networks."

Step 4 Prepare each router's configuration for the following:

- SRB (Refer to "Explorer Packets and Propagation" and "WAN Framing.")
- IP route tuning (Refer to "IP Routing Protocol Selection for SRB Networks.")

Step 5 Turn on proxy explorer as needed. (Refer to "Explorer Packets and Propagation.")

Step 6 If the network requires using NetBIOS, proceed as follows:

- Turn on NetBIOS name caching.
- Limit the explorer packet processing queue to 20 entries. (Refer to "Explorer Packets and Propagation.")

Step 7 If you expect to exceed 250 Token Rings within the next 12 months, contact your technical support representative for additional information.

CHAPTER 4

Designing SDLC, SDLLC, and QLLC Internetworks

Internetworking in IBM System Network Architecture (SNA) environments often involves making special accommodations for entities not originally intended to be connected to meshed internetworks. This chapter addresses some of the challenges of this environment and aims to help you successfully implement routing technology within an SNA environment.

This chapter describes three techniques designed to enable internetworking capabilities for SNA-based network architectures:

- SDLC via STUN
- SDLLC Implementation
- QLLC Conversion

The sections describing serial tunneling (STUN), Synchronous Data Link Control (SDLC) over the Logical Link Control type 2 (LLC) protocol (SDLLC), and Qualified Logical Link Control (QLLC) focus on the following topics:

- Technology overview and issues
- Router technology options, implementation guidelines, and configuration examples

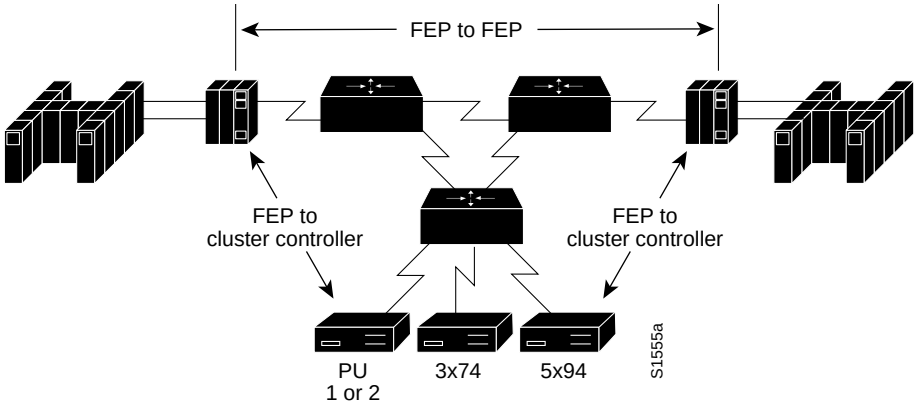
Note For information about IBM serial lines, refer to Appendix B, "IBM Serial Link Implementation Notes."

SDLC via STUN

SDLC via serial tunneling (STUN) involves encapsulating SDLC frames into Internet Protocol (IP) packets and routing the encapsulated packets over IP-supported network media. The SDLC frame is transmitted without modification and the information within the frame is transparent to the network. All SNA physical unit (PU) types are supported. This discussion focuses on the SDLC data link protocol and its various configurations and is followed by a close look at how STUN is implemented. Figure 4-1 illustrates various elements of STUN configuration in an environment that includes front-end processors (FEPs) and cluster controllers.

Note For a case study on how to configure STUN for FEPs, see Topic 3, "STUN for Front-End Processors," in the Cisco publication, *Internetworking Applications: Case Studies*, Vol 1. No.2.

Figure 4-1 Sample STUN Network Configuration



SDLC Data Link

SDLC is the synchronous, bit-oriented protocol used by the SNA data-link control layer. As formally defined by IBM, SDLC is a line discipline for managing synchronous, code-transparent, serially transmitted bit information over a data link. Transmission exchanges can be full duplex or half duplex and can occur over switched or nonswitched links. The configuration of the link connection can be point-to-point, multidrop, or loop.

Common physical link-layer implementations are V.24 (EIA/TIA-232, formerly RS-232), V.35, and X.21. This section describes SDLC as it applies to STUN.

The SDLC data link allows a reliable exchange of information over a communication facility between SNA devices. The protocol provides for synchronization of receiver and transmitter, with detection of and recovery from transmission errors. It does so through acknowledgment of frame receipt and by performing a cyclic redundancy check (CRC) on the data.

Supported Data Link Configurations

This section provides information related to router-specific hardware implementation. Table 4-1 provides a matrix of SDLC support for V.24.

Table 4-1 SDLC Support for V.24 (EIA/TIA-232)

Product Type	NRZ/NRZI	DTE/DCE	Full Duplex	Half Duplex	Maximum MTU
Cisco 7000	Both	Both	Yes	Yes	4 KB
Cisco 7010	Both	Both	Yes	Yes	4 KB
AGS+	Both	Both	Yes	Yes	4 KB
MGS	Both	Both	Yes	Yes	4 KB
Cisco 2500	Both	Both	Yes	Yes	8 KB
Cisco 4000	Both	Both	Yes	4T card only	8 KB
Cisco 4500	Both	Both	Yes	4T card only	8 KB
Cisco 3104	Both	Both	Yes	Dual serial card only	8 KB
Cisco 3204	Both	Both	Yes	Dual serial card only	8 KB

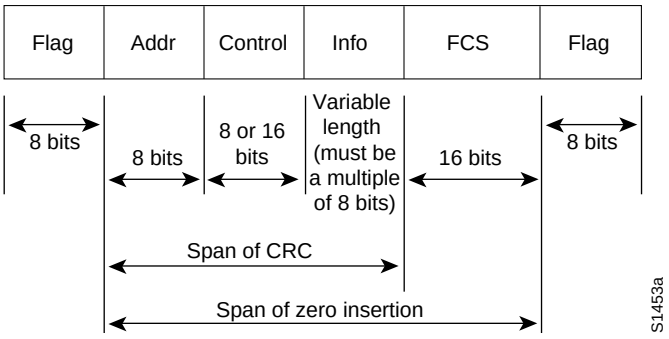
The following notes apply to the entries in Table 4-1:

- For the Cisco 7000, Cisco 4000, Cisco 4500, and Cisco 3000 products, support of data terminal equipment (DTE) or data communications equipment (DCE) functionality depends on the cable used.
- For the AGS+ and MGS, if you are using a nonreturn to zero inverted (NRZI) applique, the systems supports DCE natively. To support DTE mode operation, a special cable is required. Prior to the availability of the NRZI applique, either a DCE or DTE applique was specifically ordered.
- Half-duplex support is available for the AGS+ and MGS with Software Release 9.1(7) or later. The NRZI applique, three-port SCI card, and Software Release 9.1(7) or later are all required for half-duplex support.
- Half-duplex support is available for the Cisco 4000 and the Cisco 4500 only with the 4T card.
- Prior to Software Releases 8.3(6), 9.0(3), or 9.1(2), only 2-KB frame sizes were supported. When increasing maximum transmission unit (MTU) size, consider interface card buffer memory size constraints.

SDLC Frame Format

The SDLC frame has a specific format as illustrated in Figure 4-2.

Figure 4-2 SDLC Frame Format



The *Flag* field starts and ends the frame and initiates and terminates error checking. When the link is idle, it is common to transmit streaming flags to maintain link synchronization, but is not necessary to keep the link up. The router streams flags.

The *Addr* field contains the SDLC address of the secondary station—whether the frame is coming from the primary or secondary. The Addr field can contain a specific address, a group address, or a broadcast address. Routers support specific addresses and support broadcast addressing on a limited basis.

The *Control* field is a 1-byte field (for modulo 8 frames) or a 2-byte field (for modulo 128 frames). The extra byte is required for modulo 128 frames to accommodate larger send and receive *frame count fields*. The value of the Control field identifies three different frame formats, as shown in Table 4-2.

Table 4-2 Components of the Control Field

Format	Binary Configuration	Hexadecimal Equivalent (P/F off)	Hexadecimal Equivalent (P/F on)	Command Name	Acronym
Unnumbered Format	000 P[1]/F[2] 0011	03	13	Unnumbered Info	UI
	000 F 0111	07	17	Request Init. Mode	RIM
	000 P 0111	07	17	Set Init. Mode	SIM
	000 F 1111	0F	1F	Disconnect Mode	DM
	010 F 0011	43	53	Request Disconnect	RD
	010 P 0111	43	53	Disconnect	DISC
	011 F 0011	63	73	Unnumbered Ack	UA
	100 P 0011	83	93	Set Normal Response. Mode	SNRM
	110 P 1111	CF	DF	Set Normal Response. Mode Ex.	SNRME
	100 F 0111	87	97	Frame REJECT	FRMR
	101 P/F 1111	AF	BF	Exchange ID	XID
	111 P/F 0011	E3	F3	Test	TEST
Supervisory Format	RRR[3] P/F 0001	*x*1[4]	*x*1	Receive Ready	RR
	RRR P/F 0101	*x*5	*x*5	Receive Not Ready	RNR
	RRR P/F 1101	*x*9	*x*9	Reject	REJ
Information Format	RRR P/F SSS0[5]	*xx*	*xx*	Numbered Info Present	Transfer

1. P = Poll bit
2. F = Final bit
3. RRR = Nr (receive count)
4. x = Any single digit hexadecimal value
5. SSS = Ns (send count)

The *Info* field is a variable-length field containing a path information unit (PIU) or exchange identification (XID) information. Table 4-3 lists supported PIUs.

Table 4-3 PIU Support

PIU Type	Router Support
PIU FID0-bisync and start/stop (non-SNA)	Not supported
PIU FID1-host channel to FEP to remote FEP	Supported via STUN
PIU FID2-FEP to cluster controller (PU 2)	Supported via STUN and SDLLC
PIU FID3-FEP to SNA terminal (PU 1)	Supported via STUN
PIU FID4-FEP to FEP using virtual route	Supported via STUN
PIU FIDF-FEP to FEP (VR SNF overflow sweep)	Supported via STUN
XID 2-Contains PU parameters for PU types 1, 2, 4, and 5	PU types 2 and 4 supported via STUN and SDLLC
XID 3-APPN variable format for PU 2 and PU 2.1	Not supported

The *frame check sequence* (*FCS*) field is a 2-byte field that, for transmitted data, contains the result of a CRC performed on the first bit following the Flag field through the last bit of the Info field or Control field (if the frame is an *Unnumbered* or a *Supervisory* format). As the remote device receives the data, it performs the same CRC computation and compares the result with the contents of the FCS field. If the comparison fails to find a match, the frame is discarded and recovery procedures take place.

STUN Configuration for SDLC

The following sections provide design and implementation information for a variety of STUN-related configuration topics.

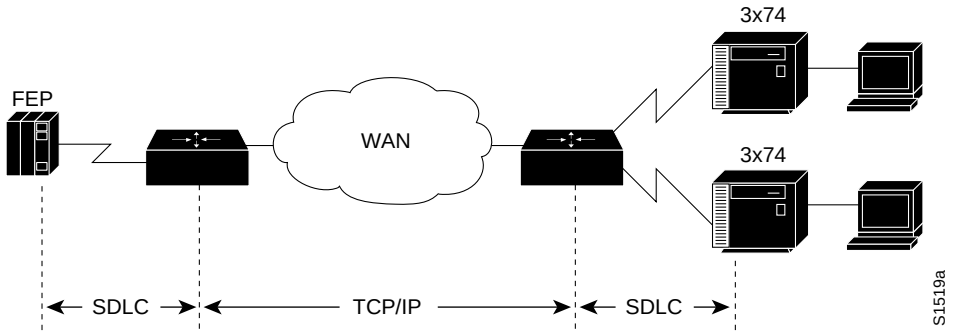
- Local Acknowledgment
- Virtual Multidrop
- SDLC Broadcast across Virtual Multidrop Lines (IOS Release 10.2)
- SDLC Address Prioritization
- SDLC Two-Way Simultaneous Mode (IOS Release 10.2)
- LU Address Prioritization
- Flow Control
- Transmission Groups and Class of Service Capabilities
- SNA Host Configuration Considerations for STUN

Local Acknowledgment

Local termination of SDLC sessions allows frames to be locally acknowledged by the receiving router. By locally terminating SDLC sessions, acknowledgment and keepalive traffic is prevented from traversing the backbone, and SNA sessions are preserved in the event of network failure.

Local acknowledgment locally terminates Supervisory frames, which include Receiver Ready, Receiver Not Ready, and Reject. Figure 4-3 illustrates the operation of SDLC local acknowledgment.

Figure 4-3 STUN-to-SDLC Local Acknowledgment



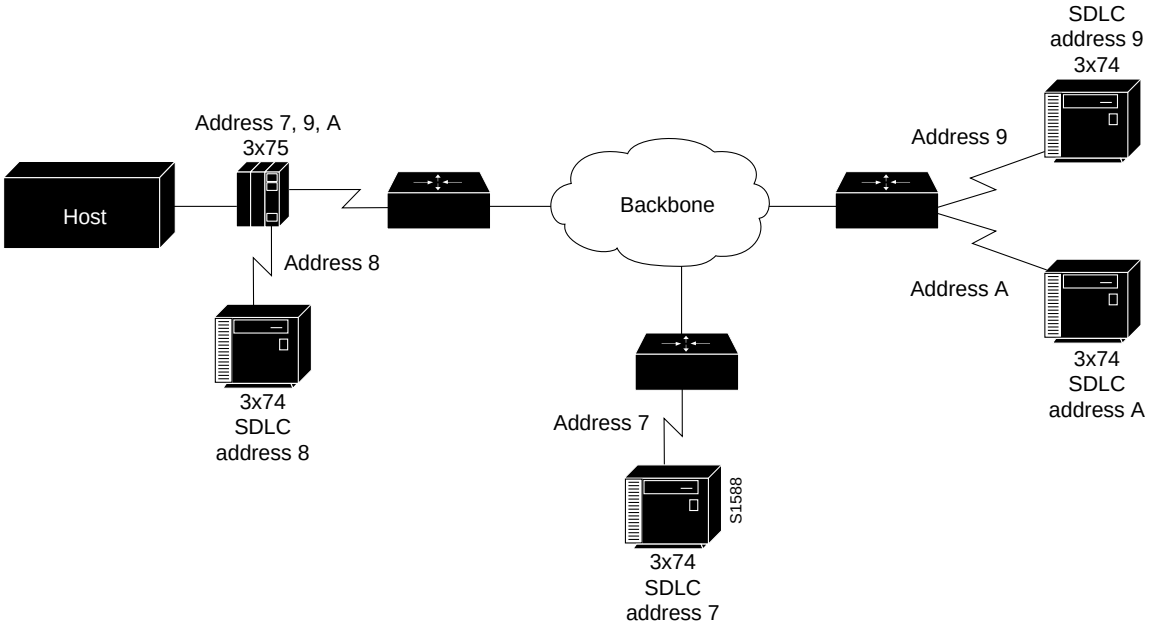
Note Local acknowledgment requires that TCP/IP sessions be maintained between the routers as a means to provide reliable transport.

Virtual Multidrop

Virtual multidrop exploits SDLC address mapping to allow an FEP to communicate with multiple cluster controllers. With a virtual multidrop configuration, the address of each SDLC frame is checked individually. Only addresses that match the configuration are forwarded to the specified

destination, which allows an FEP to communicate with multiple 3174s from a single serial link (that is, multidrop). You can also use SDLC address mapping as a security feature to restrict access based on SDLC address, as shown in Figure 4-4.

Figure 4-4 SDLC Transport in Virtual Multidrop Environment



The following steps are required to establish the network configuration illustrated in Figure 4-4:

Step 1 Include a LINE definition in the Network Control Program (NCP) running in the FEP, followed by PU definitions for address 7, 9, and A. These definitions are interpreted by the NCP as a multidrop link.

Step 2 Use the **stun route address tcp** global configuration command to specify how to forward frames.

Step 3 Determine if priority queuing is required. If so, local acknowledgment is required.

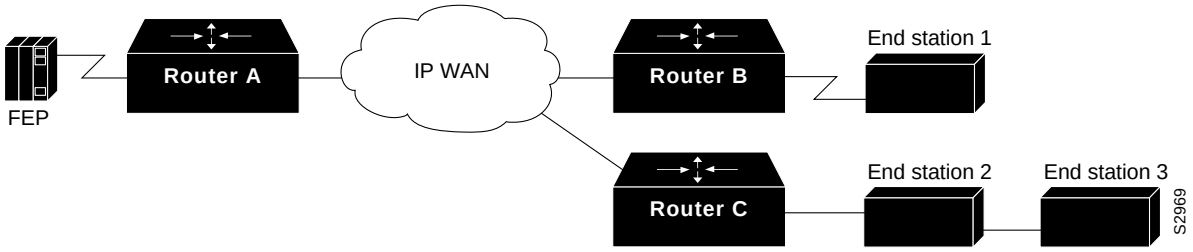
Step 4 Determine if local acknowledgment is required.

SDLC Broadcast across Virtual Multidrop Lines (IOS Release 10.2)

The SDLC broadcast feature allows SDLC broadcast address 0xFF to be replicated for each of the STUN peers, so that each end station receives the broadcast frame.

In Figure 4-5, the FEP views the end stations as if they were on an SDLC multidrop link. Router A duplicates any broadcast frames sent by the FEP and sends them to any downstream routers (in this example, Router B and Router C).

Figure 4-5 SDLC Broadcast in Virtual Multidrop Line Environment



The **sdlc virtual-multidrop** interface configuration command enables SDLC broadcast and should only be used on the router that is configured as the secondary station on the SDLC link. In addition, the **stun route address tcp** command for SDLC address 0xFF must be configured on the secondary station (in this example, Router A) for each STUN peer. A sample configuration follows:

```
stun peername xxx.xxx.xxx.xxx
stun protocol-group 1 sdlc
!
interface serial 1
encapsulation stun
stun group 1
stun sdlc-role secondary
sdlc virtual-multidrop
sdlc address 01
sdlc address 02
sdlc address 03
stun route address 01 tcp yyy.yyy.yyy.yyy local-ack
stun route address 02 tcp zzz.zzz.zzz.zzz local-ack
stun route address 03 tcp zzz.zzz.zzz.zzz local-ack
stun route address FF tcp yyy.yyy.yyy.yyy
stun route address FF tcp zzz.zzz.zzz.zzz
```

SDLC Address Prioritization

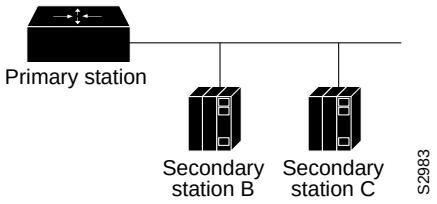
For STUN prioritization of SDLC addresses over simple serial transport connections, use the **priority-list** or the **queue-list** global configuration command and use the **priority-group** interface configuration command or the **custom-queue-list** interface configuration command, respectively, on the interface that connects the remote router (the output interface).

For STUN prioritization of SDLC addresses over TCP/IP transport connections, you must configure the **priority-list** global configuration command and use the **priority-group** interface configuration command on the interfaces that connect to the end devices (the input interfaces). Also, the **local-ack** and **priority** keywords of the **stun route address tcp** global configuration command must be specified.

SDLC Two-Way Simultaneous Mode (IOS Release 10.2)

Two-way simultaneous mode allows a router that is configured as a primary SDLC station to achieve better utilization of a full-duplex serial line. When two-way simultaneous mode is enabled in a multidrop environment, the router can poll a secondary station and receive data from that station while it sends data to or receives data from a different secondary station on the same serial line. (See Figure 4-6.)

Figure 4-6 Two-Way Simultaneous Mode in a Multidrop Environment



The **sdlc simultaneous** command enables two-way simultaneous mode in a multidrop environment.

When two-way simultaneous mode is enabled for a point-to-point connection to a secondary station, the router can send data to the secondary station even though there is an outstanding poll, as long as the window size limit is not reached. The **sdlc simultaneous single** command enables two-way simultaneous mode in a point-to-point link environment.

LU Address Prioritization

To prioritize logical units, use the **locaddr-priority-list** global configuration command on each router. For example:

```
locaddr-priority-list 1 02 high
locaddr-priority-list 1 03 high
locaddr-priority-list 1 04 low
```

You must also assign a priority list to the STUN priority ports using the **priority-list** global command. For example:

```
priority-list 1 protocol ip high tcp 1994
priority-list 1 protocol ip medium tcp 1990
priority-list 1 protocol ip low tcp 1991
```

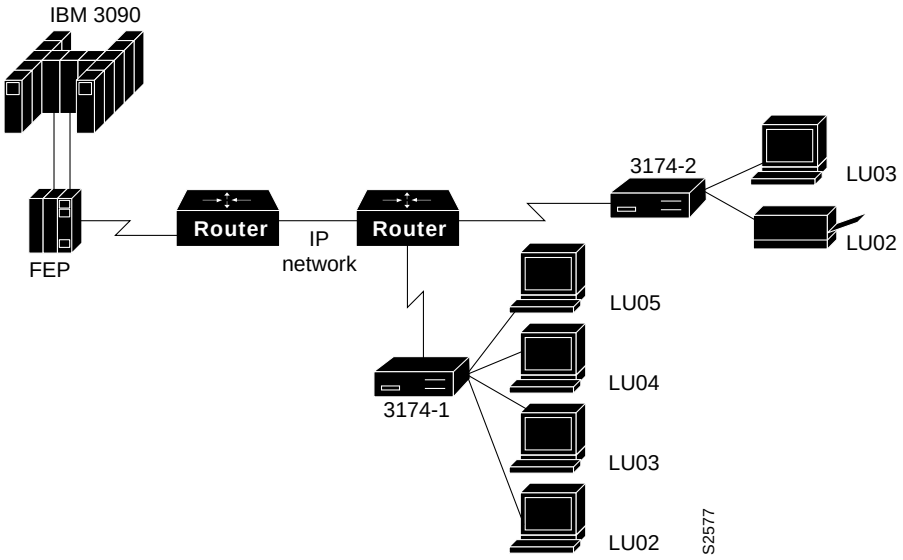
The serial interfaces attached to the end systems (input interfaces) must be associated with priority lists using the **locaddr-priority** and **priority-group** interface configuration commands. The **locaddr-priority** command links the interface to a local LU priority list (specified with the **locaddr-priority-list** global configuration command). The **priority-group** command links the interface to a TCP priority list (specified with a **priority-list** global configuration command). For example:

```
interface serial 1
locaddr-priority 1
priority-group 1
```

In addition, you must specify the **local-ack** and **priority** keyword options of the **stun route address tcp** global configuration command.

The LU address prioritization feature has been enhanced to allow you to specify the PU on which a LU resides. This enhancement is important because there might be multiple PUs on a multidropped SDLC line that have the same LU address. For example, in Figure 4-7, LU02 on 3174-2 is a 3287 printer, and LU02 on 3174-1 is a 3278 terminal. It is undesirable to assign the same priority to the printer and the terminal.

Figure 4-7 LU Prioritization for STUN



As of Software Release 9.1(9), LU address prioritization for both RSRB and STUN solved this problem. In addition to the LU address, you can specify the SDLC address to identify a PU in a multidropped SDLC line.

The syntax of the **locaddr-priority** global configuration command follows:

locaddr-priority *list lu-address* **sdlc** *secondary*

The keyword **sdlc** indicates the next byte (in hexadecimal), and *secondary* is the secondary SDLC address. (This syntax is supported in Software Release 9.1(9).)

Flow Control

SDLC level flow control is also offered with local termination. When the router detects that the TCP queue is 90 percent full, it blocks further SDLC frames until the TCP queue recedes to 80 percent full. This is accomplished by transmitting SDLC Receiver Not Ready (RNR) frames.

There is also a flow control protocol between STUN peers. When SDLC output queues become congested, a router can request the remotely attached router to exert back-pressure on the SDLC link.

Transmission Groups and Class of Service Capabilities

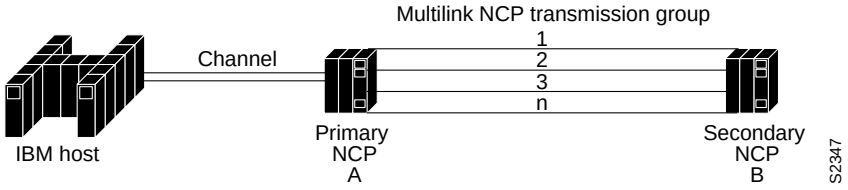
This section describes transmission group and class of service (COS) support provided by Cisco routers. Information provided here includes the following topics:

- NCP-to-NCP Communications over a Routed Network
- Transmission Group and COS Design Guidelines and Notes

Typical NCP-to-NCP Communications

In a typical NCP-to-NCP communications arrangement, a host is channel-attached to an FEP acting as an NCP. In Figure 4-8, NCP A is the primary SDLC station and NCP B (remote NCP) is a secondary SDLC station. The NCPs dynamically determine their relationship: the NCP with the higher subarea number becomes the primary SDLC station. NCP V5R4 and later allows you to configure which NCP is the primary and which NCP is the secondary.

Figure 4-8 Typical NCP-to-NCP Multilink Transmission Group Communication Configuration



A *transmission group* is defined as one or more parallel SDLC links connecting adjacent PU Type 4 (NCP) nodes. Transmission groups are used to increase the reliability of the logical link connection between NCPs and to provide additional bandwidth capacity. When one link fails or is congested, data is routed on one of the other links in the group. The transmission group function is implemented at the path control (PC) layer of the NCP architectural model. The PC layer encapsulates request/response units in PIUs and sends them to the data link control (DLC) layer for transmission.

The PC layer uses the transmission header of the PIU to route messages through the network. SNA defines different transmission header formats and identifies the different formats by Format Identification (FID) type. A transmission header of type FID 4 is used to route data between type 4 nodes that support explicit and virtual routes.

The NCP assigns a sequence order number to each link in a transmission group. In later versions of NCP, you can specify the sequence order in which an NCP should use the transmission group links; otherwise this order is determined by the order of link activation. Deactivation and reactivation of a link cause it to become the last activated link in the transmission group, and PIU traffic will be sent on the last activated link only if all other links fail or are busy.

Traditionally, the PC layer communicates directly with the DLC layer to transmit PIUs. When sending PIUs over a multilink transmission group, a transmission group layer exists between the PC and DLC layers. The transmission group layer contains a transmit queue. When the transmission group layer gets a frame to send, it checks for the availability of a link in the transmission group in priority (activation default) order. If the transmission group layer finds an available link (that is, a link that is not down and is not busy), it assigns the PIU the next NCP sequence number and sends the frame on that link. NCP sequence numbers range from 0 to 4095 and wrap on overflow.

When all links in the transmission group are busy, PIUs are placed in the transmission group transmit queue to await transmission. PIUs accumulate in the queue until a link becomes available. When an SDLC link becomes available, a COS algorithm is performed on the PIUs in the transmit queue. The PIU with the highest priority is dequeued, assigned the next NCP sequence number, and sent on the available link. It is important that sequence numbering occurs upon removal from the transmit queue because PIUs can overtake each other on the transmit queue when COS priority processing is performed. PIUs are never preempted by other PIUs on the same SDLC link queue.

There are several reasons why PIUs might arrive at the receiving NCP out of transmission group sequence: links that operate at different speeds, PIUs on different links that have different lengths, and SDLC link transmission errors that cause retransmission. Because PIUs can arrive out of order,

the receiving FEP performs resequencing by queuing incoming PIUs if their sequence number is larger than the next expected sequence number. The algorithm is not important, but the rule is that the receiving FEP propagates PIUs by sequence number order. PIUs with a lower sequence number than expected are considered duplicates and discarded.

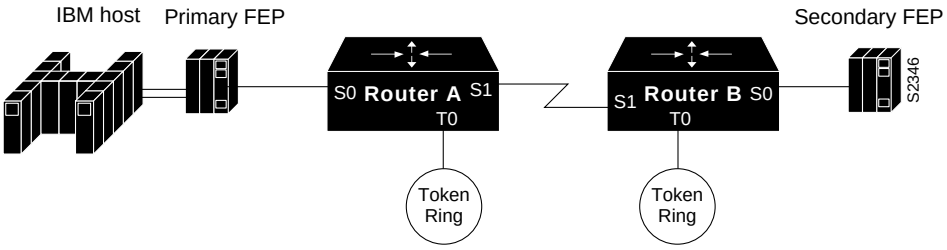
Later versions of NCP deviate from the SDLC standard in their use of SDLC echo addressing. The SDLC secondary NCP sets the high-order bit of the SDLC address when sending a response. For example, the primary NCP sends frames with address 01, and the secondary NCP sends frames with address 81. This addressing scheme limits the range of SDLC addresses from 01 to 7F. Additionally, SDLC address FF is used on frames preceding and during the NCP's XID exchange. The XID and DM frames use the broadcast address.

Another deviation from the SDLC standard occurs in NCP-to-NCP communication when a host NCP loads a remote NCP. Normally, numbered Information frames can be sent only after an SNRM (or SNRME), which resets the station counts to ensure that the two stations start with consistent counts. When a host NCP loads a remote NCP, the host sends a SIM, and the remote NCP responds with a RIM, which allows numbered Information frames to flow. NCPs are permitted to violate the SDLC standard with echo addressing, broadcast addressing, and sending numbered Information frames before SNRM because they only violate the standard when communicating with other NCPs.

NCP-to-NCP Communications over a Routed Network

There are several reasons to allow routers to carry NCP-to-NCP traffic. With the high cost of WAN (leased-line) bandwidth, routers allow other protocols to share the bandwidth set aside for SNA traffic. Additionally, a leased line is required for each line in a transmission group in the traditional NCP-to-NCP communications. The number of leased lines can be collapsed into one leased line. In addition routers provide dynamic routing capabilities, so the wide area network path chosen by a router is dynamically determined. This can result in a more reliable network. Figure 4-9 illustrates NCP-to-NCP communications in a router-based internetwork.

Figure 4-9 NCP-to-NCP Communications over a Routed Network



This chapter is not intended to address STUN pass-through using direct HDLC or TCP/IP encapsulation. There are several reasons to move from STUN pass-through to STUN local acknowledgment:

- Keep SDLC poll (RR) traffic off of an overutilized WAN
- Prevent NCP timers from expiring due to network delay
- Collapse multiple WAN leased lines into one leased line
- Reduce store-and-forward delays through the router network

Consider the situation illustrated in Figure 4-9. Notice that the router attached to the SDLC primary NCP (Router A) acts as an SDLC secondary station, and vice versa for the other router (Router B). For this discussion, assume that all serial links between NCPs and routers are in the same transmission group. This means that the NCP considers the lines to be in transmission group X, and the router considers the lines to be in transmission group Y. There is no relationship between X and Y. X is used in the NCP system generation, and Y is used in the router configuration.

Transmission Group and COS Support

The following features facilitate the support of transmission groups and COS in Cisco routers:

- SDLC address violation allowances

  Two specific instances are exempt from SDLC addressing restrictions:

  — Echo addressing

  — Broadcast addressing

- Remote NCP load sequence

  When a remote NCP is being loaded, the remote NCP is capable of minimal SDLC function. It cannot go into Normal Response Mode (NRM) until it is loaded. The load sequence for a remote NCP starts with a SIM/RIM exchange between NCPs, which initializes each NCP's SDLC frame count to zero. After the SIM/RIM exchange, the NCP's pass numbered Information frames, which normally does not occur until after a SNRM/UA sequence. For the router's SDLC transmission group local-acknowledgment support to allow loading of remote NCPs, the routers pass through all frames after a SIM/RIM sequence and before a SNRM/UA sequence. After the SNRM/UA exchange, normal local acknowledgment occurs.

- Rerouting in multilink transmission groups

  When a router acknowledges an Information frame, it must ensure delivery of that frame to the receiving NCP. If, after the frame is locally acknowledged, the corresponding link in the receiving transmission group is lost, the receiving router reroutes the frame onto another SDLC link in the same transmission group.

- COS

  COS is performed in the sending NCP. Each PIU is assigned a sequence number. The best service the routers can perform is to try to preserve the COS as assigned by the sending NCP via sequence numbers. Therefore, all SNA data PIUs are treated equally with the goal to preserve PIU order. However, virtual route pacing responses flow at SNA network priority level and do not have sequence numbers (that is, they have a sequence number of 0). The router prioritizes all SNA network priority PIUs higher than SNA data to achieve more efficient virtual route pacing.

Note The router cannot use the PIU to determine whether traffic is interactive or batch. Even if the router could make this determination, prioritizing one type of traffic over another would cause the receiving NCP to waste CPU time resequencing the PIUs and would also degrade throughput because the receiving NCP would hold PIUs longer when resequencing.

- Flow control tuning for better COS operation

  The **tcp-queue-max** keyword of the **stun route address tcp** global configuration command allows you to tune the size of the outbound TCP queue so that when the WAN becomes congested, frames generated by an NCP can be stored in the router as well as in the NCP. When the size of the outbound TCP queue is small, back-pressure via SDLC RNRs is applied to sending NCPs sooner, causing the NCP to hold more frames. The more frames that are held by the NCP, the more frames to which the NCP's COS algorithm is applied. The size of the outbound TCP queue should not be configured below 70.

Transmission Group and COS Design Guidelines and Notes

The following guidelines and notes should be considered when implementing transmission groups and COS:

1. Bandwidth of the WAN should be greater than or equal to the aggregate bandwidth of all the serial lines. If other protocols are also using the WAN, bandwidth of the WAN should be greater than the aggregate bandwidth of all the serial lines.
2. If the network delay associated with one line of an NCP transmission group is different from the network delay associated with another line in the same NCP transmission group, the receiving NCP spends additional time resequencing PIUs. This happens when one or more of the NCP transmission group lines is routed and one or more lines is directly connected between NCPs.
3. With the Software Release 9.1 prioritizing algorithm, only the highest priority traffic is guaranteed to get through. With Software Release 9.21, prioritization flexibility is enhanced with the addition of *custom output queuing*. Custom output queuing can be used to guarantee specific bandwidth allocated to protocols with respect to bandwidth allocated to other protocols.

   If you are using Software Release 9.1 and an SNA WAN as a multiprotocol backbone, give SNA traffic the highest priority and assign the next highest priority to other mission critical protocols. Table 4-4 lists equivalent commands for configuring priority queuing and custom output queuing.

Table 4-4 Comparison of Priority Queuing and Custom Output Queuing Configuration Commands

Priority Queuing	Custom Output Queuing
`priority-list 4 protocol ip high tcp 1994`	`queue-list 2 protocol ip 1 tcp 1994`
`priority-list 4 protocol ip medium tcp 1992`	`queue-list 2 protocol ip 2 tcp 1992`
`priority-list 4 protocol ip normal tcp 1991`	`queue-list 2 protocol ip 3 tcp 1991`
`priority-list 4 protocol ip low tcp 1990`	`queue-list 2 protocol ip 4 tcp 1990`

   In addition, make sure that your WAN bandwidth is significantly greater than your aggregate SNA serial line bandwidth to ensure that your SNA traffic does not monopolize the WAN.

4. When NCPs are directly connected, their poll/pause timers should be configured for maximum throughput using the NCP PAUSE statement. Configuration of this parameter depends on whether the NCP is acting as a primary or secondary SDLC station. Table 4-5 outlines the defaults and recommendations as specified in the IBM publication *Tuning and Problem Analysis for NCP SDLC Devices*.

Table 4-5 NCP PAUSE Parameter Guidelines

Pause Statement Parameter	IBM Guideline
NCP primary PAUSE	If the NCP has no data to send, this is how long the NCP will wait between sending polls. (Default is 0.2 seconds; 0 is recommended)
NCP secondary PAUSE	Specifies the time that the secondary NCP will wait before returning a frame with the Final bit set. (Default is 0.2 seconds; recommended to be high –0.2 to 1.0 seconds)

Adding routers with local acknowledgment creates two SDLC sessions instead of one. The result is that the two SDLC sessions do not preserve the original characteristics of the original NCP-to-NCP SDLC session. To adapt a secondary NCP to the router environment, change its system generation PAUSE statement to a value between 0.0 and 0.1 seconds, inclusive.

SNA Host Configuration Considerations for STUN

When designing STUN-based internetworks featuring routers and IBM SNA entities, you must carefully consider the configuration of SNA nodes and routing nodes. Tables in Appendix D, "SNA Host Configuration for SDLC Networks," provide examples of SNA host configurations that focus on two specific SNA devices:

- FEP configuration for SDLC links
- 3174 SDLC configuration example

STUN Implementation Checklist

Before implementing a serial tunneling (STUN) internetwork, make sure you are familiar with the information in publications *Router Products Configuration Guide* and the *Router Products Command Reference* that deal with Synchronous Data Link control (SDLC). Depending on your implementation, you may need to review "SDLC via STUN" earlier in this chapter.

Use the following steps as a checklist when implementing SDLC STUN in your internetwork:

Step 1 Evaluate your current environment. Compile the following list of items and use it as a starting point for integrated router-based internetworking solutions.

- What host-attached cluster controllers or front end processors (FEPs) are being used (such as 37x5, 3172, and 3174)? The host site might be referred to as a local, core, or backbone site, or as a data center.
- Through what media is the network connected to the host site?

  STUN: Serial connection at both ends.

  SDLLC: Token Ring at primary station and SDLC at secondary station, or Ethernet at primary stations and SDLC at secondary station.

  Reverse SDLLC: SDLC at primary station and Token Ring or Ethernet at secondary station.
- What are the link speeds for local and remote end systems?
- What are the current SDLC line utilization measurements of those links that will attach to the router? This information will be helpful in determining the site requirements.

- What interface types are to used? For example:V.24 (EIA/TIA-232, formerly RS-232), V.35, X.21.
- What modems, data service units (DSUs), channel service units (CSUs), or modem-sharing or line-sharing devices are to be used?
- What remote end system types are involved? For example: 3174, 3274, or AS/400.
- What kind of emulation requirements are involved? For example: half or full duplex, NRZ or NRZI.
- What are the current transaction response times? Consider peak load periods and characterize traffic patterns.
- How many PUs are in place? How many are planned? The number is important for router utilization sizing.
- How may LUs are in place? How many are planned? Many busy LUs attached to a PU will increase link utilization.

Step 2 Determine current host configurations. Important information includes:

- Network Control Program (NCP) definition listing if 3745, 3725, or 3720; in particular GROUP, LINE, PU, and LU definition statements
- Remote controller configuration worksheets for 3x74, 5X94
- OS/2 Communication Manager configuration files
- Network topology diagram

Step 3 Determine what router-based IBM features will best suit your requirements:

- If remote devices are SDLC-attached PU type 2 devices, consider SDLLC. Refer to "SDLLC Implementation" following.
- Depending on the specific situation, STUN can be used in many instances and supports all PU types.

Step 4 Determine what FEP-to-NCP conversion changes are required:

- Are FEP lines multidrop drops? Is virtual multidrop required? Refer to "Virtual Multidrop" earlier in this chapter.
- Do PU addresses require changing if SDLC address prioritization is used? Refer to "SDLC Address Prioritization" earlier in this chapter.
- Does the reply timeout T1 timer need to be increased to accommodate network delays if local acknowledgment is not used?
- Does the "Retries" parameter need to be adjusted for longer elapsed retry sequences?

Step 5 Determine how end-station controllers are configured and, if possible, configure the router to accommodate them:

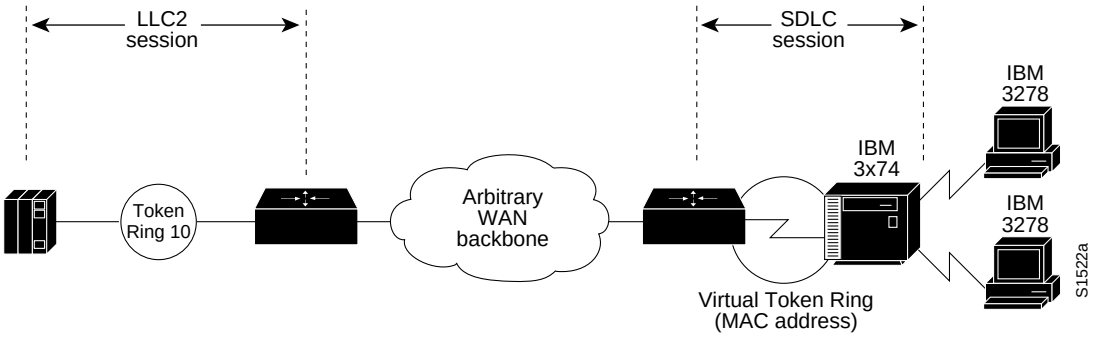
- Addresses might need changing if you use virtual multidrop. Refer to "Virtual Multidrop" earlier in this chapter.
- NRZ support might be required depending on router platform and interface used. Refer to "Supported Data Link Configurations" earlier in this chapter.
- If the controller toggles RTS (assumes half duplex mode), refer to "Supported Data Link Configurations" earlier in this chapter.

SDLLC Implementation

The SDLLC function allows serial-attached devices using the SDLC protocol to communicate with LAN-attached devices using the LLC2 protocol. The basic purpose of the SDLLC function is to consolidate the traditionally disparate SNA/SDLC networks onto a LAN-based, multiprotocol, multimedia backbone network.

Using the SDLLC feature, routers terminate SDLC sessions, translate SDLC to the LLC2 protocol and forward the LLC2 traffic through remote source route bridging (RSRB) over a point-to-point or IP network. Because a router-based IP network can use any arbitrary media such as FDDI, Frame Relay, X.25, or leased lines, routers support SDLLC over all such media through IP encapsulation. Figure 4-10 illustrates a general SDLLC media translation internetwork arrangement.

Figure 4-10 SDLLC Media Translation



Note In Figure 4-10, the Token Ring connection (Token Ring 10) could also be an Ethernet segment that connects the FEP or 3172 and router.

SDLLC Configuration

The following sections provide design and implementation information for the following SDLLC-related configuration topics:

- Local Acknowledgment
- Multidrop Access
- Router Configuration
- Encapsulation Overhead

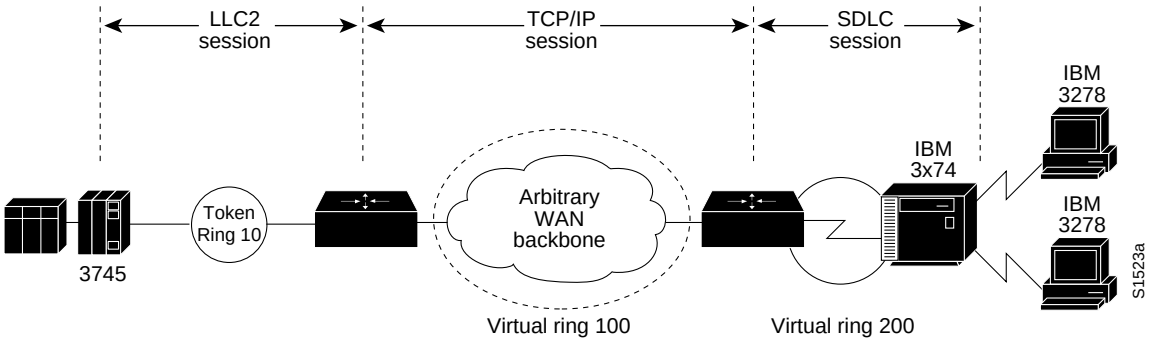
Local Acknowledgment

Local acknowledgment of LLC2 sessions allows frames to be locally terminated by the Token Ring-attached router, which guarantees delivery to the ultimate destination through the reliable transport services of TCP/IP. By locally terminating LLC2 sessions, packet reception is locally acknowledged, acknowledgment and keepalive traffic is prevented from traversing the backbone, and SNA sessions are preserved in the event of network failure. The SDLC session in an SDLLC environment is always acknowledged by the router that is performing the media translation.

Local acknowledgment locally terminates Supervisory frames, which include Receiver Ready, Receiver Not Ready, and Reject.

Figure 4-11 illustrates the operation of local acknowledgment.

Figure 4-11 Local Acknowledgment Operation



Note Local acknowledgment requires that TCP sessions be maintained between the routers. It is not uncommon to see high router CPU utilization at idle traffic times and then decreased utilization as traffic increases. Polling overhead in the router may drive processor use up.

Multidrop Access

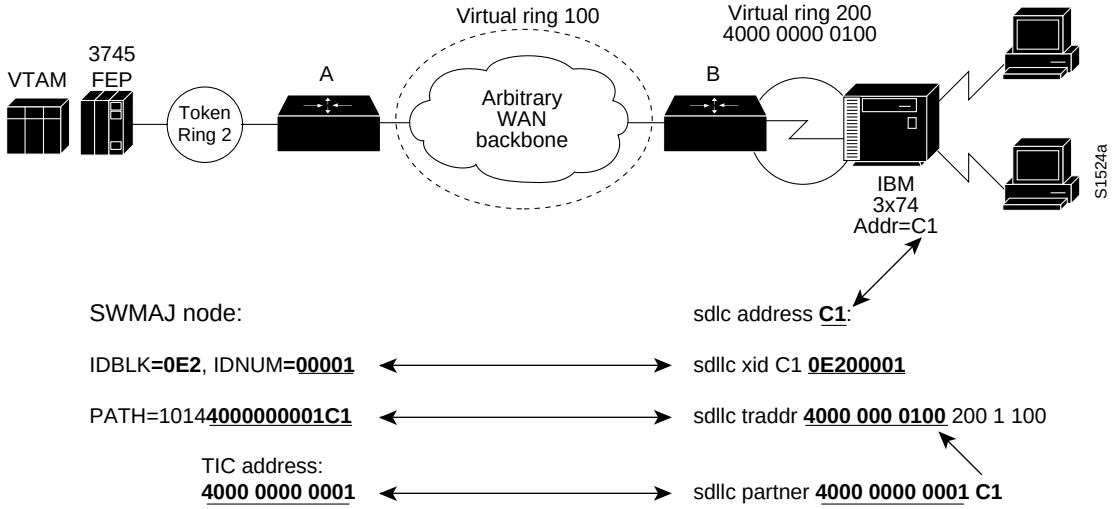
Two approaches can be taken to configure multidrop operation for the SDLC link in an SDLLC environment. First, by using a line-sharing device or a modem-sharing device (MSD), you can connect multiple controllers at a single site to a single SDLC port on the router. The second approach is to connect multiple controllers at different sites through a multidrop service provided by a telephone company. For more information about multidrop connections, refer to Appendix B, "IBM Serial Link Implementation Notes."

When designing a multidrop environment, you should consider line speed, link utilization, and the number of controllers that will share a single line. In addition, consider the number of attached LUs associated with individual PUs, and determine if these LUs are being heavily used. If so, increase the bandwidth of the attached serial line. When implementing multidrop environments featuring large numbers of PUs and LUs, contact your technical support representative for specific capabilities.

Router Configuration

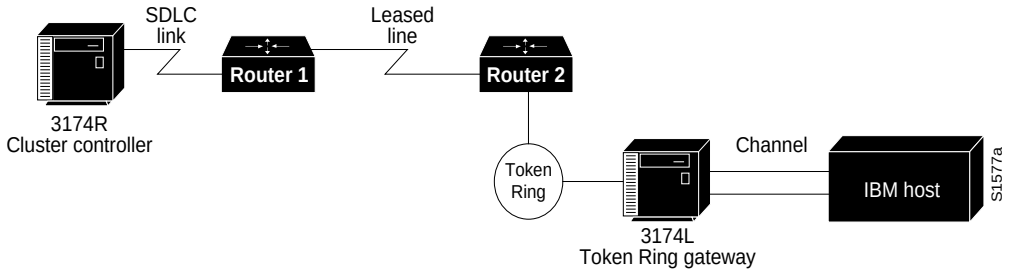
To configure a router for SDLLC, you need certain virtual telecommunications access method (VTAM) and NCP definition statements to configure the router properly. Figure 4-12 illustrates the required configuration information.

Figure 4-12 Required End-to-End SDLLC Information



Consider an example of two routers that implement the SDLLC functionality in an environment that interconnects a remote site to a host channel attached to a 3174 Token Ring gateway, as shown in Figure 4-13.

Figure 4-13 SDLLC Implementation with 3174 Token Ring Gateway



Note Routers also support SDLLC implemented in environments with a 3745 Token Ring gateway.

In the example network illustrated in Figure 4-13, the following conditions apply:

- The SDLC address of the 3174R is C1.
- The device called 3174L is a 3174 Token Ring that is channel-attached to an IBM mainframe.

The 3174R must be defined in the configuration of the 3174L using the virtual Token Ring MAC address. This address is created in the router configuration; it includes the SDLC address as the last byte. This virtual MAC address is mapped to a host subchannel address. One host subchannel address is assigned for each downstream physical unit at host system generation time. PU and LU functions are defined to VTAM within the switched major node function.

On Router 1, the following configuration commands are required:

- The **sdllc traddr** interface configuration command with a virtual ring address for the 3174R. Note that the last byte must be 00 and that you must specify the appropriate SDLC address (in this case, C1) for the same last byte during the 3174L customization.
- The **sdllc partner** interface configuration command with the MAC address of the 3174L gateway and the SDLC address of the 3174R.
- The following version of the **sdllc xid** interface configuration command:

  ```
  sdllc xid c1 00000000
  ```

  The **sdllc xid** interface configuration command is specified with all zeros in the IDBLK/IDNUM field to establish the LLC session between Router 1 and the 3174L. All zeros in the node ID field of the XID command indicates that there is no unique node identifier in this field.

Encapsulation Overhead

Cisco routers provide several types of encapsulation solutions. Because encapsulation always incurs a certain amount of overhead, you need to assess the advantages and performance trade-offs of each encapsulation solution within the constraints of your environment.

Because TCP/IP encapsulation is very robust, provides a high quality of service, and is media independent, it is recommended most frequently. If SDLLC local acknowledgment is required, TCP/IP encapsulation is required. If SDLLC local acknowledgment is not required, Fast-Sequenced Transport (FST) encapsulation is highly recommended because it is less CPU intensive.

Direct High-Level Data Link Control (HDLC) encapsulation can only be used in point-to-point environment. FST and direct HDLC encapsulation are comparable in performance, but FST has more overhead, which may be an issue on slow speed serial links. TCP/IP encapsulation has the most overhead in terms of processor utilization and the WAN connection. If TCP/IP encapsulation with header compression is a requirement, use it only on link speeds of 64 kbps or less.

Table 4-6 outlines encapsulation overhead for SDLLC and RSRB implementations.

Table 4-6 SDLLC and RSRB Encapsulation Overhead

TCP/IP	FST	TCP/IP with Header Compression	HDLC
CPU intensive	Less CPU intensive	Very CPU intensive	Least CPU intensive
4 bytes for HDLC	4 bytes for HDLC	4 bytes for HDLC	4 bytes for HDLC
20 bytes for IP	20 bytes for IP	3–8 bytes for TCP/IP	16 bytes for virtual ring
20–24 bytes for TCP	16 bytes for virtual ring	16 bytes for virtual ring	
16 bytes for virtual ring			
Total: 60–64 bytes	Total: 40 bytes	Total: 23–28 bytes	Total: 20 bytes

SDLLC Guidelines and Recommendations

The following suggestions should help you improve resource response time and network performance:

- Token Ring frame size—Allow the Token Ring Interface Coupler (TIC) FEP to send as large a frame as is possible and let the router segment the frame into multiple SDLC Information frames.
- MAXOUT (window size)—Change the MAXOUT value in the VTAM-switched major node for the 3174 PU. MAXOUT is IBM's terminology for *window size*. IBM recommends setting window sizes on LAN-attached devices to 1 because their tests found no performance benefit with a larger window size. The *red books* published by the IBM International Systems Center and widely used as installation guides by IBM SEs and customers show examples with MAXOUT=1. Because the remote device is an SDLC-attached 3x74, not a Token Ring-attached device, changing MAXOUT to 7 can make a dramatic improvement in performance.
- SDLC line speed—Increase the line speed of the 3x74 to 19.2 kbps (older units) or 64 kbps (newer units) when the controller is directly attached (as opposed to attachment through a modem) to the router. Modem and communication facilities are frequently the limiting factors in determining the line speed in the prerouter configuration.
- SDLC frame size—Set the largest SDLC frame size to 521 on newer 3274 models (not 265, which is required for older 3274 models). *See the note that follows.*
- Request To Send (RTS) control—Set the 3174 for permanent RTS if the device is not connected via a multidrop service through modem-sharing devices or line-sharing devices. Modem-sharing and line-sharing connections require that RTS be toggled when the device is transmitting. Setting permanent RTS cuts down on line turnaround delays and can improve link utilization by 10 percent. (However, setting permanent RTS is unlikely to achieve any perceptible response time improvements.)

Note Changing configurations of end devices such as terminal controllers is often considered an undesirable activity. The high number of devices requiring changes and the cost and unavailability associated with these changes can make these modifications onerous. Modify SDLC maximum frame size and RTS control with discretion.

SDLLC Implementation Scenarios

The following case study shows how an internetwork can evolve from a SNA-specific SDLC environment featuring 3x74 controllers and 3270 terminals to a network consisting of PCs with client/server applications. The most important requirement for this evolution is the protection of existing SNA investment.

Assume that the original network consisted of hundreds of SDLC 3x74 controllers connected to a number of 37x5 FEPs in the data center. A disaster recovery center maintains the "mirror-image" of the data center. Many of 3x74s are multidrop connected to the host via 9.6- or 19.2-kbps leased lines. The challenges facing the corporate MIS organization for this internetwork include the following:

- Reliability—When an SDLC line goes down, all the downstream users are affected. There is no network redundancy.
- Leased line charges—Providing lines to multiple remote SDLC devices results in excessive service charges and must be minimized.

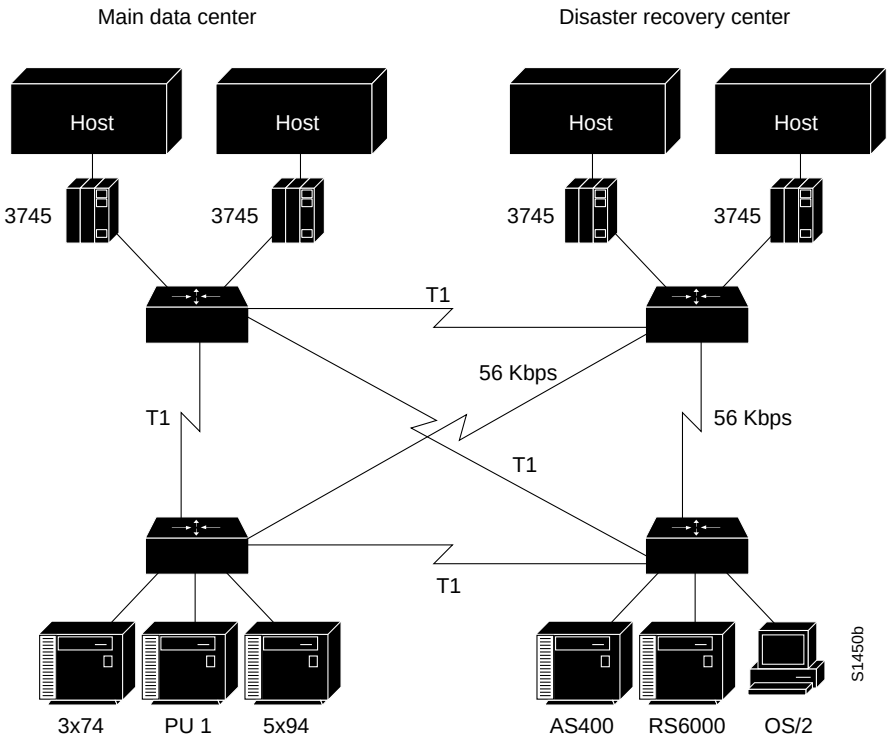
- FEP CPU use—CPU use is higher for SDLC-supported sessions than for LAN-supported sessions.
- Maintaining VTAM and NCP—Every move and change requires system programmers to regenerate VTAM/NCP, which increases the cost of maintaining a statistically defined network.
- Support of LAN-based applications—There is a growing need to support LAN-based interconnection, both PC-to-host and PC-to-PC.
- Availability and uptime—To maintain a competitive advantage, the organization needs to keep SNA sessions alive even in the event of network failure.

A phased strategy aimed at addressing these challenges would consist of three phases. Each of these phases is discussed in the following implementation examples.

Phase 1: Redundant Backbone Using STUN and Virtual Multidrop

Build a redundant backbone network with routers and high-speed E1 or T1 links in each regional office, as shown in Figure 4-14. Connect multiple SDLC devices to a router via SDLC transport with virtual multidrop. The resulting network increases reliability and minimizes leased-line charges.

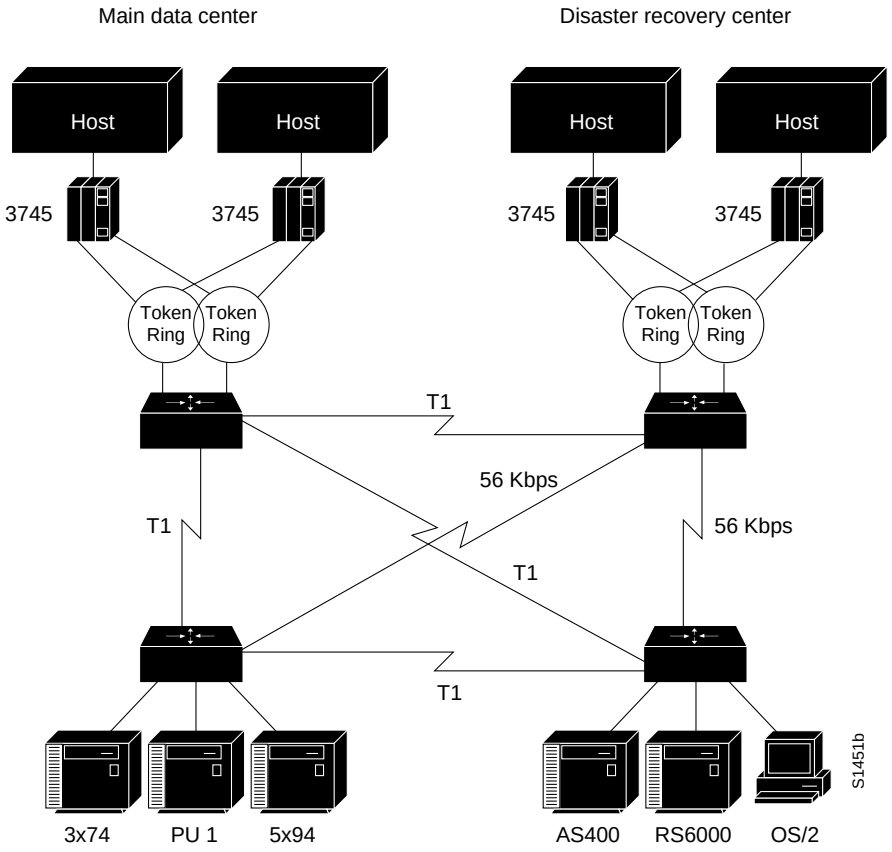
Figure 4-14 Connecting Multiple SDLC Devices via SDLC Transport with Virtual Multidrop



Phase 2: Fault-Tolerant Host FEP Token Ring and SDLLC Implementation

Implement a fault-tolerant host FEP Token Ring, as shown in Figure 4-15. Connecting existing SDLC devices to the host Token Ring via SDLLC results in improved response time. Because SDLC devices appear as Token Ring-attached devices to the host, you do not need to regenerate NCP and reload when adding or changing PUs and LUs. This can be done dynamically through VTAM-switched major nodes. This implementation also reduces FEP CPU utilization.

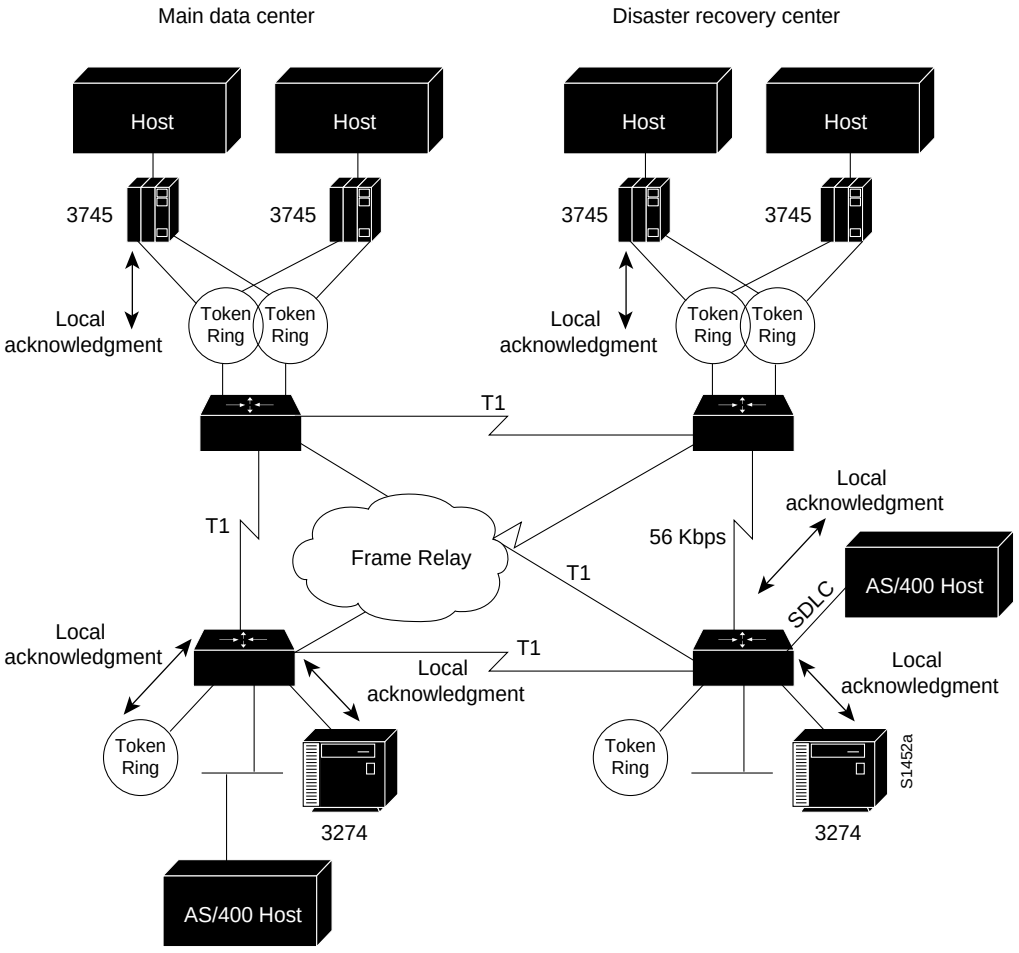
Figure 4-15 Fault-Tolerant TICs and SDLLC Implementation



Phase 3: Strategic LAN-to-WAN Implementation

Implement LAN (both Token Ring and Ethernet) internetworks in selected locations along with alternative WAN technologies such as Frame Relay, as shown in Figure 4-16. Connect LAN-based and remote SDLC devices to host FEP Token Ring via SDLLC, RSRB, and translational bridging, and to host FEP SDLC via reverse SDLLC (SDLC side primary). SNA session integrity is maintained through local termination of both LLC2 and SDLC traffic. These solutions provide needed support of LAN-based applications and provide improved availability and uptime for SNA network devices.

Figure 4-16 Implementing Alternative LAN-to-WAN Technologies for an Integrated Solution



SDLLC Implementation Checklist

Before implementing an SDLLC-based internetwork, make sure you are familiar with information in the publications *Router Products Configuration Guide* and *Router Products Command Reference* that deal with SDLC. Depending on your implementation, you might need to review the sections "SDLLC Configuration" and "SDLLC Implementation Scenarios" earlier in this chapter.

In general, the following guidelines help you create a working, manageable network:

- Use a phased approach to implement your router network.
- Establish a test environment to initially bring up the routers.
- Plan a gradual cutover of end devices into the production environment.
- During the cutover period, observe the router's behavior using **show** commands.

You should strive to create a network that has predictable behavior early on. This can prevent problems from happening as more and more devices are brought on line.

The following is a specific SDLLC implementation checklist that you can use to identify technologies, implementations, and possible solutions for your internetwork:

Step 1 Evaluate the customer requirements for SDLLC support:

- Identify all host-attached controllers. Examples include 37x5, 3172, and 3174 devices. The host sites might be referred to as local, core, backbone, or data center sites.
- How are the host site controllers connected to the network?
- Is Token Ring already in place? Ethernet?
- Determine link speeds for remote end systems.
- Determine current line utilization measurements. Network Performance Monitor is typically installed in the larger SNA shops where it makes historical data available.
- What interface type is required? For example: V.24 (EIA/TIA-232, formerly RS-232), V.35, or X.21.
- What modems, data service units, channel service units, modem-sharing devices or line-sharing devices will be used?
- Is Link Problem Determination Aid (LPDA) support required? LPDA is a feature of IBM modems and data service units that reports line quality and statistics to NetView. LPDA version 1 is not compatible with STUN and SDLLC; LAPD Version 2 may be compatible with STUN.
- What remote end-system types are expected? Examples include 3174, 3274, and AS/400.
- Will there be end-system emulation?
- What is the current transaction response time? Is subsecond response required?
- Number of PUs (this will be important for router utilization sizing).
- Number of LUs (many busy LUs attached to a PU will increase link utilization).

Step 2 Determine current configuration. Important information includes the following:

- NCP system generation for 3745, 3725, and 3720 devices; in particular, note the LINE, PU, and LU definition statements.
- Local controller current worksheets for 3174, 3172 devices.
- Remote controller configuration worksheets for 3x74 and 5x94 devices.

- OS/2 Communication Manager configuration files.
- Network topology diagram.

Step 3 Determine the SDLLC features that best suit your requirements.

Confirm that devices to be attached are SDLC PU type 2 devices. Select specific feature requirements, such as local acknowledgment and virtual multidrop.

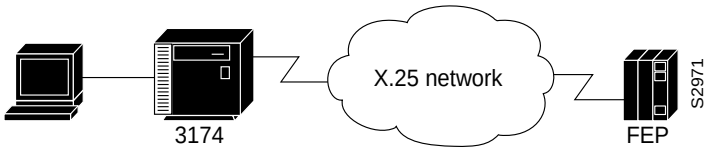
Step 4 Determine what host conversion changes are required:

- Switched major node definitions for VTAM
- FEP/NCP changes for Token Ring addition and SDLC link reduction

QLLC Conversion

QLLC is a data-link protocol defined by IBM that allows SNA data to be transported across X.25 networks. With QLLC, each SDLC physical link is replaced by a single virtual circuit. Figure 4-17 illustrates a typical QLLC topology. In this topology, both ends of the connection over the X.25 network must be configured for QLLC.

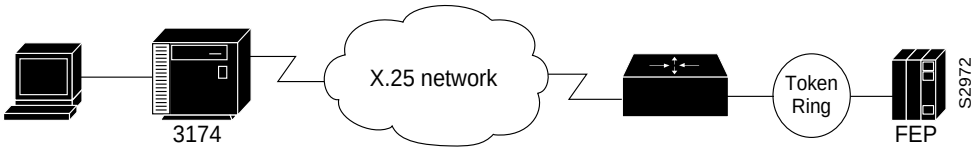
Figure 4-17 Typical QLLC Topology



QLLC conversion is an IOS Release 10.2 feature that causes the router to perform all of the translation required to send SNA data over an X.25 network so that IBM devices that are connected to a router do *not* have to be configured for QLLC.

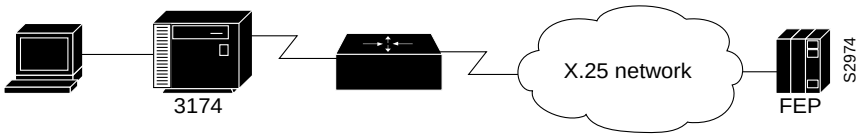
QLLC conversion allows a device (typically a FEP or an AS/400) that is attached directly to the router or through a Token Ring to communicate with a device (typically a 3174 terminal controller) that is attached to an X.25 network, as shown in Figure 4-18. In this example, only the terminal controller must be configured for QLLC and have an X.25 interface.

Figure 4-18 Simple Topology for QLLC Conversion



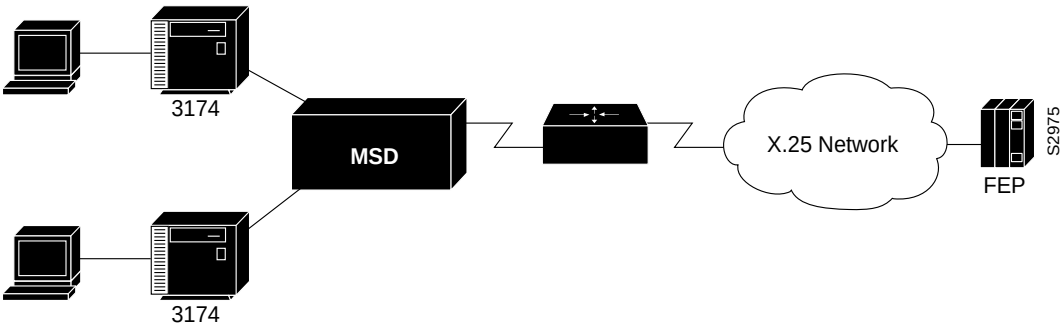
In some topologies, one router interface uses SDLC to communicate with the terminal controller, and another router interface uses X.25 to communicate with the remote device over the X.25 network. In Figure 4-19, the router, configured for QLLC conversion, handles SNA traffic between the terminal controller and the FEP.

Figure 4-19 Topology that Uses SDLC and QLLC Conversion



QLLC conversion also supports multiple SDLC connections coming through an MSD, as shown in Figure 4-20.

Figure 4-20 QLLC Conversion Supports Multidrop SDLC Topology



The router that is configured for QLLC conversion does not need to be on the same Token Ring as the FEP. In Figure 4-21, Router A is configured for QLLC and remote source-route bridging (RSRB), and Router B is configured for RSRB only. RSRB allows the FEP to connect to Router A. If a Token Ring connected to the X.25 network communicates with the Token Ring attached to the FEP by a protocol other than SRB, RSRB can provide connectivity.

Figure 4-21 Complex QLLC Conversion Topology

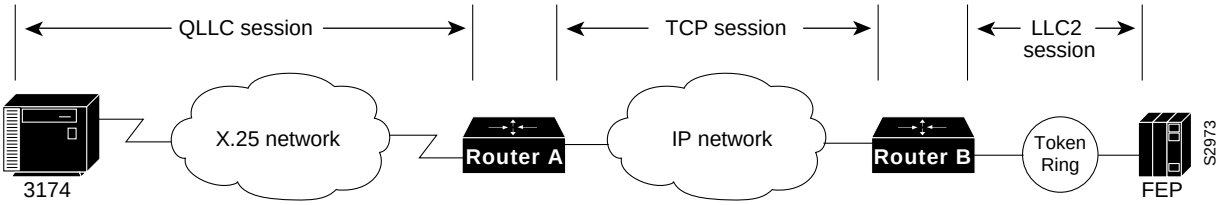
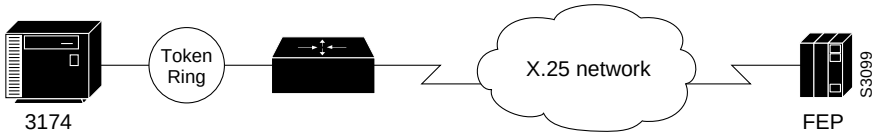


Figure 4-21 shows an example using local acknowledgment, which causes the LLC2 session from the Token Ring-attached SNA device (the FEP) to be terminated at the adjacent router (Router B). A TCP session transports the data from Router B to the router attached to the X.25 network (Router A). Only Router A is configured for QLLC conversion. When enabled, local acknowledgment applies to all QLLC connections. The **source-bridge qllc-local-ack** global configuration command enables local acknowledgment and applies to all QLLC connections.

In pass-through mode, local acknowledgment is not used. Instead, the LLC2 session from the Token Ring-attached SNA device (the FEP) is terminated at the router connected to the X.25 network (Router A).

QLLC conversion also supports a configuration in which SNA end stations (3174 or equivalent) connected to a Token Ring reach the FEP through an X.25 connection, as shown in Figure 4-22. In this case, IBM Network Packet Switching Interface (NPSI) software is installed on the FEP.

Figure 4-22 QLLC Conversion Supports SNA End Station Connections over Token Ring and X.25 Networks



CHAPTER 5

Designing ATM Internetworks

Asynchronous Transfer Mode (ATM) is an evolving technology designed for the high-speed transfer of voice, video, and data through public and private networks in a cost-effective manner. ATM is based on the efforts of Study Group XVIII of the International Telecommunication Union Telecommunication Standardization Sector (ITU-T, formerly the Consultative Committee for International Telegraph and Telephone [CCITT]) and the American National Standards Institute (ANSI) to apply very large scale integration (VLSI) technology to the transfer of data within public networks. Officially, the ATM layer of the Broadband Integrated Services Digital Network (BISDN) model is defined by CCITT I.361. Current efforts to bring ATM technology to private networks and to guarantee interoperability between private and public networks is being done by the ATM Forum, which was jointly founded by Cisco Systems, NET/ADAPTIVE, Northern Telecom, and Sprint in 1991.

This chapter describes current ATM technologies that network designers can use in their networks today. It also makes recommendations for designing non-ATM networks today that those networks can take advantage of ATM in the future without sacrificing current investments in cable.

This chapter focuses on the following topics:

- ATM Data Exchange Interface
- ATM Interface Processor Card
- Cisco HyperSwitch A100
- ATM Media

ATM Overview

ATM combines the strengths of time-division multiplexing (TDM)—whose fixed time slots are used by telephone companies to deliver voice without distortion—with the strengths of packet-switching data networks—whose variable size data units are used by computer networks, such as the Internet, to deliver data efficiently. While building on the strengths of TDM, ATM avoids the weaknesses of TDM (which wastes bandwidth by transmitting the fixed time slots even when no one is speaking) and PSDNs (which cannot accommodate time-sensitive traffic, such as voice and video, because PSDNs are designed for transmitting bursty data). By using fixed sized cells, ATM combines the isochronicity of TDM with the efficiency of PSDN.

ATM Cell Format

ATM cells consist of 5 bytes of header information and 48 bytes of payload data. Two fields in the ATM header are used to route cells through ATM networks:

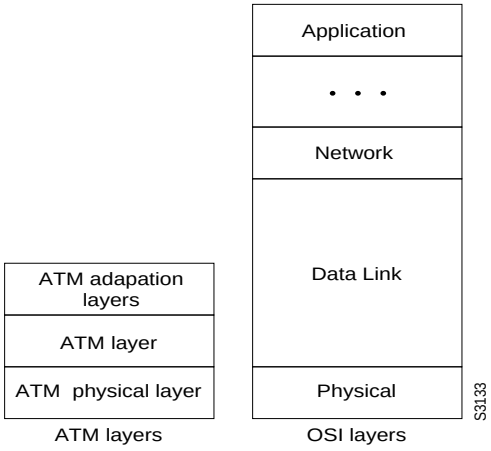
- VPI—The virtual path identifier.
- VCI—The virtual channel identifier.

The VPI and VCI fields of the cell header identify the next network segment that a cell needs to transit on its way to its final destination. A *virtual channel* is equivalent to a virtual circuit—that is, both terms describe a logical connection between the two ends of a communications connection. A *virtual path* is a logical grouping of virtual circuits that allows an ATM switch to perform operations on groups of virtual circuits.

ATM Functional Layers

ATM consists of three functional layers—the ATM physical layer, the ATM layer, and the ATM adaptation layer—that correspond roughly to layer 1 and parts of layer 2 (such as error control and data framing) of the *Open System Interconnection* (OSI) reference model, as shown in Figure 5-1.

Figure 5-1 Relationship of ATM Functional Layers to the OSI Reference Model



The ATM physical layer controls transmission and receipt of bits on the physical medium. It also keeps track of ATM cell boundaries and packages cells into the appropriate type of frame for the physical medium being used. Above the physical layer is the *ATM layer*, which is responsible for establishing virtual connections and passing ATM cells through the ATM network. Immediately above the ATM layer is the *ATM adaptation layer*, which translates between the larger *service data units* (SDUs) (for example, video streams, and data packets) of upper-layer processes and ATM cells. Several ATM adaptation layers are currently specified. They are analyzed later in this chapter. Cisco routers currently support ATM adaptation layer 5 (AAL5), which is used to transfer most non-SMDS data, such as classical IP over ATM and local-area network (LAN) emulation, and AAL3/4 for SMDS traffic.

ATM Addressing

The ATM Forum has adapted the subnetwork model of addressing in which the ATM layer is responsible for mapping network layer addresses to ATM addresses. Several ATM address formats have been developed. Public ATM networks will typically use E.164 numbers, as also used by Narrowband ISDN (N-ISDN) networks.

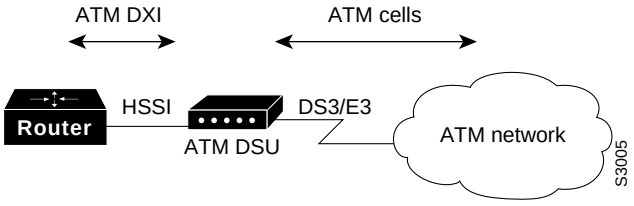
The ATM address formats are modeled on ISO Network Service Access Point (NSAP) addresses, but they identify SubNetwork Point of Attachment (SNPA) addresses. Incorporating the MAC address into the ATM address makes it easy to map ATM addresses into existing LANs.

Note For overview information about ATM, see the *Internetworking Technology Overview* publication.

ATM Data Exchange Interface

To make ATM functionality available as soon as possible, the ATM Forum developed a standard known as the *ATM Data Exchange Interface* (DXI). Network designers can use DXI to provide User-Network Interface (UNI) support between Cisco routers and ATM networks, as shown in Figure 5-2.

Figure 5-2 ATM DXI Topology



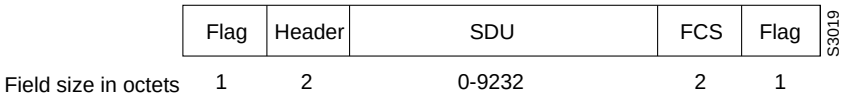
The ATM data service unit (ADSU) receives data from the router in ATM DXI format over a High-Speed Serial Interface (HSSI). The DSU converts the data into ATM cells and transfers them to the ATM network over a DS-3/E3 line.

ATM DXI is available in several modes:

- Mode 1a—Supports AAL5 only, a 9232 octet maximum, and a 16-bit FCS, and provides 1023 virtual circuits.
- Mode 1b—Supports AAL3/4 and AAL5, a 9224 octet maximum, and a 16-bit FCS. AAL5 support is the same as Mode 1a. AAL3/4 is supported on one virtual circuit.
- Mode 2—Supports AAL3/4 and AAL5 with 16,777,215 virtual circuits, a 65535 octet maximum, and 32-bit FCS.

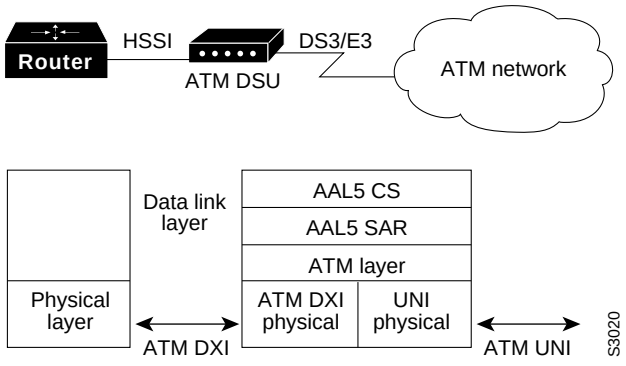
On the router, data from upper-layer protocols is encapsulated into ATM DXI frame format. Figure 5-3 shows the format of a Mode 1a ATM DXI frame.

Figure 5-3 ATM DXI Frame Format



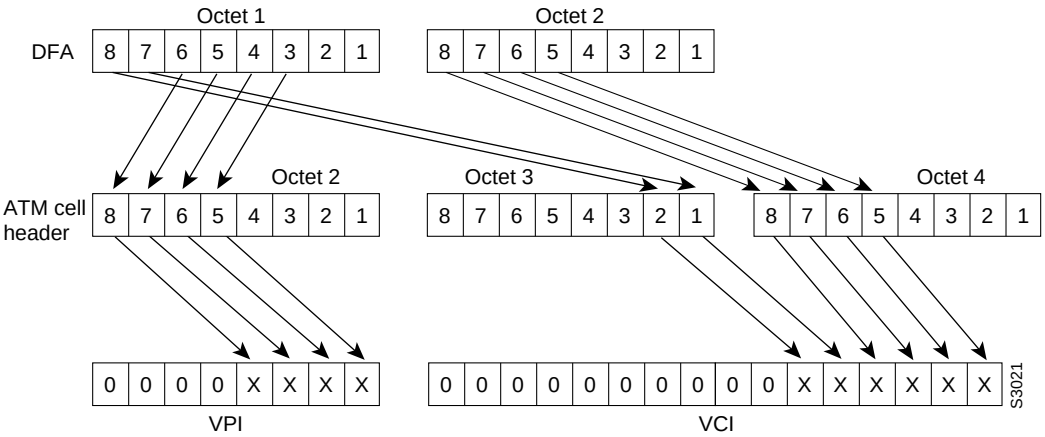
In Figure 5-4, a router configured as a data terminal equipment (DTE) device is connected to an ADSU. The ADSU is configured as a data communications equipment (DCE) device. The router sends ATM DXI frames to the ADSU, which converts the frames to ATM cells by processing them through the AAL5 convergence sublayer (CS) and the segmentation and reassembly sublayer (SAR). The ATM layer attaches the header, and the cells are sent out the ATM UNI interface.

Figure 5-4 ATM DXI Mode 1a and Mode 1b Protocol Architecture for AAL5



ATM DXI addressing consists of a DFA, which is equivalent to a Frame Relay data link connection identifier (DLCI). The DSU maps the DFA into appropriate VPI and VCI values in the ATM cell. Figure 5-5 shows how the DSU performs address mapping.

Figure 5-5 ATM DXI Address Mapping



Note ATM DXI 3.2 is supported in Software Release 9.21 and subsequent software releases. Mode 1a is the only mode supported.

ATM Interface Processor Card

The ATM interface processor (AIP) card supports native ATM in Cisco 7000 and Cisco 7010 routers that are running IOS Release 10.0 or later.

Note IOS Release 10.0 supports AAL5 permanent virtual circuits (PVCs) only.

IOS 10.0 supports ATM Forum UNI Specification V3.0, which includes the user-to-network ATM signaling specification. The AIP card uses RFC 1483 (Multiprotocol Encapsulation over AAL5) to transport data through an ATM network. RFC 1483 specifies the use of an LLC/SNAP 8-byte header to identify the encapsulated protocol. It also specifies a null encapsulation (VC Mux) which, instead of headers, creates a separate virtual circuit per protocol.

The AIP supports the following protocols:

- AppleTalk
- Banyan Virtual Network System (VINES)
- Connectionless Network Service (CLNS)
- DECnet
- Internet Protocol (IP)
- Novell Internetwork Packet Exchange (IPX)

The following physical layer interface modules (PLIMs) are available for the AIP:

- TAXI 4B/5B 100-megabits-per-second (Mbps) multimode fiber-optic cable
- SONET/SDH 155-Mbps fiber-optic (STS-3c or STM1) cable
- SONET/SDH 155-Mbps single-mode fiber-optic (STS-3c or STM1) cable
- E3 34-Mbps coaxial cable
- DS-3 45-Mbps cable

The total bandwidth though all the AIPs configured in a router should be limited to 200 Mbps full duplex. For that reason, only the following combinations are supported:

- 2 TAXI interfaces
- 1 SONET and one E3 interface
- 2 SONET interfaces, one of which is lightly used
- 5 E3 interfaces

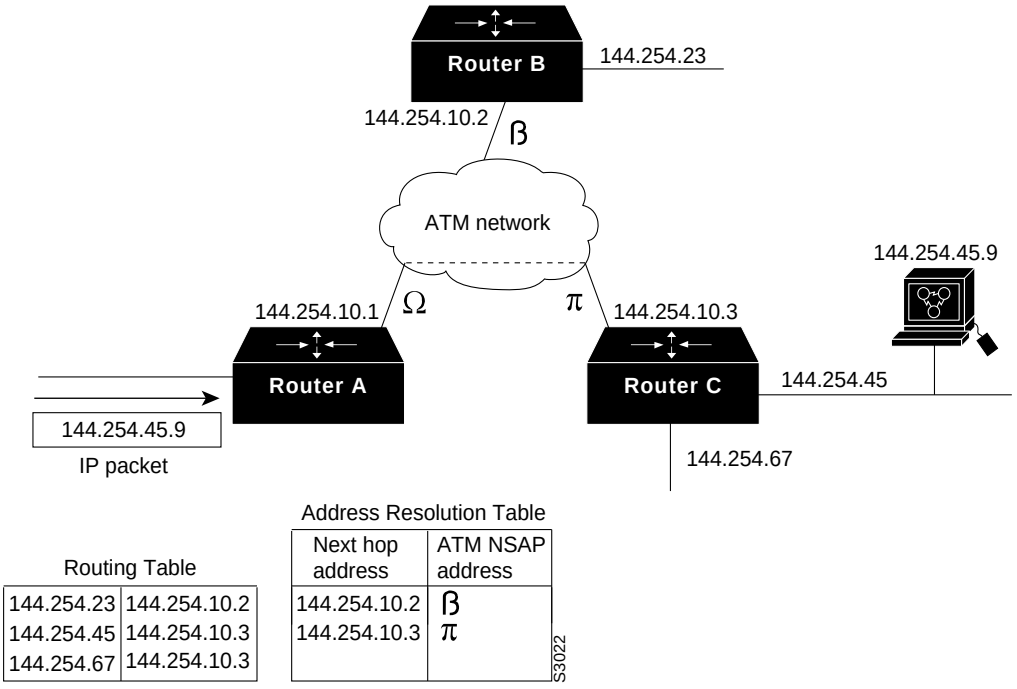
The AIP includes hardware support for various traffic shaping functions. Virtual circuits can be assigned to one of eight rate queues, each of which is programmable for a different peak rate. Each virtual circuit can be assigned an average rate and specific burst size. The signaling request specifies the size of the burst that will be sent at the peak rate, and after that burst, the rest of the data will be sent at the average rate.

Following are the configurable traffic parameters on the AIP:

- Forward peak cell rate
- Backward peak cell rate
- Forward sustainable cell rate
- Backward sustainable cell rate
- Forward maximum burst
- Backward maximum burst

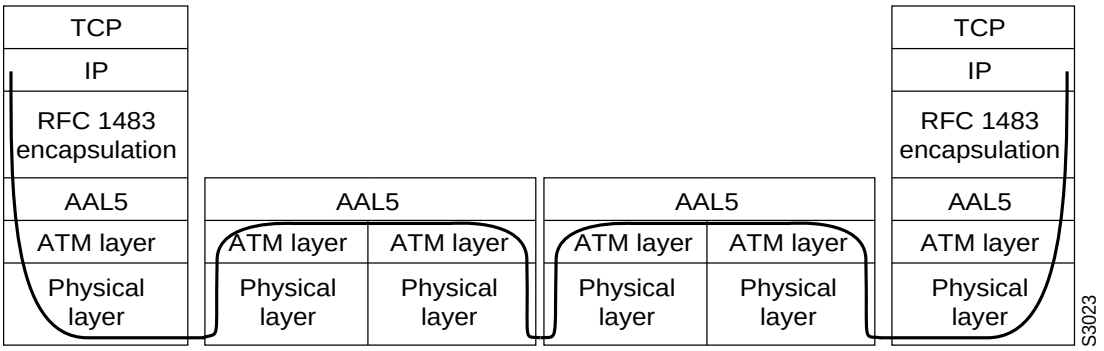
Figure 5-6 shows how the routing table and address resolution table on Router A are used to forward data to a workstation behind Router C.

Figure 5-6 AIP Connects LANs to ATM Fabric



The routing table on Router A performs its usual function of determining the next hop by mapping the network number of the destination (in this case 144.254.45 from the incoming packet) to the IP address of the router to which the destination network is connected (in this case, 144.254.10.3, which is the IP address of Router C). An address resolution table maps the next-hop IP address to an ATM NSAP address (in this case, represented by ¼). Router A signals Router C over the ATM network to establish a virtual connection, and Router A uses that connection to forward the packet to Router C. Figure 5-7 shows the layers through which the packet travels.

Figure 5-7 Path of an IP Packet over the ATM Fabric



Configuring the AIP for ATM Signaling

The following commands configure an AIP for ATM signaling:

```
interface atm 4/0
ip address 128.24.2.1 255.255.255.0
no keepalive
atm nsap-address AB.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12
atm pvc 1 0 5 qsaal
map-group shasta
atm rate-queue 0 155
atm rate-queue 1 45

map-list shasta
ip 144.222.0.0 atm-nsap BB.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12
ip 144.3.1.2 atm-nsap BB.CDEF.01.234567.890A.BCDE.F012.3456.7890.1234.12 class QOSclass

map-class QOSclass
atm forward-peak-cell-rate-clp0 15000
atm backward-max-burst-size-clp0 96
```

The following explains relevant portions of the ATM signaling configuration:

- **no keepalive**—Required because IOS Release 10.0 does not support the Interim Local Management Interface (ILMI), an ATM Forum specification.
- **atm nsap-address**—Required for signaling.
- **atm pvc**—Sets up a PVC to carry signaling requests to the switch. In this case, the command sets up a circuit whose VPI value is 0 and whose VCI value is 5, as recommended by the ATM Forum.
- **map-group**—Associates a map list named **shasta** to this interface.
- **atm rate-queue**—Set up two rate queues. Rate queue number 0 is for 155 Mbps transfers, and rate queue number 1 is for 45-Mbps transfers.
- **map-list** and **ip 144.222.0.0**—Set up the static mapping of an IP network number to an ATM NSAP address without any QOS parameters. The **ip 144.3.1.2** command maps an IP host address to an ATM NSAP address with the QOS parameters specified in the map class named **QOSclass**.
- **map-class**, **atm forward-peak-cell-rate-clp0**, and **atm backward-max-burst-size-clp0**—Set up QOS parameters associated with this connection. The connection must support a forward peak cell rate of 15 kbps and a backward burst size of 96 cells.

Interoperability with DXI

When configuring an AIP to communicate with a Cisco router that uses ATM DXI to connect to the ATM network, the AIP requires Network Layer Protocol Identifier (NLPID) encapsulation, which is provided in IOS Release 10.2, or the ATM DXI requires LLC/SNAP encapsulation.

Cisco HyperSwitch A100

The Cisco HyperSwitch A100 is an ATM switch that provides up to 16 modular 155-Mbps ATM interfaces. Each ATM interface is capable of providing 4096 point-to-point connections. The A100 can also support up to 1024 point-to-multipoint connections.

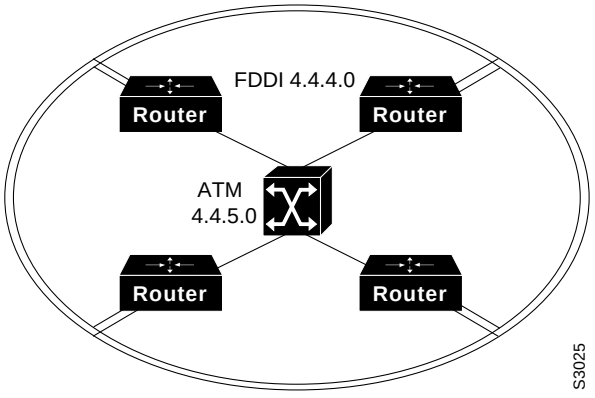
Table 5-1 Cisco HyperSwitch A100 Physical Layer Support

Physical Layer	Data Rate	Media	Connector
STS 3c/STM1	155 Mbps	Multimode fiber	SC
TAXI 4B/5B	100 Mbps	Multimode fiber	MIC (FDDI-style)
STS3c/STM1	155 Mbps	Single-mode fiber	ST
DS-3	45 Mbps	Coaxial cable	BNC
E3	34 Mbps	Coaxial cable	BNC

Single Switch Designs

Because ATM can use existing multimode fiber networks, Fiber Distributed Data Networks (FDDI) campus backbones can be easily upgraded from 100-Mbps FDDI to 155-Mbps point-to-point ATM. If the network has spare fiber, AIPs can be installed in each router and interconnected with a HyperSwitch A100, as shown in Figure 5-8. In this topology, each router has a 155-Mbps point-to-point connection to every other router on the ring.

Figure 5-8 Parallel FDDI and ATM Backbone



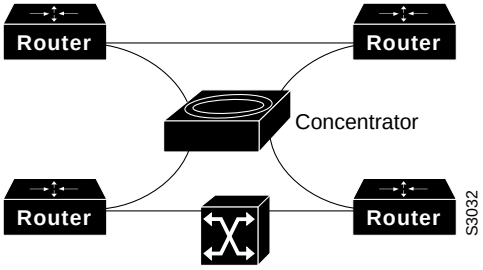
The addition of the ATM switch creates a parallel subnet. During the migration to ATM, a routing protocol, such as the Interior Gateway Routing Protocol (IGRP), can be used to force FDDI routing, as shown by the following commands:

```
interface fddi 1/0
ip address 4.4.4.1 255.255.255.0
interface atm 2/0
ip address 4.4.5.1 255.255.255.0
router igrp 109
network 4.4.0.0
distance 150 4.4.5.0 0.0.0.255
```

The **distance** command causes ATM to appear as a less desirable network and forces routing over FDDI.

If the network does not have spare fiber, a concentrator can be installed. Later, an ATM switch can be installed, as shown in Figure 5-9, which can be used to migrate ATM slowly throughout the network, using FDDI as a backup.

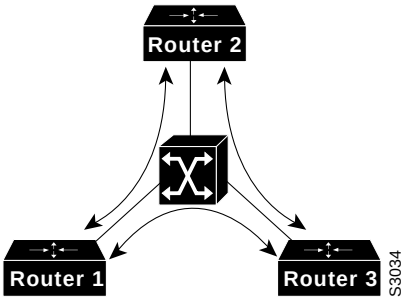
Figure 5-9 FDDI Topology with Concentrator and ATM Switch



Broadcasting in Single-Switch ATM Networks

There are two ways to configure broadcasting in a single-switch ATM network. First, the routers can be configured for “pseudo” broadcasting over point-to-point PVCs, as shown in Figure 5-10.

Figure 5-10 Router-Based “Pseudo” Broadcasting Using Point-to-Point PVCs



The following commands on each router set up a PVC between each router:

```
atm pvc 1 1 1 aal5snap
atm pvc 2 2 1 aal5snap
atm pvc 3 3 1 aal5snap
```

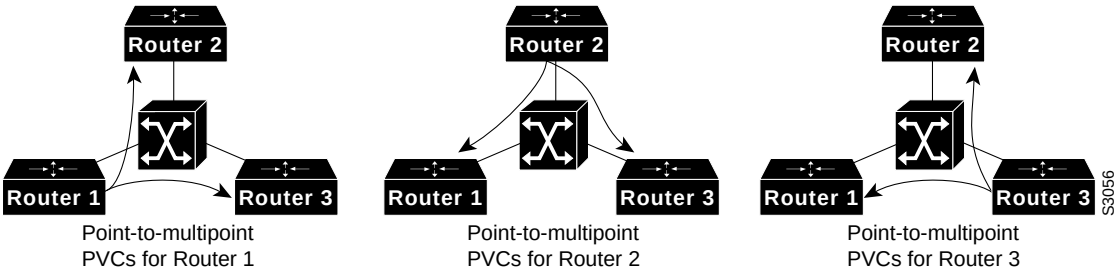
The following commands on each router cause that router to replicate broadcast packets and send them out on each PVC:

```
ip 4.4.5.1 atm-vc 1 broadcast
ip 4.4.5.2 atm-vc 2 broadcast
ip 4.4.5.3 atm-vc 3 broadcast
```

The disadvantage of router-based broadcasting is that it places the burden of replicating packets on the routers instead of on the switch, which has the resources to replicate packets at a lower cost to the network.

The second way to configure broadcasting is to configure the routers for switch-based broadcasting, as shown in Figure 5-11. With switch-based broadcasting, each router sets up a point-to-multipoint PVC to the other routers in the network. When each router maintains a point-to-multipoint PVC to every other router in the network, the broadcast replication burden is transferred to the switch.

Figure 5-11 Switch-Based Broadcasting



The following commands configure a point-to-multipoint PVC on each router:

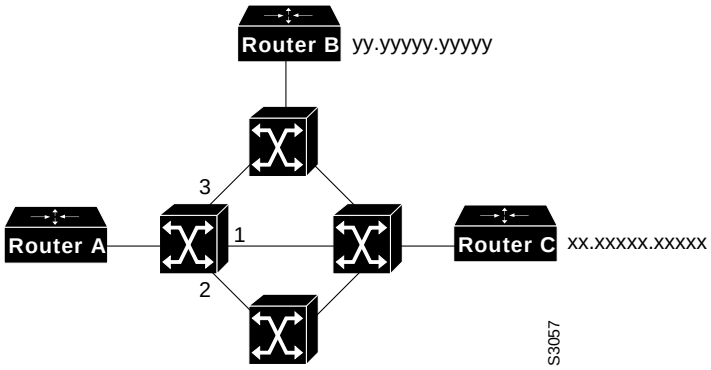
```
ip 4.4.4.1 atm-vc 1
ip 4.4.4.2 atm-vc 2
ip 4.4.4.3 atm-vc 3
ip 4.4.4.0 atm-vc broadcast
```

In Figure 5-11, the routers still have full mesh connectivity to every other router in the network, but the connections are not set up as broadcast PVCs. Instead, each router designates the point-to-multipoint PVC as a broadcast PVC and lets the switch handle replication, which is a function for which the switch is optimized.

Multiple-Switch Designs

The A100 supports the ATM Forum Private Network-Network Interface (P-NNI) Phase 0 protocol, which uses static maps to switch around failed links. Figure 5-12 shows the static maps on the switch to which Router A is connected.

Figure 5-12 Example of an Multi-Switch Network That Uses the P-NNI Phase 0 Protocol



When a physical link fails, the ATM switch tears down the virtual circuits for that link. When the AIP in Router A detects that a virtual circuit has been torn down, it resignals the network to reestablish the VCC. When the switch receives the new signaling packet and realizes that the primary interface is down, it forwards the request on the alternate interface.

ATM Media

The ATM Forum has defined multiple standards for encoding ATM over various types of media. Table 5-2 lists the framing type and data rates for the various media, including unshielded twisted-pair (UTP) and shielded twisted-pair (STP) cable.

Table 5-2 ATM Physical Rates

		Media					
Framing	**Data Rate (Mbps)**	**Multimode Fiber**	**Single Mode Fiber**	**Coaxial Cable**	**UTP-3**	**UTP-5**	**STP**
DS-1	1.544			Ð			
E1	2.048			Ð			
DS-3	45			Ð			
E3	34			Ð			
STS-1	51				Ð		
SONET STS3c SDH STM1	155	Ð	Ð	Ð		Ð	
SONET STS12c SDH STM4	622	Ð	Ð				
TAXI 4B/5B	100	Ð					
8B/10B (Fiber Channel)	155	Ð					Ð

Because the FDDI chipset standard, TAXI 4B/5B, was readily available, the ATM Forum encouraged initial ATM development efforts by endorsing TAXI 4B/5B as one of the first ATM media encoding standards. Today, however, the most common fiber interface is STS3c/STM.

There are two standards for running ATM over copper cable: UTP-3 and UTP-5. The UTP-5 specification supports 155 Mbps with NRZI encoding, while the UTP-3 specification supports 51 Mbps with CAP-16 encoding. CAP-16 is more difficult to implement, so, while it may be cheaper to wire with UTP-3 cable, workstation cards designed for CAP-16 based UTP-3 may be more expensive and will offer less bandwidth.

Because ATM is designed to run over fiber and copper cable, investments in these media today will maintain their value when networks migrate to full ATM implementations as ATM technology matures.

CHAPTER 6

Designing Packet Service Internetworks

This chapter focuses on the implementation of packet-switching services and addresses internetwork design in terms of the following packet-switching service topics:

- Hierarchical internetwork design
- Topology design
- Broadcast issues
- Performance issues

Information provided in this chapter is organized around these central topics. An introductory discussion outlines the general issues; subsequent discussions focus on considerations for the specific packet-switching technologies.

Note This release of the *Internetwork Design Guide* focuses on general packet-switching considerations and Frame Relay internetworks. Subsequent releases will present information tailored to other commonly implemented packet services. Frame Relay was selected as the emphasis for the first release of this document because it presents a comprehensive illustration of design considerations for interconnection to packet-switching services.

Packet-Switched Internetwork Design

The chief trade-off in linking local-area networks (LANs) and private wide-area networks (WANs) into packet-switching data network (PSDN) services is between cost and performance. An ideal design optimizes packet-services. Service optimization does not necessarily translate into picking the service mix that represents the lowest possible tariffs. Successful packet-service implementations result from adhering to two basic rules:

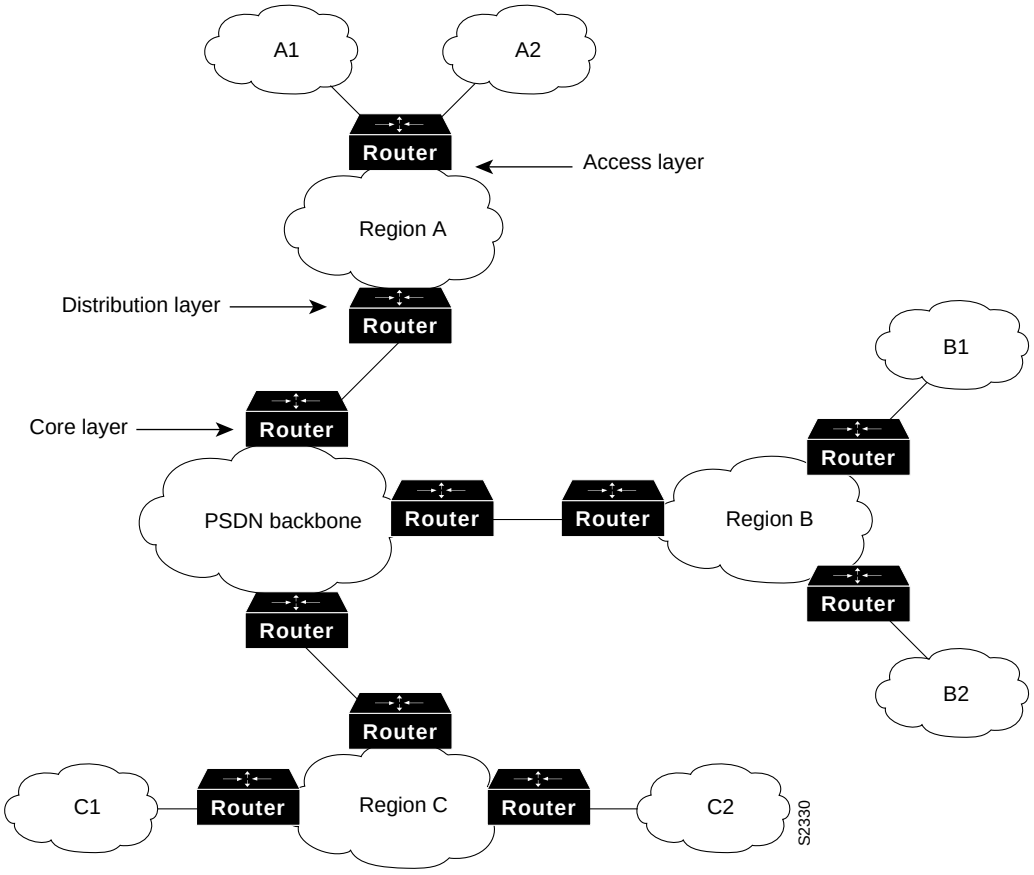
- When implementing a packet-switching solution, be sure to balance cost savings derived by instituting PSDN interconnections with your computing community's performance requirements.
- Build an environment that is manageable and that can scale up as more WAN links are required.

These rules recur as underlying themes in the discussions that follow. The introductory sections outline the overall issues that influence the ways in which packet-switched internetworks are designed.

Hierarchical Design

The objective of a hierarchical internetwork design is to modularize the elements of a large internetwork into layers of internetworking. The general model of this hierarchy is described in Chapter 1 “Internetworking Design Basics.” The key functional layers in this model are the access, distribution, and backbone (or core) routing layers. In essence, a hierarchical approach strives to split networks into subnetworks, so that traffic and nodes can be more easily managed. Hierarchical designs also facilitate scaling of internetworks because new subnetwork modules and internetworking technologies can be integrated into the overall scheme without disrupting the existing backbone. Figure 6-1 illustrates the basic approach to hierarchical design.

Figure 6-1 Hierarchical Packet-Switched Interconnection



Three basic advantages tilt the design decision in favor of a hierarchical approach:

- Scalability of hierarchical internetworks
- Manageability of hierarchical internetworks
- Optimization of broadcast and multicast control traffic

Scalability of Hierarchical Internetworks

Scalability is a primary advantage that supports using a hierarchical approach to packet-service connections. Hierarchical internetworks are more scalable because they allow you to grow your internetwork in incremental modules without running into the limitations that are quickly encountered with a flat, nonhierarchical structure.

However, hierarchical internetworks raise certain issues that require careful planning. These issues include the costs of virtual circuits, the complexity inherent in a hierarchical design (particularly when integrated with a meshed topology), and the need for additional router interfaces to separate layers in your hierarchy.

To take advantage of a hierarchical design, you must match your hierarchy of internetworks with a complementary approach in your regional topologies. Design specifics depend on the packet services you implement, as well as your requirements for fault tolerance, cost, and overall performance.

Manageability of Hierarchical Internetworks

Hierarchical designs offer several management advantages:

- Internetwork simplicity—Adopting a hierarchical design reduces the overall complexity of an internetwork by partitioning elements into smaller units. This partitioning of elements makes troubleshooting easier, while providing inherent protection against the propagation of broadcast storms, routing loops, or other potential problems.
- Design flexibility—Hierarchical internetwork designs provide greater flexibility in the use of WAN packet services. Most internetworks benefit from using a hybrid approach to the overall internetwork structure. In many cases, leased lines can be implemented in the backbone, with packet-switching services used in the distribution and access internetworks.
- Router management—With the use of a layered, hierarchical approach to router implementation, the complexity of individual router configurations is substantially reduced because each router has fewer neighbors or peers with which to communicate.

Optimization of Broadcast and Multicast Control Traffic

The effect of broadcasting in packet-service networks (discussed in “Broadcast Issues” later in this chapter) require you to implement smaller groups of routers. Typical examples of broadcast traffic are the routing updates and Novell Service Advertisement Protocol (SAP) updates that are broadcast between routers on a PSDN. An excessively high population of routers in any area or layer of the overall internetwork might result in traffic bottlenecks brought on by broadcast replication. A hierarchical scheme allows you to limit the level of broadcasting between regions and into your backbone.

Topology Design

Once you have established your overall internetwork scheme, you must settle on an approach for handling interconnections among sites within the same administrative region or area. In designing any regional WAN, whether it is based on packet-switching services or point-to-point interconnections, there are three basic design approaches that you can adopt:

- Star topologies
- Fully meshed topologies
- Partially meshed topologies

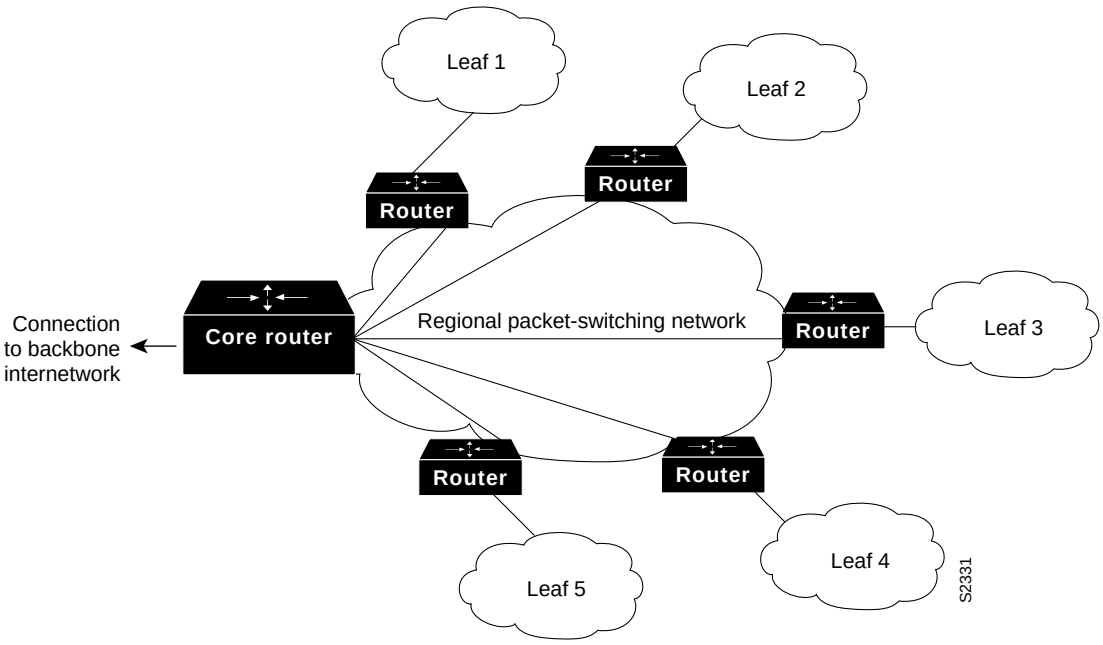
The following discussions introduce these topologies. Technology-specific discussions presented in this chapter address the applicability of these topologies for the specific packet-switching services.

Note Illustrations in this chapter use lines to show the interconnections of specific routers on the PSDN network. These interconnections are virtual connections, facilitated by mapping features within the routers. Actual physical connections are generally made to switches within the PSDN. Unless otherwise specified, the connecting lines represent these virtual connections in the PSDN.

Star Topologies

A star topology features a single internetworking hub providing access from leaf internetworks into the backbone and access to each other only through the core router. Figure 6-2 illustrates a packet-switched star topology for a regional internetwork.

Figure 6-2 Star Topology

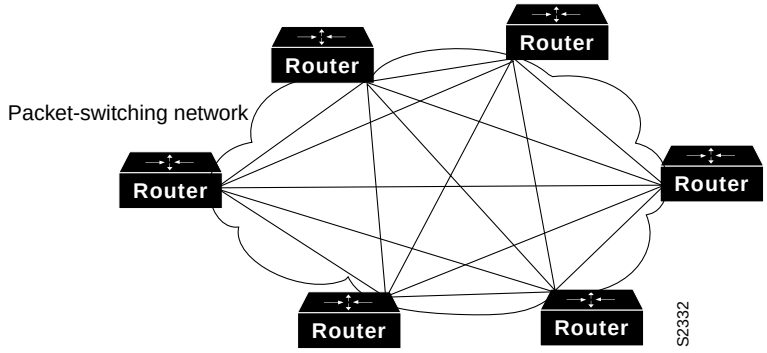


The advantages of a star approach are simplified management and minimized tariff costs. However, the disadvantages are significant. First, the core router represents a single point of failure. Second, the core router limits overall performance for access to backbone resources because it is a single pipe through which all traffic intended for the backbone (or for the other regional routers) must pass. Third, this topology is not scalable.

Fully Meshed Topologies

A fully meshed topology means that each routing node on the periphery of a given packet-switching network has a direct path to every other node on the cloud. Figure 6-3 illustrates this kind of arrangement.

Figure 6-3 Fully Meshed Topology



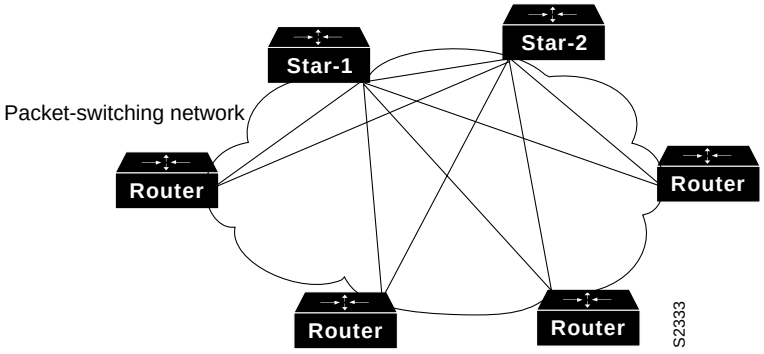
The key rationale for creating a fully meshed environment is to provide a high level of redundancy. Although a fully meshed topology facilitates support of all network protocols, it is not tenable in large packet-switched internetworks. Key issues are the large number of virtual circuits required (one for every connection between routers), problems associated with the large number of packet/broadcast replications required, and the configuration complexity for routers in the absence of multicast support in nonbroadcast environments.

By combining fully meshed and star approaches into a partially meshed environment, you can improve fault tolerance without encountering the performance and management problems associated with a fully meshed approach. The next section discusses the partially meshed approach.

Partially Meshed Topologies

A partially meshed topology reduces the number of routers within a region that have direct connections to all other nodes in the region. All nodes are not connected to all other nodes. For a nonmeshed node to communicate with another nonmeshed node, it must send traffic through one of the collection point routers. Figure 6-4 illustrates such a situation.

Figure 6-4 Partially Meshed Topology



There are many forms of partially meshed topologies. In general, partially meshed approaches are considered to provide the best balance for regional topologies in terms of the number of virtual circuits, redundancy, and performance.

Broadcast Issues

The existence of broadcast traffic can present problems when introduced into packet-service internetworks. Broadcasts are necessary for a station to reach multiple stations with a single packet when the specific address of each intended recipient is not known by the sending node. Table 6-1 lists common networking protocols and the general level of broadcast traffic associated with each, assuming a large-scale internetwork with many routing nodes.

Table 6-1 Broadcast Traffic Levels of Protocols in Large-Scale Internetworks

Network Protocol	Routing Protocol	Relative Broadcast Traffic Level
AppleTalk	Routing Table Maintenance Protocol (RTMP)	High
	Enhanced Interior Gateway Routing Protocol (Enhanced IGRP)	Low
Novell Internetwork Packet Exchange (IPX)	Routing Information Protocol (RIP)	High
	Service Advertisement Protocol (SAP)	High
	Enhanced IGRP	Low
Internet Protocol (IP)	RIP	High
	Interior Gateway Routing Protocol (IGRP)	High
	Open Shortest Path First (OSPF)	Low
	Intermediate System-to-Intermediate System (IS-IS)	Low
	Enhanced IGRP	Low
	Border Gateway Protocol (BGP)	None
	Exterior Gateway Protocol (EGP)	None
DECnet Phase IV	DECnet Routing	High
DECnet Phase V	IS-IS	Low
International Organization for Standardization (ISO) Connectionless Network Service (CLNS)	IS-IS	Low
	ISO-IGRP	High
Xerox Network Systems (XNS)	RIP	High
Banyan Virtual Integrated Network Service (VINES)	Routing Table Protocol (RTP)	High
	Sequenced RTP	Low

The relative values *high* and *low* in Table 6-1 provide a general range for these protocols. Your situation and implementation will determine the magnitude of broadcast traffic. For instance, the level of broadcast traffic generated in an AppleTalk EIGRP environment depends on the setting of the EIGRP hello-timer interval. Another issue relates to the size of the internetwork. In a small-scale internetwork, the amount of broadcast traffic generated by EIGRP nodes might be *higher* than with a comparable RTMP-based internetwork. However, for large-scale internetworks EIGRP nodes generate substantially less broadcast traffic than RTMP-based nodes.

Managing packet replication is an important design consideration when integrating broadcast-type LANs (such as Ethernet) with nonbroadcast packet services (such as X.25). With the multiple virtual circuits that are characteristic of connections to packet-switched environments, routers must replicate broadcasts for each virtual circuit on a given physical line.

With highly meshed environments, replicating broadcasts can be expensive in terms of increased required bandwidth and number of CPU cycles. Despite the advantages that meshed topologies offer, they are generally impractical for large packet-switching internetworks. Nonetheless, some level of circuit meshing is essential to ensure fault tolerance. The key is to balance the trade-off in performance with requirements for circuit redundancy.

Performance Issues

When designing a WAN around a specific packet service type, you must consider the individual characteristics of the virtual circuit. For instance, performance under certain conditions will depend on a given virtual circuit's ability to accommodate mixed protocol traffic. Depending on how the multiprotocol traffic is queued and streamed from one node to the next, certain protocols may require special handling. One solution might be to assign specific virtual circuits to specific protocol types. Performance concerns for specific packet-switching services include *committed information rates* (CIR) in Frame Relay internetworks and window size limitations in X.25 internetworks.

Frame Relay Internetwork Design

One of the chief concerns when designing a Frame Relay implementation is *scalability.* As your requirements for remote interconnections grow, your internetwork must be able to grow to accommodate changes. The internetwork must also provide an acceptable level of performance, while minimizing maintenance and management requirements. Meeting all these objectives simultaneously can be quite a balancing act.

The discussions that follow focus on several important factors:

- Hierarchical design for Frame Relay internetworks
- Regional topologies for Frame Relay internetworks
- Broadcast issues for Frame Relay internetworks
- Performance issues for Frame Relay internetworks

The guidelines and suggestions that follow are intended to provide a foundation for constructing scalable Frame Relay internetworks that balance performance, fault tolerance, and cost.

Hierarchical Design for Frame Relay Internetworks

In general, the arguments supporting hierarchical design for packet-switching networks discussed in the section "Hierarchical Design" earlier in this chapter apply to hierarchical design for Frame Relay internetworks. To review, the three factors driving the recommendation for implementing a hierarchical design are:

- Scalability of hierarchical internetworks
- Manageability of hierarchical internetworks
- Optimization of broadcast and multicast control traffic

The method by which many Frame Relay vendors tariff services is by Data Link Connection Identifier (DLCI), which identifies a Frame Relay permanent virtual connection. A Frame Relay permanent virtual connection is equivalent to an X.25 permanent virtual circuit, which, in X.25

terminology, is identified by a logical channel number (LCN). The DLCI defines the interconnection between Frame Relay elements. For any given internetwork implementation, the number of Frame Relay permanent virtual connections is highly dependent on the protocols in use and actual traffic patterns.

In general, Frame Relay designs should feature a maximum of 10 to 50 DLCIs per interface in a given internetwork. The specific number depends on several factors that should be considered together:

- Protocols being routed—Any broadcast-intensive protocol constrains the number of assignable DLCIs. For example, AppleTalk is a protocol that is characterized by high levels of broadcast overhead. Another example is Novell IPX, which sends both routing and service updates resulting in higher broadcast bandwidth overhead. In contrast, IGRP is less broadcast intensive because it sends routing updates less often (by default, every 90 seconds). However, IGRP can become broadcast intensive if its IGRP timers are modified so that updates are sent more frequently.
- Broadcast traffic—Broadcasts, such as routing updates, are the single most important consideration in determining the number of DLCIs that can be defined. The amount and type of broadcast traffic will guide your ability to assign DLCIs within this general recommended range. Refer to Table 6-1 earlier in this chapter for a list of the relative level of broadcast traffic associated with common protocols.
- Speed of lines—If broadcast traffic levels are expected to be high, you should consider faster lines and DLCIs with higher CIR and excess burst (B_e) limits. You should also implement fewer DLCIs.
- Static routes—If static routing is implemented, you can use a larger number of DLCIs per line, because a larger number of DLCIs reduces the level of broadcasting.
- Size of routing protocol and SAP updates—The larger the internetwork, the larger the size of these updates. The larger the updates, the fewer the number of DLCIs that you can assign.

Two forms of hierarchical design can be implemented with Frame Relay internetworks:

- Hierarchical meshed
- Hybrid meshed

Both designs have advantages and disadvantages. The brief discussions that follow contrast these two approaches.

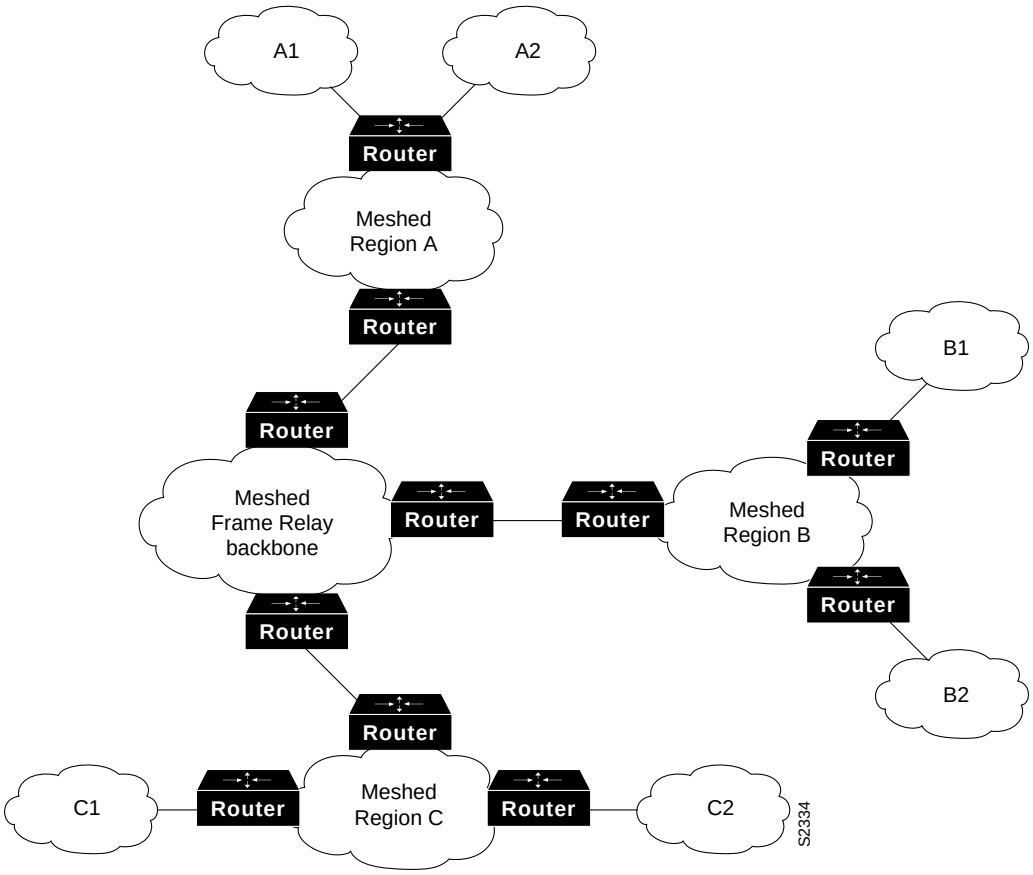
Hierarchical Meshed Frame Relay Internetworks

The objectives of implementing a hierarchical mesh for Frame Relay environments is to avoid implementing excessively large numbers of DLCIs and to provide a manageable, segmented environment. The hierarchical meshed environment features full meshing within the core PSDN and full meshing throughout the peripheral internetworks. The hierarchy is created by strategically locating routers between internetwork elements in the hierarchy. Figure 6-5 illustrates a simple hierarchical mesh. The internetwork illustrated in Figure 6-5 illustrates a fully meshed backbone, with meshed regional internetworks and broadcast networks at the outer periphery.

The key advantages of the hierarchical mesh is that it scales well and localizes traffic. By placing routers between fully meshed portions of the internetwork, you limit the number of DLCIs per physical interface, segment your internetwork, and make the internetwork more manageable. However, consider the following two issues when implementing a hierarchical mesh:

- Broadcast and packet replication—In an environment that has a large number of multiple DLCIs per router interface, excessive broadcast and packet replication can impair overall performance. With a high level of meshing throughout a hierarchical mesh, excessive broadcast and packet replication is a significant concern. In the backbone, where traffic throughput requirements are typically high, preventing bandwidth loss due to broadcast traffic and packet replication is particularly important.
- Increased costs associated with additional router interfaces—Compared with a fully meshed topology, additional routers are needed to separate the meshed backbone from the meshed peripheral internetworks. However, by using these routers, you can create much larger internetworks that scale almost indefinitely in comparison with a fully meshed internetwork.

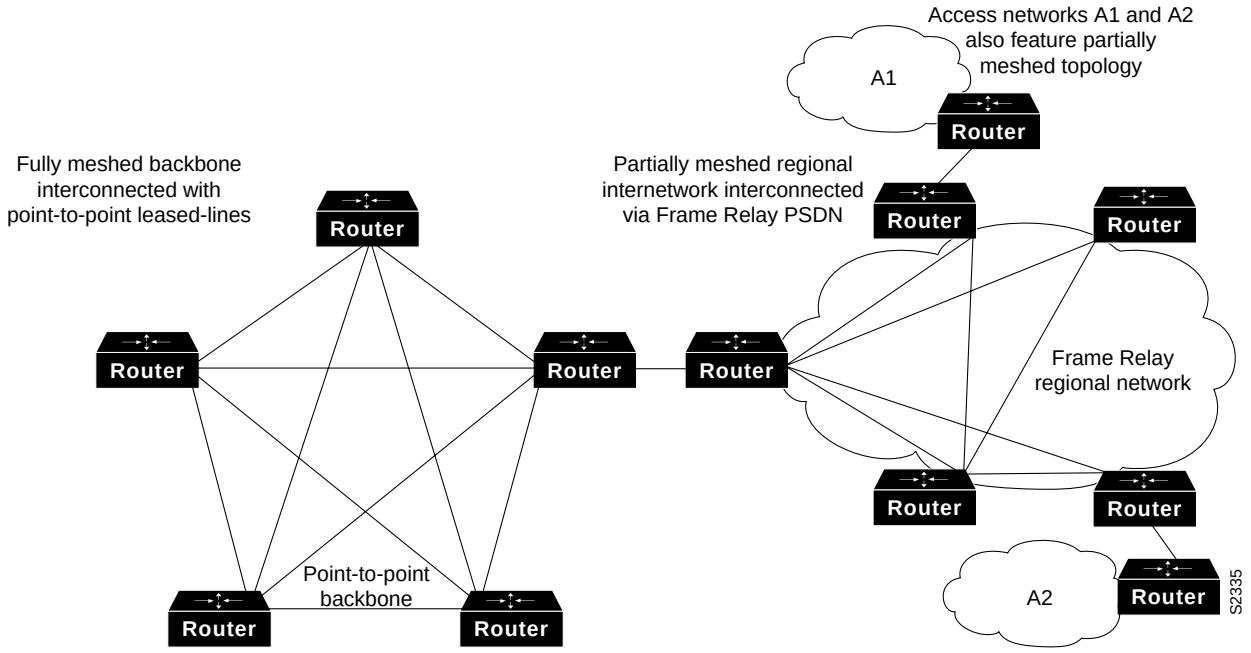
Figure 6-5 Fully Meshed Hierarchical Frame Relay Environment



Hybrid Meshed Frame Relay Internetworks

The economic and strategic importance of backbone environments often force internetwork designers to implement a hybrid meshed approach to WAN internetworks. Hybrid meshed internetworks feature redundant, meshed leased lines in the WAN backbone and partially (or fully) meshed Frame Relay PSDNs in the periphery. Routers separate the two elements. Figure 6-6 illustrates such a hybrid arrangement.

Figure 6-6 Hybrid Hierarchical Frame Relay Internetwork



Hybrid hierarchical meshes have the advantages of providing higher performance on the backbone, localizing traffic, and simplifying scaling of the internetwork. In addition, hybrid meshed internetworks for Frame Relay are attractive because they can provide better traffic control in the backbone and they allow the backbone to be made of dedicated links, resulting greater stability.

The disadvantages of hybrid hierarchical meshes include high costs associated with the leased lines as well as broadcast and packet replication that can be significant in access internetworks.

Regional Topologies for Frame Relay Internetworks

You can adopt one of three basic design approaches for a Frame Relay-based packet service regional internetwork:

- Star
- Fully meshed
- Partially meshed

Each of these is discussed in the following sections. In general, emphasis is placed on partially meshed topologies integrated into a hierarchical environment. Star and fully meshed topologies are discussed for structural context.

Star Topologies

The general form of the star topology is addressed in the introductory discussion "Topology Design" earlier in this chapter. Stars are attractive because they minimize the number of DLCIs required and result in a low-cost solution. However, a star topology presents some inherent bandwidth limitations. Consider an environment where a backbone router is attached to a Frame Relay cloud at 256 kbps, while the remote sites are attached at 56 kbps. Such a topology will throttle traffic coming off the backbone intended for the remote sites.

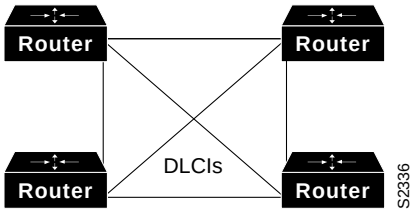
As suggested in the general discussion, a strict star topology does not offer the fault tolerance needed for many internetworking situations. If the link from the hub router to a specific leaf router is lost, all connectivity to the leaf router is lost.

Fully Meshed Topologies

A fully meshed topology mandates that every routing node connected to a Frame Relay internetwork is logically linked via an assigned DLCI to every other node on the cloud. This topology is not tenable for larger Frame Relay internetworks for several reasons:

- Large, fully meshed Frame Relay internetworks require many DLCIs. One is required for each logical link between nodes. As shown in Figure 6-7, a fully connected topology requires the assignment of $[n(n-1)]/2$ DLCIs, where n is the number of routers to be directly connected.

Figure 6-7 Fully Meshed Frame Relay



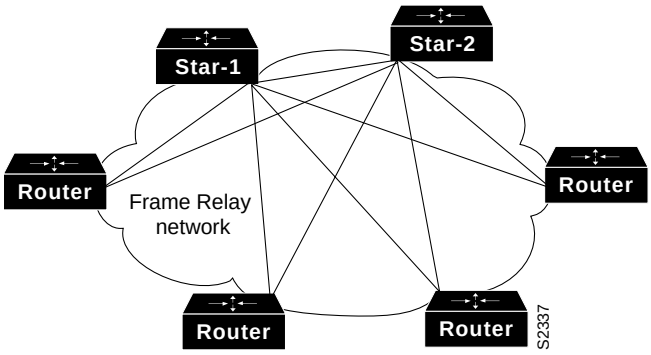
- Broadcast replication will choke internetworks in large, meshed Frame Relay topologies. Routers inherently treat Frame Relay as a broadcast media. Each time a router sends a multicast frame (such as a routing update, spanning tree update, or SAP update), the router must copy the frame to each DLCI for that Frame Relay interface.

These problems combine to make fully meshed topologies unworkable and unscaleable for all but relatively small Frame Relay implementations.

Partially Meshed Topologies

Combining the concepts of the star topology and the fully meshed topology results in the partially meshed topology. Partially meshed topologies are generally recommended for Frame Relay regional environments because they offer superior fault tolerance (through redundant stars) and are less expensive than a fully meshed environment. In general, you should implement the minimum meshing to eliminate single point-of-failure risk. Figure 6-8 illustrates a twin-star, partially meshed approach. This arrangement is supported in Frame Relay internetworks running IP, ISO CLNS, DECnet, Novell IPX, AppleTalk and bridging.

Figure 6-8 Twin-Star Partially Meshed Frame Relay Internetwork

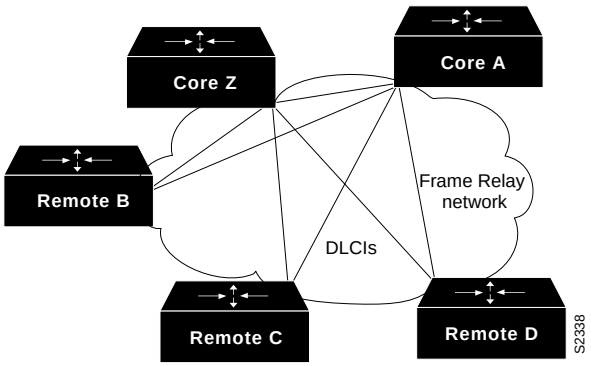


A feature called *virtual interface*s (introduced with Software Release 9.21) allows you to create internetworks using partially meshed Frame Relay designs as shown in Figure 6-8.

To create this type of internetwork, individual physical interfaces are split into multiple virtual (logical) interfaces. The implication for Frame Relay is that DLCIs can be grouped or separated to maximize utility. For example, small fully meshed clouds of Frame Relay-connected routers can travel over a group of four DLCIs clustered on a single virtual interface, while a fifth DLCI on a separate virtual interface provides connectivity to a completely separate internetwork. All of this connectivity occurs over a single physical interface connected to the Frame Relay service.

Prior to Software Release 9.21, virtual interfaces were not available and partially meshed topologies posed potential problems, depending on the internetwork protocols used. Consider the topology illustrated in Figure 6-9.

Figure 6-9 Partially Meshed Frame Relay Internetwork



Given a standard router configuration and router software predating Software Release 9.21, the connectivity available in the internetwork shown in Figure 6-9 can be characterized as follows:

- Core A and Core Z can reach all the remote routers.
- Remote B, Remote C, and Remote D cannot reach each other.

For Frame Relay implementations running software prior to Software Release 9.21, the only way to permit connectivity among all these routers is by using a distance vector routing protocol that can disable split horizon, such as RIP or IGRP for IP. Any other internetwork protocol, such as AppleTalk or ISO CLNS, does not work. The following configuration listing illustrates an IGRP configuration to support a partially meshed arrangement.

```
router igrp 20
network 45.0.0.0
!
interface serial 3
encapsulation frame-relay
ip address 45.1.2.3 255.255.255.0
no ip split-horizon
```

This topology only works with distance vector routing protocols assuming you want to establish connectivity from Remote B, C, or D to Core A or Core Z, but not across paths. This topology does not work with link state routing protocols because the router cannot verify complete adjacencies. Note that you will see routes and services of the leaf nodes that cannot be reached.

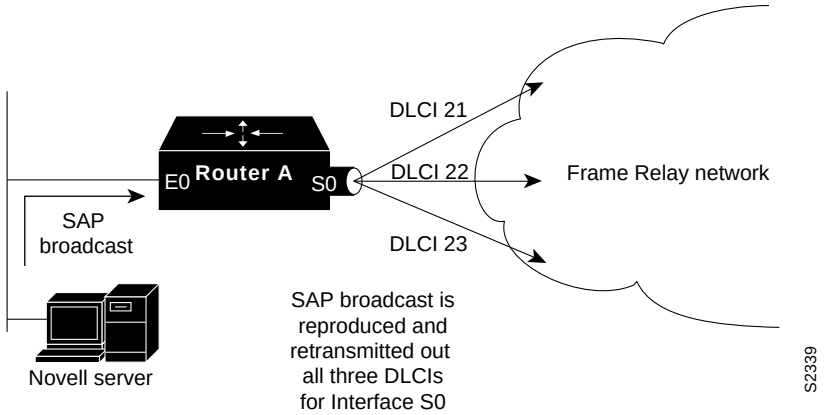
Broadcast Issues for Frame Relay Internetworks

Routers treat Frame Relay as a broadcast media, which means that each time the router sends a multicast frame (such as a routing update, spanning tree update, or SAP update), the router must replicate the frame to each DLCI for the Frame Relay interface. Frame replication results in substantial overhead for the router and for the physical interface.

Consider a Novell IPX environment with multiple DLCIs configured for a single physical serial interface. Every time a SAP update is detected, the router must replicate it and sent it down the virtual interface associated with each DLCI. Figure 6-10 illustrates this situation.

Note One way to reduce broadcasts is to implement more efficient routing protocols, such as EIGRP, and to adjust timers on lower speed Frame Relay services.

Figure 6-10 SAP Replication in Frame Relay Virtual Interface Environment



Performance Issues for Frame Relay Internetworks

Two important performance concerns must be addressed when you are implementing a Frame Relay internetwork:

- Packet-switched service provider tariff metrics
- Multiprotocol traffic management requirements

Each of these must be considered during the internetwork planning process. The following sections briefly discuss the impact that tariff metrics and multiprotocol traffic management can have on overall Frame Relay performance.

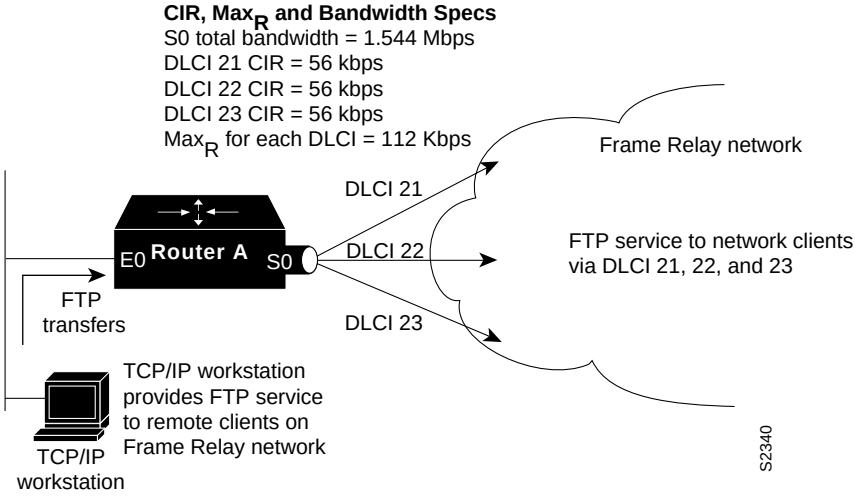
Packet-Switched Service Provider Tariff Metrics

When you contract with Frame Relay packet-switched service providers for specific capabilities, CIR, measured in bits per second, is one of the key negotiated tariff metrics. CIR is the maximum permitted traffic level that the carrier will allow on a specific DLCI into the packet-switching environment. CIR can be anything up to the capacity of the physical limitation of the connecting line.

Other key metrics are committed burst (B_c) and excess burst (B_e). B_c is the number of bits that the Frame Relay internetwork is committed to accept and transmit at the CIR. B_e sets the absolute limit for a DLCI in bits. This is the number of bits that the Frame Relay internetwork will attempt to transmit after B_c is accommodated. B_e determines a peak or maximum Frame Relay data rate (Max_R), where $Max_R = (B_c + B_e)/B_c * CIR$, measured in bits per second.

Consider the situation illustrated in Figure 6-11. In this environment, DLCIs 21, 22, and 23 are assigned CIRs of 56 kbps. Assume the Max_R for each line is 112 kbps (double the CIR). The serial line to which Router A is connected is a T1 line capable of 1.544 Mbps total throughput. Given that the type of traffic being sent into the Frame Relay internetwork consists of FTP file transfers, the potential is high that the router will attempt to transmit at a rate in excess of Max_R. If this occurs, traffic might be dropped without notification if the B_e buffers (allocated at the Frame Relay switch) overflow.

Figure 6-11 Example CIR and CBR Traffic Limiting Situation



Unfortunately, there are relatively few ways to automatically prevent traffic on a line from exceeding the Max_R. Although Frame Relay itself uses the Forward Explicit Congestion Notification (FECN) and Backward Explicit Congestion Notification (BECN) protocols to control traffic in the Frame Relay internetwork, there is no formally standardized mapping between the Frame Relay (link) level and most upper layer protocols. At this time, an FECN bit detected by a router is mapped to the congestion notification byte for DECnet Phase IV or ISO CLNS. No other protocols are supported.

The actual effect of exceeding specified CIR and derived Max_R settings depends on the types of application running on the internetwork. For instance, TCP/IP's backoff algorithm will see dropped packets as a congestion indication and sending hosts might reduce output. However, NFS has no backoff algorithm, and dropped packets will result in lost connections.

When determining the CIR, B_c, and B_e for Frame Relay connection, you should consider the actual line speed and applications to be supported.

Most Frame Relay carriers provide an appropriate level of buffering to handle instances when traffic exceeds the CIR for a given DLCI. These buffers allow excess packets to be spooled at the CIR and reduce packet loss, given a robust transport protocols such as TCP. Nonetheless overflows can happen. Remember that although routers can prioritize traffic, Frame Relay switches cannot. You cannot specify the particular traffic that is to be dropped when overflow conditions occur.

Note To avoid packet loss, implement unacknowledged application protocols (such as packet video) carefully. With these protocols, there is a greater potential for buffer overflow.

Multiprotocol Traffic Management in Frame Relay Internetworks

With multiple protocols being transmitted into a Frame Relay internetwork through a single physical interface, you might find it useful to separate traffic among different DLCIs based on protocol type. To split traffic in this way, you must assign specific protocols to specific DLCIs. This can be done by specifying static mapping on a per virtual interface basis or by defining only specific types of encapsulations for specific virtual interfaces.

Figure 6-12 illustrates the use of virtual interfaces (assigned using subinterface configuration commands) to allocate traffic to specific DLCIs. In this case, traffic of each configured protocol is sent down a specific DLCI and segregated on a per circuit basis. In addition, each protocol can be assigned a separate CIR and a separate level of buffering by the Frame Relay service provider.

Figure 6-12 Virtual Interfaces Assigned Specific Protocols

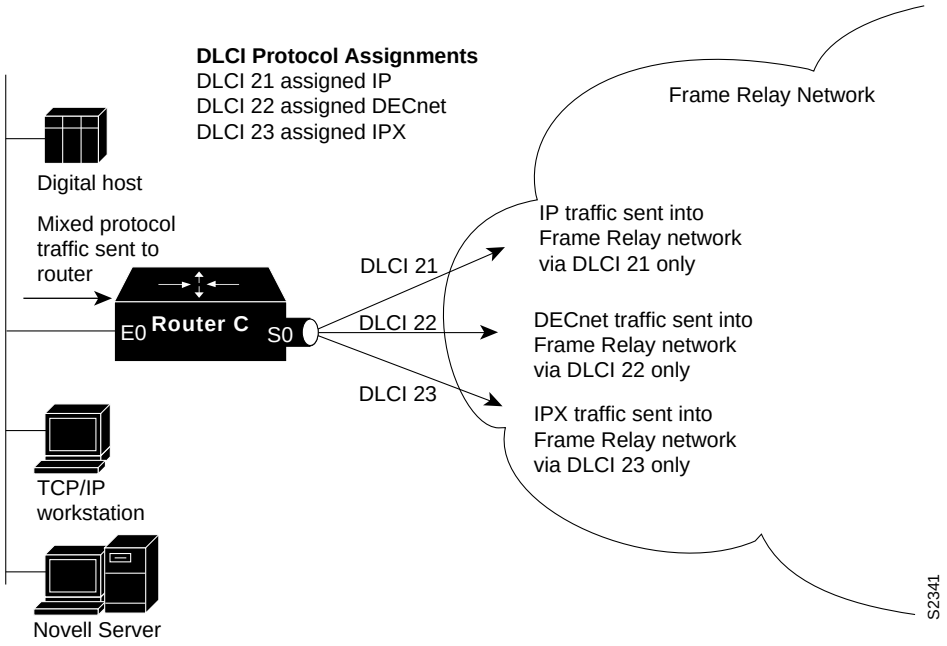


Figure 6-13 provides a listing of the subinterface configuration commands needed to support the configuration illustrated in Figure 6-12. The command listing in Figure 6-13 illustrates the enabling of the relevant protocols and the assignment of the protocols to the specific subinterfaces and associated Frame Relay DLCIs. Software Release 9.1 and later uses Frame Relay Inverse Address Resolution Protocol (IARP) to map protocol addresses to Frame Relay DLCIs dynamically. For that reason, Figure 6-13 does not show Frame Relay mappings.

Figure 6-13 Virtual Interface Configuration Example

```
interface Ethernet0
ip address 192.198.78.9 255.255.255.0
ipx network AC
decnet cost 4
no mop enabled
!
interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.1 point-to-point
ip address 131.108.3.12 255.255.255.0
frame-relay interface-dlci 21 broadcast
no frame-relay inverse-arp IP 21
no frame-relay inverse-arp NOVELL 21
no frame-relay inverse-arp APPLETALK 21
no frame-relay inverse-arp XNS 21
!
interface Serial0.2 point-to-point
no ip address
decnet cost 10
frame-relay interface-dlci 22 broadcast
no frame-relay inverse-arp IP 22
no frame-relay inverse-arp NOVELL 22
no frame-relay inverse-arp APPLETALK 22
no frame-relay inverse-arp XNS 22
!
interface Serial0.3 point-to-point
no ip address
ipx network A3
frame-relay interface-dlci 23 broadcast
no frame-relay inverse-arp IP 23
no frame-relay inverse-arp NOVELL 23
no frame-relay inverse-arp APPLETALK 23
no frame-relay inverse-arp XNS 23
!
router igrp 109
network 192.198.78.0
!
ip name-server 255.255.255.255
!
snmp-server community
!
line con 0
line aux 0
line vty 0 4
end
```

Subinterface command configuration defining Frame Relay DLCIs and assigning protocols to specific DLCIs.

S2735

You can use the following commands in Software Release 9.1 to achieve a configuration that is similar to the configuration shown in Figure 6-13:

```
Version 9.1
interface serial 0
ip address 131.108.3.12 255.255.255.0
decnet cost 10
novell network A3
frame-relay map IP 131.108.3.62 21 broadcast
frame-relay map DECNET 10.3 22 broadcast
frame-relay map NOVELL C09845 23 broadcast
```

CHAPTER 7

Designing DDR Internetworks

Dial-on-demand routing (DDR) provides network connections across Public Switched Telephone Networks (PSTNs). Traditionally, PSTN connections have been dedicated leased lines. DDR provides low-volume, periodic network connections, allowing on-demand services and decreasing network costs. In traditional routing, a router examines packets for a destination address, looks up the address in its routing table, selects an interface through which the packets can be transmitted, and then sends the packets to the destination.

With Software Release 10.0, DDR is supported for IP, Novell IPX, and AppleTalk internetworks. DDR also supports single destination transparent bridging. DDR can be used over synchronous serial interfaces, Integrated Services Digital Network (ISDN) interfaces, or asynchronous serial interfaces. V.25bis and DTR dialing are used for Switched 56 CSU/DSUs, ISDN terminal adapters (TAs), or synchronous modems. Asynchronous serial lines are available on the auxiliary port on Cisco routers and on Cisco communication servers for connections to asynchronous modems. DDR is supported over ISDN using the Basic Rate Interface (BRI). Cisco routers that run Internetworking Operating System (IOS) 10.2 and have T1 channelized interfaces support the ISDN Primary Rate Interface (PRI).

To establish a DDR connection, a router goes through the following steps:

1. Determines that there is a route to the destination.
2. Locates the DDR interface to that destination.
3. Checks the DDR interface to see if it is connected to the destination.
4. Determines if the packet is *interesting* (permitted by access list) or *uninteresting* (denied by access list). If the packet is uninteresting and there is no connection established, the packet is dropped. If the packet is uninteresting, but a connection is already established to the specified destination, the packet is sent across the connection. If the packet is interesting, it is sent and the idle timer is reset. If the packet is interesting, and there is no connection, the router attempts to establish a connection.

Note This design guide assumes that the reader has a familiarity with DDR and terms associated with it. For an in-depth case study that contains several scenarios with detailed configuration examples illustrating DDR over IP internetworks, see Topic 2, “Dial-on-Demand Routing” in the Cisco publication *Internetwork Applications: Case Studies, Vol. 1 No. 1*. This case study explains the flow of traffic through the router in a DDR environment in detail.

When designing DDR internetworks, ask the following questions:

- What is the topology to be used?

  There are three basic topologies used with DDR networks: point-to-point, hub and spoke, and fully meshed. Issues which need to be decided as part of the topology are what type of addressing scheme is desired and what are the security issues which need to be addressed.

- What media is to be used?

  Media choices include asynchronous serial, synchronous serial, and ISDN. This choice will affect how packets are sent.

- Where are the packets going?

  To define where packets are sent, configure static routes, zones, and services. Static routes or zones are critical, because dynamic routing information is sent only after a DDR connection has been established. Static services ensure that only specified service advertisements will establish a DDR connection.

- How are the packets sent?

  To determine how packets reach their destination, configure dialer interfaces and map addresses to telephone numbers.

- When should the router connect?

  Interesting packets will establish DDR connections. To avoid unwanted DDR connections, configure packets to be uninteresting by denying packets through access lists. Packets types you may want to configure as uninteresting are regular routing updates, service advertisements, and serialization packets. You can also eliminate AppleTalk broadcasts and spoof IPX watchdog packets to further avoid unwanted connections.

The guidelines and suggestions that follow are intended to provide a foundation for constructing scalable dial-on-demand internetworks that balance performance, fault tolerance, and cost.

DDR Topology Design

You can adopt one of three basic topologies for a DDR internetwork:

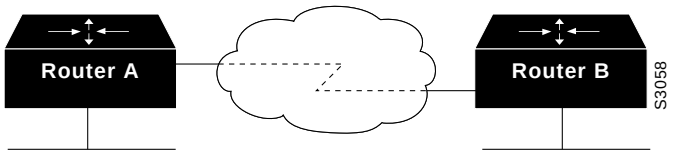
- Point-to-point
- Hub and spoke
- Fully meshed

In each topology, consider whether the local or remote sites are set to answer calls, to place calls, or to both place and answer calls.

Point to Point

In a simple point-to-point topology, two sites are connected to each other. Each has a dialer interface and maps the other site's address to a telephone number. If load sharing is desired, more than one interface can be configured for bandwidth on demand capability. See Figure 7-1.

Figure 7-1 Point-to-Point Topology

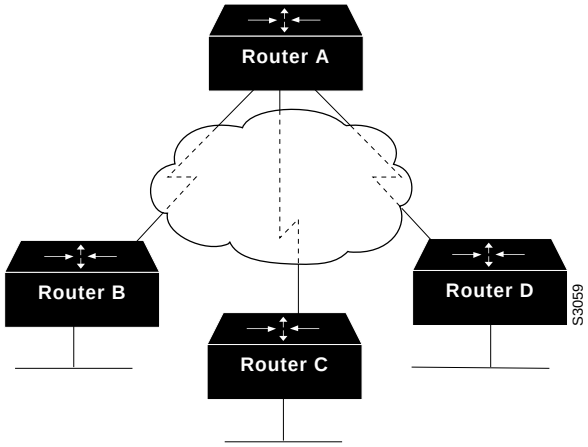


Hub and Spoke

In a hub and spoke topology, a central site is connected to several remote sites. The remote sites only communicate with the central site directly. They do not call any of the other remote sites. See Figure 7-2. The central site has several interfaces that map to the remote sites. These interfaces are placed into a rotary group. A rotary group allows several sites to share several interfaces without dedicating an interface to each site. When a rotary is used for placing calls, a free interface is selected out of all of the physical interfaces in the rotary group. When used for incoming calls, the incoming call can be received by any of the physical interfaces, and packets will still be routed correctly. If an interface is already connected, incoming or outgoing calls can be received or placed by the next available interface in the rotary group. Hub and spoke topologies are easier to configure than fully meshed topologies (described in the next section) because remote site dialer interfaces are mapped only to the central site. A hub and spoke topology works well for communication servers (8 or 16 ports) or routers with multiple BRIs, multiple serial lines, or PRI interfaces.

If you want the spokes to communicate with each other in a hub and spoke topology for IP using RIP or IGRP, IP or IPX using Enhanced IGRP, or AppleTalk using Enhanced IGRP internetworks, disable the split horizon feature which is enabled by default. With split horizon enabled, packets that are received by a particular interface are not sent out the same interface because it is assumed that all devices on that interface heard the packet that was received. In hub and spoke topologies, spokes learn about each other through the hub site to which they are connected by a single interface. In order for spokes to communicate with each other, split horizon must be disabled so that information can be sent and received over the same interface. If load sharing is desired, interfaces can be configured for bandwidth on demand capability.

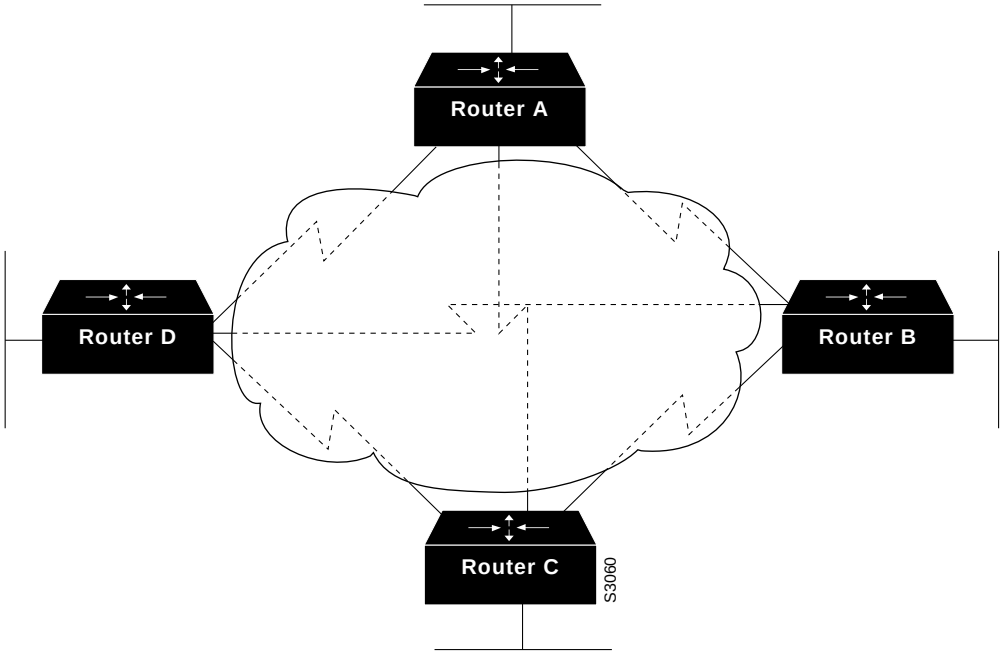
Figure 7-2 Hub and Spoke Topology



Fully Meshed

Fully meshed topologies (see Figure 7-3) streamline the dialing process because each site can call any other site directly instead of having to call through a central site (as in the hub and spoke topology) which then places another call to the target site. However, the configuration for each site is more complex because each site must have mapping information for every other site. If load sharing is desired, interfaces can be configured for bandwidth on demand capability. In addition to the configuration to being more complex, either sufficient interfaces must be available on each device to deal with the possibility of all of the other devices calling in, or the possibility of contention for interfaces needs to be dealt with and understood. The fully meshed configuration is only recommended for very small dial on demand networks.

Figure 7-3 Fully Meshed Topology



Addressing Considerations

There are normally two ways of viewing serial addressing requirements. The first is that each serial link is its own subnet. That subnet is a point-to-point connection. This is the common method used for leased lines using High-level Data Link Control (HDLC) and Point-to-Point Protocol (PPP) encapsulation. This approach tends to be used on point-to-point interfaces, or where interfaces are dedicated to specific destinations.The second addressing scheme is commonly used by Switched Multimegabit Data Service (SMDS)—each router is a different host number on the same subnet. This second approach is the method most widely used with dialer rotary groups and the hub and spoke topology. With the use of static routes pointing to the networks beyond the remote routers, the configuration is quite simple. This technique can be used for IP, IPX and Appletalk.

Security

As part of choosing a topology, thought needs to be given to where authentication is required. Authentication is used for two different reasons. The first is for security, the second to identify who is calling in so that the called router can correctly forward packets to the correct interface. This is mostly required when using dialer rotary groups where multiple sites will be calling into a single router.

For security and authentication, Point-to-Point Protocol (PPP) with Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) can be used; or for ISDN interfaces, calling line identification (if available) can be used. CHAP and PAP, used with PPP encapsulation, provide a way for routers to authenticate incoming calls.

With CHAP, a remote device attempting to connect to the local router is requested, or challenged, to respond. When the local router receives the challenge response, it verifies the response by looking up the name of the remote device given in the response. The passwords must be identical on the remote device and the local router. The names and passwords are configured using the **username** command. In the following example, Router Macbeth will allow Router Macduff to call in using the password “bubble”:

```
hostname Macbeth
username Macduff password bubble
!
encapsulation ppp
ppp authentication chap
```

In the following example, Router Macduff will allow Router Macbeth to call in using the password “bubble”:

```
hostname Macduff
username Macbeth password bubble
!
encapsulation ppp
ppp authentication chap
```

PAP, while similar to CHAP in that it is an authentication protocol used with PPP is less secure than CHAP. While CHAP never passes the password on the physical link, PAP does pass the password and hostname or username in the clear.

On asynchronous lines when using interactive mode rather than dedicated mode, the **username** command allows a router to verify a username in an internal database before allowing the user to call into the router. In the following example, user Joe Smith will be allowed to call into the router if he uses the password “freedom”:

```
username JoeSmith password freedom
line 1
login
```

You can configure BRI interfaces on Cisco 2500 or Cisco 3000 series routers to use caller ID (identification). Incoming calls are screened in order to verify that the calling line ID is from an expected origin. Caller ID screening requires a local switch that is capable of delivering the caller ID to the router.

DDR Media Considerations

The following are DDR internetwork media considerations:

- Encapsulation methods
- Synchronous serial lines
- ISDN connections
- Asynchronous modem connections

Encapsulation Methods

Cisco supports Point-to-Point Protocol (PPP), High-Level Data Link Control (HDLC), Serial Line Interface Protocol (SLIP), and X.25 data-link encapsulations for DDR.

PPP is the recommended encapsulation method because it supports multiple protocols and is used for synchronous, asynchronous, or ISDN connections. In addition, PPP performs address negotiation and authentication and is interoperable with different vendors.

HDLC is supported on synchronous serial lines and ISDN connections only. HDLC supports multiple protocols. However, HDLC does not provide authentication which may be required if using dialer rotary groups.

SLIP works on asynchronous interfaces only and is supported by IP only. Addresses must be configured manually, and SLIP does not provide authentication and is interoperable only with other vendors that use SLIP.

X.25 is supported on synchronous serial lines (IOS 10.0[5]), and a single ISDN B channel (IOS 10.2).

Synchronous Serial Lines

Dialing on synchronous serial lines can be initiated using V.25bis dialing or DTR dialing. V.25bis is an International Telecommunication Union Telecommunications Standardization Sector (ITU-T) standard for inband dialing. With inband dialing, dialing information is sent over the same connection that carries data. V.25bis is used with a variety of devices including synchronous modems, ISDN terminal adapters (TAs), and Switched 56 DSU/CSUs.

With DTR dialing, the DTR signal on the physical interface is activated, which will cause some devices to dial a number configured into that device. When using DTR dialing, the interface will not be able to receive calls. But using DTR dialing allows lower cost devices to be used in cases where only a single number needs to be dialed.

Note The ITU-T carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).

ISDN Connections

All ISDN devices subscribe to services provided by an ISDN service provider, usually a telephone company. Some service providers use Service Profile Identifiers (SPIDs) to define the services used by the ISDN device. The service provider assigns the ISDN device one or more SPIDs when you first subscribe to the service. If you are using a service provider that requires SPIDs, your ISDN device cannot place or receive calls until it sends a valid, assigned SPID to the service provider when accessing the switch to initialize the connection. Currently, only the DMS-100, NI-1, and 5ESS switch types require SPIDs. The 5ESS only requires SPIDs with multidrops. Other switches use subaddresses. In addition, SPIDs only have significance at the local access ISDN interface. SPIDs are never sent to the device being called.

ISDN calls are placed at 56 or 64 kbps. When dialing internationally, or making a DDR connection in the United States across more than one switch, ISDN lines may be available only at 56 kbps. ISDN supports caller ID (identification), providing security through authentication. For details on caller ID, see the section “Security” earlier in this chapter.

Basic Rate Interface

The ISDN BRI provides 2 B channels for sending data at 64 kbps, and 1 D channel for signalling or communicating with the ISDN switch at 16 kbps. The 2 B channels are automatically placed in a rotary group. On ISDN dialers configured for X.25 encapsulation (IOS 10.2), however, only one B channel can be used. Cisco has received certification for ISDN BRI compliance in the countries listed in Table 7-1.

Table 7-1 Cisco Certification for ISDN BRI Compliance

Country
Australia
Austria
Belgium
Canada
Denmark
Finland
Germany
Ireland
Japan
New Zealand
Norway
Portugal
Spain
Sweden
Switzerland
The Netherlands
UK
US

Primary Rate Interface

The ISDN PRI for the United States, Canada, and Japan provides 23 B channels and 1 D-channel (all at 64 kbps) for a cumulative speed equivalent to T1 (1.5 Mbps). The B channels are automatically placed in a rotary group. ISDN for Europe provides 30 B channels and 1 D channel (all at 64 kbps) for a cumulative speed equivalent to E1 (2 Mbps). ISDN PRI is supported with IOS 10.2 for 5ESS, 4ESS, and DMS-100 in North America.

Asynchronous Modem Connections

Asynchronous connections are used by communication servers or through the auxiliary port on a router. Asynchronous connections can be used by routers running TCP/IP, Novell IPX, and AppleTalk (IOS 10.2). Asynchronous connections support modems from 110 bps to 115 kbps—a real-world throughput is typically 2400 bps to 28.8 kbps.

When designing DDR internetworks over asynchronous connections, determine the type of connection you want users to make: interactive or dedicated. In interactive mode, the DDR line can be used for any type of connection, including Telnet, SLIP, or PPP encapsulation. In dedicated mode, the line is automatically placed into interface mode so that the user cannot change the encapsulation method, address, or other parameters.

If you rely on a dynamic routing protocol to receive routing information, you must enable asynchronous dynamic routing on the DDR interface so that the router will trust routes learned from that interface.

In order to dial out using asynchronous connections, chat scripts must be configured so that modem dialing and login commands are sent to remote systems. There are typically two scripts—modem (dialing) and system (login). Modem commands vary widely, depending upon modem and type and communication software. Several line commands are required to specify modem line characteristics, such as speed and parity setting. Login scripts request the network protocol and might include user name and password information for authentication purposes.

Creating Static Routes, Zones, and Service Updates

Typically, routers make decisions based on routing tables which they build from dynamic routing information. However, because routing updates are not sent over inactive DDR links, the administrator must configure static routes, services, and zones, so that routing can continue and so that hosts can still find services when the DDR link is not connected.

IP Static Routes

Use the **ip route command** to create static routes to specified destinations. To advertise static routes to other routers on the network, use the **redistribute static** command.

For example, to redistribute the static route to other networks in IGRP autonomous system 109, use the following commands:

```
router igrp 109
network 131.108.0.0
redistribute static
```

IP Default Routes

Some routers may not be able to determine the routes to all other networks. It is common to configure these routers (using the **ip default-network** command) with default routes so that if the router cannot determine the destination for particular packets, it can forward the packets to the default address. The router at the default address will forward the packets to the intended destination.

Passive Interfaces

Interfaces that are tagged as *passive* will not send routing updates. To prevent routing updates from establishing DDR connections on dialer interfaces that do not rely on dynamic routing information, configure DDR interfaces with the **passive-interface** command or use access lists as previously described. Either the **passive-interface** command or an access list is sufficient to prevent routing updates from triggering a call. However, if you want routing updates to be passed when the link is active, use an access list instead of the **passive-interface** command.

Split Horizon

Routers connected to broadcast-type IP networks and routers that use distance-vector routing protocols use split horizon to reduce the possibility of routing loops. Split horizon blocks information about routes from being advertised by a router out any interface from which that information originated.

Note If remote sites need to communicate with each other, split horizons should be disabled for hub and spoke topologies. In hub and spoke topologies, spokes learn about each other through the hub site to which they are connected by a single interface. In order for spokes to send and receive information to each other, split horizons must be disabled so that information can be sent and received over the same interface.

IPX Static Routes and SAP Updates

With DDR, it is necessary to configure static routes because routing updates are not received across inactive DDR connections. To create static routes to specified destinations, use the **ipx route** command. You can also configure static Service Advertisement Protocol (SAP) updates with the **ipx sap** command so that clients can always find a particular server. In this way, you can determine the areas on your internetwork where SAP updates will establish DDR connections.

In the following example, traffic to network 50 will always be sent to address 45.0000.0c07.00d3. Traffic to network 75 will always be sent to address 45.0000.0c07.00de. The router will respond to GNS queries with the server WALT if there are no dynamic SAPs available:

```
ipx route 50 45.0000.0c07.00d3
ipx route 75 45.0000.0c07.00de
ipx sap 4 WALT 451 75.0000.0000.0001 15
```

Configuring AppleTalk Static Zones

AppleTalk zones are by default dynamically updated with new AppleTalk addresses. To avoid unwanted DDR connections caused by dynamic zone updates, you can control the size and content of a zone statically instead. In the following example, the Marketing zone is configured to contain only addresses within a cable range of 110 to 110:

```
appletalk static cable-range 110-110 to 45.2 zone Marketing
```

Note For versions of IOS Release 10 that support snapshot routing, DDR dependence on static routes is minimized. Snapshot routing is a time-triggered technique optimized for remote sites with occasional access requirements, allowing a remote router to take a periodic snapshot of a central site routing table during a short *active* period. This information is then stored for a user-configurable period of inactivity until the next *active* period. In the event that no routing updates are exchanged during the active period (because a DDR phone number or interface is unavailable), a user-configurable retry period is activated to ensure that a full inactive period does not pass before an attempt is made to exchange routing information again. Snapshot routing supports IP (RIP and IGRP), Novell IPX (RIP and SAP) and AppleTalk (RTMP) protocols.

Setting up Dialer Maps

In addition to configuring static routes, you need to map network addresses to telephone numbers to design the DDR internetwork. Used with rotary groups, dialer maps can be configured to support multiple physical lines or multiple destinations on one interface. Dialer map statements map next hop addresses to telephone numbers. If a match is not found between a packet's next hop address and the dialer map statement defined for an interface, the packet is dropped. The next hop address for a packet is determined based on routing information. In the following example, packets received for a host on network 144.254.50.0 are routed to a next hop address of 144.254.45.2 and mapped to telephone number 555-1212:

```
ip route 144.254.50.0 255.255.255.0 144.254.45.2
interface dialer 1
dialer map IP 144.254.45.2 name HostA 5551212
```

Checks against dialer map statements for broadcasts will fail because a broadcast packet is transmitted with a next hop address of the broadcast address. If you want broadcast packets transmitted to telephone numbers defined by dialer map statements, use the **broadcast** keyword with the **dialer map** command.

To determine whether calls are placed at 56 or 64 kbps for ISDN calls, you can use the **speed** option with the **dialer map** command when configuring interfaces. See the "ISDN Connections" section earlier in this chapter for details on ISDN media. If you are calling a system that requires a login script and is running in interactive mode, use **the system-script** keyword with the **dialer map** command on asynchronous interfaces.

To take advantage of authenticated callers, use the **name** keyword with the **dialer map** command as illustrated in the following example:

```
dialer map ip 144.254.45.2 name localcall speed 64 5551212
dialer map ip 144.254.45.4 name longdistance speed 56 14155558888
```

For hub and spoke or fully-meshed topologies that use multiple connections between single sites, configure rotary groups with the **interface dialer** and **dialer rotary-group** commands. A dialer interface is an entity that allows you to propagate an interface configuration to multiple interfaces. Physical interfaces assigned to the dialer rotary group inherit the interface dialer configuration parameters.

If one of the physical interfaces in a rotary group is busy, the next available interface can be used to place or receive a call. It is not necessary to configure rotary groups for BRI or PRI interfaces because ISDN channels are automatically placed into a rotary group, but multiple BRI or PRI interfaces may be placed in a rotary group to gain the advantages of rotary groups over a large number of B channels.

To load share so that additional bandwidth is provided as needed, use the **dialer load-threshold** command. In the following example, if the load to a particular destination on an interface in dialer rotary group 1 exceeds an interface load of 55% of the total bandwidth, the dialer will initiate another call to the destination. The load is displayed in a show interface as n/255. Load is calculated dynamically, based on the configured bandwidth of 9 kbps.

```
interface dialer 1
dialer load-threshold 55
bandwidth 9
```

Note On most of the hardware platforms supporting DDR, the packets are distributed among multiple links to the same destination based on the link with the shortest queue. If there are no packets queued on the device, the same link will always be used. With this technique, if a link is not being utilized to the point at which packets are backing up in the router, the extra link will be disconnected as a cost savings. The disadvantage of this technique is that if the dialer load-threshold is set too low, the second channel will be brought up, but it will never carry traffic and will be disconnected after the idle time. ISDN PRI currently does not use this approach, but instead employs a round-robin technique across all of the ports that are active to the same destination.

Determining Interesting and Uninteresting Packets

As described earlier, if a packet is uninteresting and there is no connection established, the packet is dropped. If the packet is uninteresting, but a connection is already established to the specified destination, the packet is sent across the connection, but the idle timer is not reset. If the packet is interesting, and there is no connection on the available interface, the router attempts to establish a connection.

Once static routes are configured, you can apply access lists with the **access-list** command to DDR interfaces in order to control DDR connections by tagging packets as uninteresting and interesting. For example, RIP and IGRP routing update packets are sent across the internetwork periodically. These packets may need to be filtered with access lists to prevent unwanted DDR connections. Novell IPX SAP requests are sent periodically and will automatically activate any DDR link in their path. In the design of DDR internetworks, it is important to understand where updates and service requests are useful and where these packet types can be safely filtered. Use the **deny** option with access lists to tag packets as uninteresting. Use the **permit** option to configure interesting packets. You may also use the **passive-interface** command described earlier in the section “Passive Interfaces.”

On IP internetworks, it is important to consider filtering routing updates on DDR interfaces for the packet types listed in Table 7-2.

Table 7-2 IP Routing Update Packet Cycles

Packet Type	Periodic Update Cycle
Enhanced IGRP	5 seconds (Hello)
IGRP	90 seconds
RIP	60 seconds
OSPF	10 seconds (Hello)
IS-IS	10 seconds (Hello)

Note The routing protocols IS-IS, BGP, and OSPF are not recommended with DDR because they require an acknowledgment for routing updates. Because DDR lines are brought up as needed, DDR will not necessarily be active and available to send responses at the times the updates are sent.

On Novell IPX internetworks, it is important to consider filtering routing updates on DDR interfaces for the protocols listed in Table 7-3.

Table 7-3 Novell IPX Update Packet Cycles

Packet Type	Periodic Update Cycle
RIP	60 seconds
SAP	60 seconds
Serialization	66 seconds

Protocol-Specific Issues

While the issues of topology and the configuration of dialer maps and filtering of interesting traffic is common to all DDR connections, there specific issues for each of the following protocols:

- IP
- Novell IPX
- AppleTalk

IP

IP hosts use a variety of methods to access other IP hosts, including Telnet and the File Transfer Protocol (FTP). To initiate a DDR link, an IP host opens, for example, a Telnet session to the IP address of the destination.

IP Access Lists

Access lists determine whether packets are interesting or uninteresting. *Interesting packets* activate DDR connections automatically. Uninteresting packets do not trigger DDR connections, although if a DDR connection is already active, uninteresting packets will travel across the existing connection. You do not need to create a separate access list for each interface on a router. You can create several key access lists and apply them to as many interfaces as needed.

For example, you could apply access list 101 from the following example to several interfaces. Access list 101 prevents periodic IGRP routing updates from establishing an unwanted DDR connections, and allows all other IP packets to automatically trigger a DDR connection:

```
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

Also refer to the section “Passive Interfaces” earlier in this chapter.

Novell IPX

Novell IPX hosts are attached to local Novell IPX servers and send a Get Nearest Server (GNS) packet to discover Novell servers on the internetwork. Novell IPX hosts find routers during NetWare shell loads.

IPX Access Lists

Access lists determine whether packets are interesting or uninteresting. *Interesting packets* activate DDR connections automatically. Uninteresting packets do not trigger DDR connections, although if a DDR connection is already active, uninteresting packets will travel across the existing connection.

Novell IPX internetworks use several types of update packets that may need to be filtered with access lists. Novell hosts broadcast serialization packets as a copy-protection precaution. Routing Information Protocol (RIP) routing table updates and SAP advertisements are broadcast every 60 seconds. Serialization packets are sent approximately every 66 seconds.

In the following example, access list 901 classifies SAP (452), RIP (453), and serialization (457) packets as uninteresting and classifies IPX packet type unknown/any (0), any or RIP (1), any or SAP (4), SPX (5), NCP (17), and NetBIOS (20) as interesting:

```
access-list 901 deny 0 FFFFFFFF 452
access-list 901 deny 4 FFFFFFFF 452
access-list 901 deny 0 FFFFFFFF 453
access-list 901 deny 1 FFFFFFFF 453
access-list 901 deny 0 FFFFFFFF 457
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 452
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 453
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 457
access-list 901 permit 0
access-list 901 permit 1
access-list 901 permit 2
access-list 901 permit 4
access-list 901 permit 5
access-list 901 permit 17
```

You can permit any other type of IPX packet as needed.

With IOS 10.2, the configuration of Novell IPX access lists is improved with the support of wildcard (-1), so the previous example would be as follows:

```
access-list 901 deny -1 FFFFFFFF 452
access-list 901 deny -1 FFFFFFFF 453
access-list 901 deny -1 FFFFFFFF 457
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 452
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 453
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 457
access-list 901 permit -1
```

IPX Watchdog Packets and Spoofing

Novell IPX watchdog packets are keepalive packets that are sent from servers to clients after a client session has been idle for approximately 5 minutes. Watchdog packets (keepalives) automatically establish connections over DDR links. To configure a router to ignore watchdog packets and thus prevent unwanted DDR connections from being established, use the **ipx watchdog-spoof** command.

AppleTalk

AppleTalk hosts use the Name Binding Protocol (NBP) to map network names to AppleTalk addresses. AppleTalk hosts access routers through the Chooser.

AppleTalk Broadcasts

While you cannot filter AppleTalk updates through access lists, you can eliminate broadcast traffic by not using the **broadcast** option with the **dialer map** command. For example, you may want to eliminate Zone Information Protocol (ZIP)—ZIP broadcasts are sent to track which networks are in which zone.

Eliminating Apple Filing Protocol Updates

AppleTalk servers use the Apple Filing Protocol (AFP) to send out *tickles* approximately every 10 seconds to hosts on the network.These tickles will establish connections when propagated across DDR interfaces. To avoid unwanted DDR connections, you must manually unmount AppleTalk servers or install software on the servers that will automatically disconnect idle users after a timeout period.

Summary

When designing DDR internetworks, consider topology type: point-to-point, hub and spoke, and fully meshed. With the topology type, consider the type of addressing scheme used and security issues. Keep in mind that media choice affects how packets are sent. Define where packets are sent by configuring static routes, zones, and services. Determine how packets reach their destination by configuring dialer interfaces and mapping addresses to telephone numbers. Finally, determine when the router should connect by configuring interesting versus uninteresting packets, eliminating unwanted AppleTalk broadcasts and spoofing IPX watchdog packets. Following these guidelines will help provide a foundation for constructing scalable dial-on-demand internetworks that balance performance, fault tolerance, and cost.

APPENDIX A

Subnetting an IP Address Space

This appendix provides a partial listing of a Class B area intended to be divided into approximately 500 Open Shortest Path First (OSPF) areas. For the purposes of this example, the network is assumed to be a Class B network with the address 150.100.0.0.

Note Although a 500-area OSPF internetwork is unrealistic, using an address space like this can help to illustrate the general methodology employed to subnet an OSPF address space.

Only the address space for two of 512 areas is shown in Table A-1. These areas are defined with the base address 150.100.2.0. Illustrating the entire address space for 150.100.0.0 would require hundreds of additional pages of addressing information. Each area would require the equivalent number of entries for each of the example areas illustrated here.

Table A-1 illustrates the assignment of 255 IP addresses that have been split between two OSPF areas. Table A-1 also illustrates the boundaries of the subnets and of the two OSPF areas shown (area 8 and area 17).

For the purposes of this discussion, consider a network that requires point-to-point serial links in each area to be assigned a subnet mask that allows two hosts per subnet. All other subnets are to be allowed 14 hosts per subnet. The use of bit-wise subnetting and variable-length subnet masks (VLSMs) permit you to customize your address space by facilitating the division of address spaces into smaller groupings than is allowed when subnetting along octet boundaries. The address layout shown in Table A-1 illustrates a structured approach to assigning addresses that uses VLSM. Table A-1 presents two subnet masks: 255.255.255.240 and of 255.255.255.252. The first mask creates subnet address spaces that are four bits wide; the second mask creates a subnet address spaces that are two bits wide.

Because of the careful assignment of addresses, each area can be summarized with a single **area** router configuration command (used to define address range).

The first set of addresses starting with 150.100.2.*0xxxxxxx* (last octet represented here in binary) can be summarized into the backbone with the following command:

```
area 8 range 150.100.2.0 255.255.255.128
```

This command assigns all addresses from 150.100.2.0 to 150.100.2.127 to area 8.

Similarly, the addresses from 150.100.2.128 to 150.100.2.255 for the second area can be summarized as follows:

```
area 17 range 150.100.2.128 255.255.255.128
```

This command assigns all addresses from 150.100.2.128 to 150.100.2.255 to area 17.

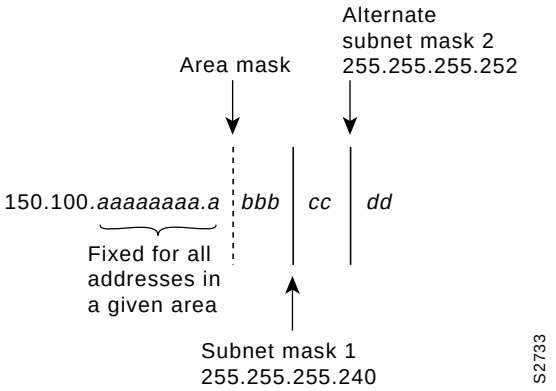
Allocation of subnets allows you to decide where to draw the line between the subnet and host (using a subnet mask) within each area. Note that in this example there are only seven bits remaining to use because of the creation of the artificial area mask. The nine bits to the left of the area mask are actually part of the subnet portion of the address. By keeping these nine bits the same for all addresses in a given area, route summarization is easily achieved at area border routers as illustrated by the scheme used in Table A-1.

Table A-1 lists individual subnets, valid IP addresses, subnet identifiers, and broadcast addresses. This method of assigning addresses for the VLSM portion of the address space guarantees that there is no address overlap. If the requirement had been different, any number of the larger subnets might be chosen and divided into smaller ranges with fewer hosts, or combined into several ranges to create subnets with more hosts.

The design approach used in this appendix allows the area mask boundary and subnet masks to be assigned to any point in the address space, which provides significant design flexibility. A change in the specification of the area mask boundary or subnet masks may be required if a network outgrows its initial address space design.

In Table A-1, the area mask boundary is to the right of most significant bit of the last octet of the address, as shown by Figure A-1.

Figure A-1 Breakdown of the Addresses Assigned by the Example



With a subnet mask of 255.255.255.240, the *a* and *b* bits together represent the subnet portion of the address, while the *c* and *d* bits together provide 4-bit host identifiers. When a subnet mask of 255.255.255.252 (a typical subnet mask for point-to-point serial lines), the *a*, *b*, and *c* bits together represent the subnet portion of the address, and the *d* bits provide 2-bit host identifiers. As mentioned earlier, the purpose of the area mask is to keep all of the *a* bits constant in a given OSPF area (independent of the subnet mask) so that route summarization is easy to apply.

The following steps outline the process used to allocate addresses:

Step 1 Determine the number of areas required for your OSPF network. A value of 500 is used for this example.

Step 2 Create an artificial *area mask boundary* in your address space. This example uses nine bits of subnet addressing space to identify the areas uniquely. Because $2^9 = 512$, nine bits of subnet meets our requirement of 500 areas.

Step 3 Determine the number of subnets required in each area and the maximum number of hosts required per subnet. This allows you to determine the placement of the subnet mask(s). In Table A-1, the requirement is for 7 subnets with 14 hosts each and 4 subnets with 2 hosts each.

Table A-1 Partial Example of Subnet Address Assignment Using VLSM

IP Address (Decimal)	Subnet Portion of Last Octet (Binary)	Host Portion of Last Octet (Binary)	Subnet Number	Subnet Mask	Notes
150.100.2.0	0000	0000	150.100.2.0	255.255.255.240	Subnet identifier; area boundary; area 8 starts
150.100.2.1	0000	0001	150.100.2.0	255.255.255.240	
150.100.2.2	0000	0010	150.100.2.0	255.255.255.240	
150.100.2.3	0000	0011	150.100.2.0	255.255.255.240	
150.100.2.4	0000	0100	150.100.2.0	255.255.255.240	
150.100.2.5	0000	0101	150.100.2.0	255.255.255.240	
150.100.2.6	0000	0110	150.100.2.0	255.255.255.240	
150.100.2.7	0000	0111	150.100.2.0	255.255.255.240	
150.100.2.8	0000	1000	150.100.2.0	255.255.255.240	
150.100.2.9	0000	1001	150.100.2.0	255.255.255.240	
150.100.2.10	0000	1010	150.100.2.0	255.255.255.240	
150.100.2.11	0000	1011	150.100.2.0	255.255.255.240	
150.100.2.12	0000	1100	150.100.2.0	255.255.255.240	
150.100.2.13	0000	1101	150.100.2.0	255.255.255.240	
150.100.2.14	0000	1110	150.100.2.0	255.255.255.240	
150.100.2.15	0000	1111	150.100.2.0	255.255.255.240	Subnet broadcast
150.100.2.16	0001	0000	150.100.2.16	255.255.255.240	Subnet identifier
150.100.2.17	0001	0001	150.100.2.16	255.255.255.240	
150.100.2.18	0001	0010	150.100.2.16	255.255.255.240	
150.100.2.19	0001	0011	150.100.2.16	255.255.255.240	
150.100.2.20	0001	0100	150.100.2.16	255.255.255.240	
150.100.2.21	0001	0101	150.100.2.16	255.255.255.240	
150.100.2.22	0001	0110	150.100.2.16	255.255.255.240	
150.100.2.23	0001	0111	150.100.2.16	255.255.255.240	
150.100.2.24	0001	1000	150.100.2.16	255.255.255.240	
150.100.2.25	0001	1001	150.100.2.16	255.255.255.240	

IP Address (Decimal)	Subnet Portion of Last Octet (Binary)	Host Portion of Last Octet (Binary)	Subnet Number	Subnet Mask	Notes
150.100.2.26	0001	1010	150.100.2.16	255.255.255.240	
150.100.2.27	0001	1011	150.100.2.16	255.255.255.240	
150.100.2.28	0001	1100	150.100.2.16	255.255.255.240	
150.100.2.29	0001	1101	150.100.2.16	255.255.255.240	
150.100.2.30	0001	1110	150.100.2.16	255.255.255.240	
150.100.2.31	0001	1111	150.100.2.16	255.255.255.240	Subnet broadcast
150.100.2.32	0010	0000	150.100.2.32	255.255.255.240	Subnet identifier
150.100.2.33	0010	0001	150.100.2.32	255.255.255.240	
150.100.2.34	0010	0010	150.100.2.32	255.255.255.240	
150.100.2.35	0010	0011	150.100.2.32	255.255.255.240	
150.100.2.36	0010	0100	150.100.2.32	255.255.255.240	
150.100.2.37	0010	0101	150.100.2.32	255.255.255.240	
150.100.2.38	0010	0110	150.100.2.32	255.255.255.240	
150.100.2.39	0010	0111	150.100.2.32	255.255.255.240	
150.100.2.40	0010	1000	150.100.2.32	255.255.255.240	
150.100.2.41	0010	1001	150.100.2.32	255.255.255.240	
150.100.2.42	0010	1010	150.100.2.32	255.255.255.240	
150.100.2.43	0010	1011	150.100.2.32	255.255.255.240	
150.100.2.44	0010	1100	150.100.2.32	255.255.255.240	
150.100.2.45	0010	1101	150.100.2.32	255.255.255.240	
150.100.2.46	0010	1110	150.100.2.32	255.255.255.240	
150.100.2.47	0010	1111	150.100.2.32	255.255.255.240	Subnet broadcast
150.100.2.48	0011	0000	150.100.2.48	255.255.255.240	Subnet identifier
150.100.2.49	0011	0001	150.100.2.48	255.255.255.240	
150.100.2.50	0011	0010	150.100.2.48	255.255.255.240	
150.100.2.51	0011	0011	150.100.2.48	255.255.255.240	
150.100.2.52	0011	0100	150.100.2.48	255.255.255.240	
150.100.2.53	0011	0101	150.100.2.48	255.255.255.240	
150.100.2.54	0011	0110	150.100.2.48	255.255.255.240	
150.100.2.55	0011	0111	150.100.2.48	255.255.255.240	
150.100.2.56	0011	1000	150.100.2.48	255.255.255.240	
150.100.2.57	0011	1001	150.100.2.48	255.255.255.240	
150.100.2.58	0011	1010	150.100.2.48	255.255.255.240	
150.100.2.59	0011	1011	150.100.2.48	255.255.255.240	
150.100.2.60	0011	1100	150.100.2.48	255.255.255.240	
150.100.2.61	0011	1101	150.100.2.48	255.255.255.240	
150.100.2.62	0011	1110	150.100.2.48	255.255.255.240	

IP Address (Decimal)	Subnet Portion of Last Octet (Binary)	Host Portion of Last Octet (Binary)	Subnet Number	Subnet Mask	Notes
150.100.2.63	0011	1111	150.100.2.48	255.255.255.240	Subnet broadcast
150.100.2.64	010000	00	150.100.2.64	255.255.255.252	Subnet identifier
150.100.2.65	010000	01	150.100.2.64	255.255.255.252	
150.100.2.66	010000	10	150.100.2.64	255.255.255.252	
150.100.2.67	010000	11	150.100.2.64	255.255.255.252	Subnet broadcast
150.100.2.68	010001	00	150.100.2.68	255.255.255.252	Subnet identifier
150.100.2.69	010001	01	150.100.2.68	255.255.255.252	
150.100.2.70	010001	10	150.100.2.68	255.255.255.252	
150.100.2.71	010001	11	150.100.2.68	255.255.255.252	Subnet broadcast
150.100.2.72	010010	00	150.100.2.72	255.255.255.252	Subnet identifier
150.100.2.73	010010	01	150.100.2.72	255.255.255.252	
150.100.2.74	010010	10	150.100.2.72	255.255.255.252	
150.100.2.75	010010	11	150.100.2.72	255.255.255.252	Subnet broadcast
150.100.2.76	010011	00	150.100.2.76	255.255.255.252	Subnet identifier
150.100.2.77	010011	01	150.100.2.76	255.255.255.252	
150.100.2.78	010011	10	150.100.2.76	255.255.255.252	
150.100.2.79	010011	11	150.100.2.76	255.255.255.252	Subnet broadcast
150.100.2.80	0101	0000	150.100.2.80	255.255.255.240	Subnet identifier
150.100.2.81	0101	0001	150.100.2.80	255.255.255.240	
150.100.2.82	0101	0010	150.100.2.80	255.255.255.240	
150.100.2.83	0101	0011	150.100.2.80	255.255.255.240	
150.100.2.84	0101	0100	150.100.2.80	255.255.255.240	
150.100.2.85	0101	0101	150.100.2.80	255.255.255.240	
150.100.2.86	0101	0110	150.100.2.80	255.255.255.240	
150.100.2.87	0101	0111	150.100.2.80	255.255.255.240	
150.100.2.88	0101	1000	150.100.2.80	255.255.255.240	
150.100.2.89	0101	1001	150.100.2.80	255.255.255.240	
150.100.2.90	0101	1010	150.100.2.80	255.255.255.240	
150.100.2.91	0101	1011	150.100.2.80	255.255.255.240	
150.100.2.92	0101	1100	150.100.2.80	255.255.255.240	
150.100.2.93	0101	1101	150.100.2.80	255.255.255.240	
150.100.2.94	0101	1110	150.100.2.80	255.255.255.240	
150.100.2.95	0101	1111	150.100.2.80	255.255.255.240	Subnet broadcast
150.100.2.96	0110	0000	150.100.2.96	255.255.255.240	Subnet identifier
150.100.2.97	0110	0001	150.100.2.96	255.255.255.240	
150.100.2.98	0110	0010	150.100.2.96	255.255.255.240	
150.100.2.99	0110	0011	150.100.2.96	255.255.255.240	

IP Address (Decimal)	Subnet Portion of Last Octet (Binary)	Host Portion of Last Octet (Binary)	Subnet Number	Subnet Mask	Notes
150.100.2.100	0110	0100	150.100.2.96	255.255.255.240	
150.100.2.101	0110	0101	150.100.2.96	255.255.255.240	
150.100.2.102	0110	0110	150.100.2.96	255.255.255.240	
150.100.2.103	0110	0111	150.100.2.96	255.255.255.240	
150.100.2.104	0110	1000	150.100.2.96	255.255.255.240	
150.100.2.105	0110	1001	150.100.2.96	255.255.255.240	
150.100.2.106	0110	1010	150.100.2.96	255.255.255.240	
150.100.2.107	0110	1011	150.100.2.96	255.255.255.240	
150.100.2.108	0110	1100	150.100.2.96	255.255.255.240	
150.100.2.109	0110	1101	150.100.2.96	255.255.255.240	
150.100.2.110	0110	1110	150.100.2.96	255.255.255.240	
150.100.2.111	0110	1111	150.100.2.96	255.255.255.240	Subnet broadcast
150.100.2.112	0111	0000	150.100.2.112	255.255.255.240	Subnet identifier
150.100.2.113	0111	0001	150.100.2.112	255.255.255.240	
150.100.2.114	0111	0010	150.100.2.112	255.255.255.240	
150.100.2.115	0111	0011	150.100.2.112	255.255.255.240	
150.100.2.116	0111	0100	150.100.2.112	255.255.255.240	
150.100.2.117	0111	0101	150.100.2.112	255.255.255.240	
150.100.2.118	0111	0110	150.100.2.112	255.255.255.240	
150.100.2.119	0111	0111	150.100.2.112	255.255.255.240	
150.100.2.120	0111	1000	150.100.2.112	255.255.255.240	
150.100.2.121	0111	1001	150.100.2.112	255.255.255.240	
150.100.2.122	0111	1010	150.100.2.112	255.255.255.240	
150.100.2.123	0111	1011	150.100.2.112	255.255.255.240	
150.100.2.124	0111	1100	150.100.2.112	255.255.255.240	
150.100.2.125	0111	1101	150.100.2.112	255.255.255.240	
150.100.2.126	0111	1110	150.100.2.112	255.255.255.240	
150.100.2.127	0111	1111	150.100.2.112	255.255.255.240	Subnet broadcast; area boundary; area 8 ends
150.100.2.128	1000	0000	150.100.2.128	255.255.255.240	Subnet identifier; area boundary; area 17 starts
150.100.2.129	1000	0001	150.100.2.128	255.255.255.240	
150.100.2.130	1000	0010	150.100.2.128	255.255.255.240	
150.100.2.131	1000	0011	150.100.2.128	255.255.255.240	
150.100.2.132	1000	0100	150.100.2.128	255.255.255.240	
150.100.2.133	1000	0101	150.100.2.128	255.255.255.240	
150.100.2.134	1000	0110	150.100.2.128	255.255.255.240	
150.100.2.135	1000	0111	150.100.2.128	255.255.255.240	
150.100.2.136	1000	1000	150.100.2.128	255.255.255.240	

IP Address (Decimal)	Subnet Portion of Last Octet (Binary)	Host Portion of Last Octet (Binary)	Subnet Number	Subnet Mask	Notes
150.100.2.137	1000	1001	150.100.2.128	255.255.255.240	
150.100.2.138	1000	1010	150.100.2.128	255.255.255.240	
150.100.2.139	1000	1011	150.100.2.128	255.255.255.240	
150.100.2.140	1000	1100	150.100.2.128	255.255.255.240	
150.100.2.141	1000	1101	150.100.2.128	255.255.255.240	
150.100.2.142	1000	1110	150.100.2.128	255.255.255.240	
150.100.2.143	1000	1111	150.100.2.128	255.255.255.240	Subnet broadcast
150.100.2.144	1001	0000	150.100.2.144	255.255.255.240	Subnet identifier
150.100.2.145	1001	0001	150.100.2.144	255.255.255.240	
150.100.2.146	1001	0010	150.100.2.144	255.255.255.240	
150.100.2.147	1001	0011	150.100.2.144	255.255.255.240	
150.100.2.148	1001	0100	150.100.2.144	255.255.255.240	
150.100.2.149	1001	0101	150.100.2.144	255.255.255.240	
150.100.2.150	1001	0110	150.100.2.144	255.255.255.240	
150.100.2.151	1001	0111	150.100.2.144	255.255.255.240	
150.100.2.152	1001	1000	150.100.2.144	255.255.255.240	
150.100.2.153	1001	1001	150.100.2.144	255.255.255.240	
150.100.2.154	1001	1010	150.100.2.144	255.255.255.240	
150.100.2.155	1001	1011	150.100.2.144	255.255.255.240	
150.100.2.156	1001	1100	150.100.2.144	255.255.255.240	
150.100.2.157	1001	1101	150.100.2.144	255.255.255.240	
150.100.2.158	1001	1110	150.100.2.144	255.255.255.240	
150.100.2.159	1001	1111	150.100.2.144	255.255.255.240	Subnet broadcast
150.100.2.160	1010	0000	150.100.2.160	255.255.255.240	Subnet identifier
150.100.2.161	1010	0001	150.100.2.160	255.255.255.240	
150.100.2.162	1010	0010	150.100.2.160	255.255.255.240	
150.100.2.163	1010	0011	150.100.2.160	255.255.255.240	
150.100.2.164	1010	0100	150.100.2.160	255.255.255.240	
150.100.2.165	1010	0101	150.100.2.160	255.255.255.240	
150.100.2.166	1010	0110	150.100.2.160	255.255.255.240	
150.100.2.167	1010	0111	150.100.2.160	255.255.255.240	
150.100.2.168	1010	1000	150.100.2.160	255.255.255.240	
150.100.2.169	1010	1001	150.100.2.160	255.255.255.240	
150.100.2.170	1010	1010	150.100.2.160	255.255.255.240	
150.100.2.171	1010	1011	150.100.2.160	255.255.255.240	
150.100.2.172	1010	1100	150.100.2.160	255.255.255.240	
150.100.2.173	1010	1101	150.100.2.160	255.255.255.240	

IP Address (Decimal)	Subnet Portion of Last Octet (Binary)	Host Portion of Last Octet (Binary)	Subnet Number	Subnet Mask	Notes
150.100.2.174	1010	1110	150.100.2.160	255.255.255.240	
150.100.2.175	1010	1111	150.100.2.160	255.255.255.240	Subnet broadcast
150.100.2.176	101100	00	150.100.2.176	255.255.255.252	Subnet identifier
150.100.2.177	101100	01	150.100.2.176	255.255.255.252	
150.100.2.178	101100	10	150.100.2.176	255.255.255.252	
150.100.2.179	101100	11	150.100.2.176	255.255.255.252	Subnet broadcast
150.100.2.180	101101	00	150.100.2.180	255.255.255.252	Subnet identifier
150.100.2.181	101101	01	150.100.2.180	255.255.255.252	
150.100.2.182	101101	10	150.100.2.180	255.255.255.252	
150.100.2.183	101101	11	150.100.2.180	255.255.255.252	Subnet broadcast
150.100.2.184	101110	00	150.100.2.184	255.255.255.252	Subnet identifier
150.100.2.185	101110	01	150.100.2.184	255.255.255.252	
150.100.2.186	101110	10	150.100.2.184	255.255.255.252	
150.100.2.187	101110	11	150.100.2.184	255.255.255.252	Subnet broadcast
150.100.2.188	101111	00	150.100.2.188	255.255.255.252	Subnet identifier
150.100.2.189	101111	01	150.100.2.188	255.255.255.252	
150.100.2.190	101111	10	150.100.2.188	255.255.255.252	
150.100.2.191	101111	11	150.100.2.188	255.255.255.252	Subnet broadcast
150.100.2.192	1100	0000	150.100.2.192	255.255.255.240	Subnet identifier
150.100.2.193	1100	0001	150.100.2.192	255.255.255.240	
150.100.2.194	1100	0010	150.100.2.192	255.255.255.240	
150.100.2.195	1100	0011	150.100.2.192	255.255.255.240	
150.100.2.196	1100	0100	150.100.2.192	255.255.255.240	
150.100.2.197	1100	0101	150.100.2.192	255.255.255.240	
150.100.2.198	1100	0110	150.100.2.192	255.255.255.240	
150.100.2.199	1100	0111	150.100.2.192	255.255.255.240	
150.100.2.200	1100	1000	150.100.2.192	255.255.255.240	
150.100.2.201	1100	1001	150.100.2.192	255.255.255.240	
150.100.2.202	1100	1010	150.100.2.192	255.255.255.240	
150.100.2.203	1100	1011	150.100.2.192	255.255.255.240	
150.100.2.204	1100	1100	150.100.2.192	255.255.255.240	
150.100.2.205	1100	1101	150.100.2.192	255.255.255.240	
150.100.2.206	1100	1110	150.100.2.192	255.255.255.240	
150.100.2.207	1100	1111	150.100.2.192	255.255.255.240	Subnet broadcast
150.100.2.208	1101	0000	150.100.2.208	255.255.255.240	Subnet identifier
150.100.2.209	1101	0001	150.100.2.208	255.255.255.240	
150.100.2.210	1101	0010	150.100.2.208	255.255.255.240	

IP Address (Decimal)	Subnet Portion of Last Octet (Binary)	Host Portion of Last Octet (Binary)	Subnet Number	Subnet Mask	Notes
150.100.2.211	1101	0011	150.100.2.208	255.255.255.240	
150.100.2.212	1101	0100	150.100.2.208	255.255.255.240	
150.100.2.213	1101	0101	150.100.2.208	255.255.255.240	
150.100.2.214	1101	0110	150.100.2.208	255.255.255.240	
150.100.2.215	1101	0111	150.100.2.208	255.255.255.240	
150.100.2.216	1101	1000	150.100.2.208	255.255.255.240	
150.100.2.217	1101	1001	150.100.2.208	255.255.255.240	
150.100.2.218	1101	1010	150.100.2.208	255.255.255.240	
150.100.2.219	1101	1011	150.100.2.208	255.255.255.240	
150.100.2.220	1101	1100	150.100.2.208	255.255.255.240	
150.100.2.221	1101	1101	150.100.2.208	255.255.255.240	
150.100.2.222	1101	1110	150.100.2.208	255.255.255.240	
150.100.2.223	1101	1111	150.100.2.208	255.255.255.240	Subnet broadcast
150.100.2.224	1110	0000	150.100.2.224	255.255.255.240	Subnet identifier
150.100.2.225	1110	0001	150.100.2.224	255.255.255.240	
150.100.2.226	1110	0010	150.100.2.224	255.255.255.240	
150.100.2.227	1110	0011	150.100.2.224	255.255.255.240	
150.100.2.228	1110	0100	150.100.2.224	255.255.255.240	
150.100.2.229	1110	0101	150.100.2.224	255.255.255.240	
150.100.2.230	1110	0110	150.100.2.224	255.255.255.240	
150.100.2.231	1110	0111	150.100.2.224	255.255.255.240	
150.100.2.232	1110	1000	150.100.2.224	255.255.255.240	
150.100.2.233	1110	1001	150.100.2.224	255.255.255.240	
150.100.2.234	1110	1010	150.100.2.224	255.255.255.240	
150.100.2.235	1110	1011	150.100.2.224	255.255.255.240	
150.100.2.236	1110	1100	150.100.2.224	255.255.255.240	
150.100.2.237	1110	1101	150.100.2.224	255.255.255.240	
150.100.2.238	1110	1110	150.100.2.224	255.255.255.240	
150.100.2.239	1110	1111	150.100.2.224	255.255.255.240	Subnet broadcast
150.100.2.240	1111	0000	150.100.2.240	255.255.255.240	Subnet identifier
150.100.2.241	1111	0001	150.100.2.240	255.255.255.240	
150.100.2.242	1111	0010	150.100.2.240	255.255.255.240	
150.100.2.243	1111	0011	150.100.2.240	255.255.255.240	
150.100.2.244	1111	0100	150.100.2.240	255.255.255.240	
150.100.2.245	1111	0101	150.100.2.240	255.255.255.240	
150.100.2.246	1111	0110	150.100.2.240	255.255.255.240	
150.100.2.247	1111	0111	150.100.2.240	255.255.255.240	

IP Address (Decimal)	Subnet Portion of Last Octet (Binary)	Host Portion of Last Octet (Binary)	Subnet Number	Subnet Mask	Notes
150.100.2.248	1111	1000	150.100.2.240	255.255.255.240	
150.100.2.249	1111	1001	150.100.2.240	255.255.255.240	
150.100.2.250	1111	1010	150.100.2.240	255.255.255.240	
150.100.2.251	1111	1011	150.100.2.240	255.255.255.240	
150.100.2.252	1111	1100	150.100.2.240	255.255.255.240	
150.100.2.253	1111	1101	150.100.2.240	255.255.255.240	
150.100.2.254	1111	1110	150.100.2.240	255.255.255.240	
150.100.2.255	1111	1111	150.100.2.240	255.255.255.240	Subnet broadcast; area boundary; area 17 ends

APPENDIX B

IBM Serial Link Implementation Notes

The following discussions clarify some common misconceptions and points of confusion associated with half-duplex, full-duplex, and multipoint connections.

Half Duplex and Full Duplex Compared

There is often confusion with half duplex and full-duplex serial links. One reason is that there are several different contexts in which these two terms are used. These contexts include asynchronous line implementations, IBM Systems Network Architecture (SNA)-specific implementations, and data communications equipment (DCE) implementations. Each is addressed in the discussions that follow.

Asynchronous Line Definitions

Duplex as seen on asynchronous communication lines (and in terminal emulation software parameters) implies *full duplex* as it applies to the echoing of transmitted characters by a host back to a terminal. This is also referred to as *echoplex* mode. In this context, half-duplex mode involves no character echo. Some common misconfigurations of terminals and hosts follow:

- Full duplex specified on a terminal when the host is set for half duplex results in typing blind at the terminal.
- Half duplex specified on a terminal when the host is set for full duplex results in double characters on the terminal because the terminal displays entered characters if the terminal's configuration indicates that the host will not echo characters.

Note This interpretation of duplex does not apply in a router context.

IBM SNA-Specific Definitions

IBM's master glossary for VTAM, NCP, and NetView terms defines *duplex*, *full duplex*, and *half duplex* as follows:

- Duplex—In data communications, pertaining to a simultaneous two-way independent transmission in both directions. Synonymous with full duplex. Contrast with half duplex.
- Half duplex—In data communications, pertaining to an alternate, one-way-at-a-time, independent transmission. Contrast with duplex.

These definitions can be applied in two contexts that are the main source of duplex definition confusion.

- First, there is *full-duplex* and *half-duplex data transfer*. This typically applies to the ability or inability of data terminal equipment (DTE) to support simultaneous, two-way data flow. SNA PU 4 devices (front-end processors such as 3705, 3720, 3725, and 3745 devices) are capable of full-duplex data transfer. Each such device employs a separate data and control path into the control program's transmit and receive buffers.
- Some PU 2.1 devices are also capable of *full duplex data mode*, which is negotiable in the XID-3 format frame—unless the NCP PU definition statement DATMODE=FULL is specified. If FULL is specified, full-duplex mode is forced. PU 2s and PU 1s operate in *half-duplex data mode*.

DCE Definitions

Finally, there is *full duplex* and *half duplex* as it applies to the communication facility, or DCE. This is where the most technological advancement has been achieved with respect to half and full duplex. DCE installations primarily consist of channel service units (CSUs), data service units (DSUs), or modem devices, and a communications line. The modem can be synchronous or asynchronous and can be analog or digital. The communications line can be two-wire or four-wire and can be leased or switched (that is, dial-up).

Older modems are only capable of transmitting or receiving at a given time. When a DTE wants to transmit data using an older modem, the DTE asserts the Request To Send (RTS) signal to the modem. If the modem *is not* in receive mode, the modem enables its carrier signal in preparation for transmitting data and asserts Clear To Send (CTS). If the modem *is* in receive mode, its Data Carrier Detect (DCD) signal (that is, the carrier signal from the remote modem) is in the active state. Because DCD is in the active state, the modem does not activate the CTS signal, and the DTE does not transmit.

Contemporary modems are capable of transmitting and receiving simultaneously over two-wire or four-wire and leased or switched lines. One method uses multiple carrier signals at different frequencies, so that the local modem's transmit and receive signal as well as the remote modem's transmit and receive signal each has its own carrier frequency.

DTE equipment in an SDLC environment have configuration options that specify which mode of operation is supported by DCE equipment. The default parameters for most PU 2 devices are set for half duplex, although they can also support full-duplex operation. If the facility is capable of full duplex, RTS can be asserted at all times. If the facility supports half duplex, or is operating in a *multipoint* environment using modem-sharing devices (as opposed to multipoint provided by a Postal Telephone and Telegraph [PTT] or by a telephone company), RTS must only be asserted when transmitting. A full-duplex-capable communication facility that connects a PU 4 to a PU 2 device or to a PU 1 device (with each PU device specifying full-duplex DCE capability) experiences improved response time because of reduced turnaround delays.

Older PU 2 and PU 1 devices cannot be configured for full-duplex DCE mode. Also, because older PU 2 and PU 1 devices can only support half-duplex data transfer, transmit and receive data cannot be on the line at the same time (in contrast to a PU 4-to-PU 4 full-duplex exchange).

Multipoint Connections

Multipoint operation is a method of sharing a communication facility with multiple locations. The telephone company or PTT communications authorities offer two-wire and four-wire multipoint configurations for analog service (modem attachment) or four-wire for digital service (CSU/DSU attachment). Most implementations are master-polling, multiple-slave drop implementations. The

master only connects to one drop at a time. The switching takes place at a designated local exchange in proximity of the master DTE site. Some service providers offer analog multipoint services that support two-way simultaneous communication, which allows DTEs to be configured for permanent RTS.

Modem-sharing devices and line-sharing devices also provide multipoint capability. These implementations allow a single point-to-point link to be shared by multiple devices. Some of these devices have configurable ports for DTE or DCE operation, which allow for configurations that can accommodate multiple sites (called cascaded configurations). The main restriction of these devices is that when RTS is active, everyone else is locked out. You cannot configure DTEs for permanent RTS and must accept the turnaround delays associated with this mode of operation.

APPENDIX C

SNA Host Configuration for SRB Networks

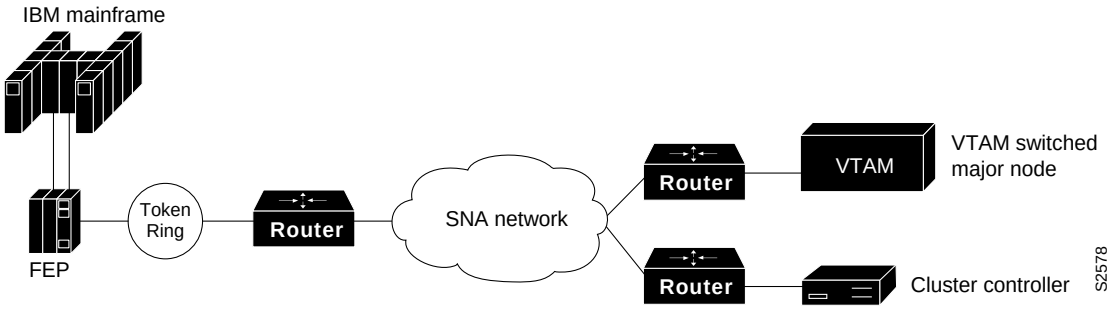
When designing source-route bridging (SRB) internetworks featuring routers and IBM Systems Network Architecture (SNA) entities, you must carefully consider the configuration of SNA nodes as well as routing nodes. This appendix provides examples that focus on three specific SNA devices:

- Front-end processors (FEPs)
- Virtual Telecommunications Access Method (VTAM)-switched major nodes
- 3174 cluster controllers

Figure C-1 illustrates a typical environment. Table C-1 through Table C-6 present the definition parameters for the devices shown in Figure C-1.

Note This material provides host-related configuration information pertinent to design material provided in Chapter 3"Designing SRB Internetworks."

Figure C-1 Typical SNA Host Environment



FEP Configuration

The parameters listed in Table C-1 through Table C-6 illustrate input to the Network Control Program (NCP) system generation process that runs in the host processor using the Network Definition Facility (NDF). The NDF is part of the ACF/NCP/System Support Program utility. The output produced by the generation process is a *load module* that runs in an FEP. Its typical size can be anywhere from a little under a 1 MB to more than 3 MB. The ACF/NCP/System Support Program utility is also used for loading and dumping an FEP.

The following tables outline relevant parameters for generating Token Ring resources. For more specific information, refer to the IBM manual SC30-3448, *NCP/SSP Resource Definition Reference*.

Table C-1 BUILD Definition Parameters

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
LOCALTO	1.5	Local ring acknowledgment timer (seconds).
REMOTTO	2.5	Remote ring acknowledgment timer (seconds).
MAXSESS	5000	Maximum amount of sessions for all attached resources.
MXRLINE	None	Maximum number of NTRI physical connections (Version 5.2.1 and earlier only).
MXVLINE	None	Maximum number of NTRI logical connections (Version 5.2.1 and earlier only).
T2TIMER	(*localt2, remott2, N3*)	(Version 5.R4 and later only.) Parameters specify a receiver acknowledgement/timer(T2) for local and remote Token Rings whether from peripheral or subarea nodes. Acceptable values: *localt2* range is 0 to 2.0 seconds; *remott2* range is 0 to 2.0 seconds; *N3* range is 1 to 127 (default is 2). The values for *localt2* and *remott2* should be 10 percent of value of the adjacent stations's T1 timer. *N3* specifies the maximum number of I-frames received without sending an acknowledgment for subarea connections.

The LUDRPOOL definition shown in Table C-2 specifies the number of peripheral resources required for the correct amount of control block storage to be reserved for new connections.

Table C-2 LUDRPOOL Definition Parameters

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
NUMTYP2	None	Maximum is 16000.
NUMILU	None	Required for LU Type 2.1 devices (independent LUs).

Table C-3 GROUP Definition Parameters

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
AUTOGEN	*Number*	Specifies the number of LINE/PU pairs for this group.
COMPOWN	Y	Twin FEP backup capable resource.
COMPSWP	Y	TIC portswap capable (hot backup).
COMPTAD	Y	TIC capable of IPL loading FEP.
DIAL	YES or NO	Applies to ECLTYPE parameter specifications. YES required for (LOGICAL,PERIPHERAL); NO required for all other combinations indicated in ECLTYPE specification.

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
ECLTYPE	(PHYSICAL,ANY)	Allows PU 4 and PU 2 devices to attach.
	(PHYSICAL, PERIPHERAL)	Allows PU 2 type devices only.
	(PHYSICAL, SUBAREA	Allows PU 4 devices only.
	(LOGICAL, PERIPHERAL)	Defines devices attaching as PU 2.
	(LOGICAL, SUBAREA	Defines devices attaching as PU 4.
LNCTL	SDLC	Required for NCP processing compatibility.
PHYPORT	None	Required for ECLTYPE LOGICAL only. Links this to a ECLTYPE PHYSICAL.
TIMER	error, ras, stap, or lstap	Entry points for NTRI timer routines.

Table C-4 LINE Definition Parameters

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
ADAPTER	TIC1	4-MB Token Ring interface.
	TIC2	4- or 16-MB Token Ring interface.
ADDRESS	1088 to1095	Range of valid addresses for TICs; only one specified per LINE definition.
BEACTO	52	Time in seconds the ring can beacon before TIC considers it down; maximum is 600.
LOCADD	4000*abbbbbbb*	Locally administered TIC address, where *a* is any value from 0 to 7; and *b* is any integer value from 0 to 9.
LOCALTO	1.5	V5R4; same as in BUILD, but only for PU 4 (LOGICAL, SUBAREA) devices; allows granularity for individual TICs for SUBAREA connections.
REMOTTO	2.5	V5R4 parameter; same as LOCALTO; see BUILD parameters in Table C-1.
T2TIMER	*localt2, remott2, N3*	V5.4 parameter; see BUILD parameters in Table C-1; can be defined in LINE definition only if a subarea node was defined in GROUP definition.
MAXTSL	2044 to 16732	Specifies maximum data in bytes that NTRI can transmit.; TIC1 maximum is 2044; TIC2 maximum at TRSPEED16 is 16732.
PORTADD	*Number*	For association of physical to logical ECLTYPEs. Matches physical or logical ECLTYPE specification.
RETRIES	*m, t, n, ml*	Where *m* = number of retries for remote ring sessions, *t* = pause between retry sequence, *n* = number of retry sequences, and *ml* = number of retries in a sequence for local ring sessions.
TRSPEED	4 or 16	TIC speed.

Table C-5 FEP Physical Unit (PU) Definition Parameters

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
ADDR	*aa*4000*bccccccc*	Destination service access point (DSAP) and MAC address for the PU of the Token Ring device in the FEP, where *aa* = the DSAP and is a nonzero hexadecimal multiple of 4; *b* = 0 to 7; *c* = 0 to 9. Enter 4000 as shown. Only specified if ECLTYPE defined in GROUP definition is one of the following: (LOG,SUB), (PHY,SUB), (PHY,ANY) only.
PUTYPE	1, 2, or 4	Depends on ECLTYPE: • For NTRI LOGICAL resources, only PUTYPE=2 is valid; for NTRI PHYSICAL resources, only PUTYPE=1 is valid • For NTRI PHYSICAL/SUBAREA LINES and PHYSICAL PERIPHERAL LINES, only PUTYPE=1 is valid. For NTRI LOGICAL PERIPHERAL LINES, only PUTYPE=2 is valid.
XID	NO or YES	Defines the ability of a PU to receive and respond to an XID while in normal disconnected mode; for NTRI PHYSICAL LINES, only NO is valid. For NTRI LOGICAL LINES, only YES is valid.

Table C-6 FEP Logical Unit (LU) Definition Parameter

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
LOCADDR	0	Specify this response only.

VTAM-Switched Major Node Definitions

Devices that are attached to Token Ring and communicate with an IBM host application must be defined via the VTAM access method associated with the host. These devices are seen as dial-in resources from the host side and are defined in a configuration component named *Switched Major Node*. Some common definitions used in network configurations are outlined in Table C-7 through Table C-9.

Table C-7 VBUILD Definition Parameter

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
TYPE	SWNET	Specifies a type of resource for VTAM; SWNET indicates switched major node type.

Table C-8 VTAM PU Definition Parameters

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
IDBLK	017	Typical values: • 017 = 3X74 • 05D = PC-base VTAM PU • 0E2 = Cisco SDLLC (registered with IBM)
IDNUM	*xxxxx*	Unique number identifying a device.
MAXOUT	1 to 7	Number of I-frames sent before acknowledgment is required.
MAXDATA	265	Indicates maximum number of bytes a PU 2 device can receive; ignored for PU 2.1, as this value is negotiable. Default for 3174 is 521.
PUTYPE	2	Only valid value.
XID	YES or NO	YES should be used for PU 2.1 devices. NO should be specified for any other device.

Table C-9 VTAM LU Definition Parameter

Parameter	Example Parameter Value or Range	Parameter Description and Implementation Notes
LOCADDR	2 through FF	Logical unit (LU) addresses attached to a PU.

3174 Cluster Controller Configuration Example

The following configuration was taken from 3174-13R cluster controller serial number 45362 connected to a Token Ring. These entries were used with a specific 3174 running on a 4-Mbps Token Ring. The configuration of this 3174-13R involved three specific configuration screens. Table C-10 through Table C-12 list the configuration line numbers, entries used, and descriptions of the configuration line. Where applicable, extended descriptions are included for configuration entries that are relevant to the requirements of the routed internetwork.

Note Of particular interest when configuring 3174 devices for a router-based SRB environment are configuration line items 106, 107, and 384 in configuration screen 2 (refer to Table C-11). These specify the required addresses and relevant Token Ring type for the cluster controller.

Table C-10 3174-13R Screen 1 Configuration Details

Configuration Line Number	Example Value	Parameter Description and Implementation Notes
98		Online test password.
99	TKNRNG	Description field.
100	13R	Model number.
101	7	Host attachment type.

Table C-11 3174-13R Screen 2 Configuration Details

Configuration Line Number	Example Value	Parameter Description and Implementation Notes
106	4000 2222 4444 04	The first 12 hexadecimal digits form the source MAC address of the cluster controller (4000 2222 4444); the last two digits are the source SAP (SSAP) for LLC2 (0x04 = SNA).
107	4000 0037 4501 04	The first 12 hexadecimal digits form the destination MAC address of the FEP (4000 0037 4501); the last two digits are the DSAP for LLC2 (0x04 for SNA).
108	0045362	Serial number of the cluster controller.
110	0	MLT storage support.
116	0	Individual port assignment.
121	01	Keyboard language.
123	0	Country extended code page support.
125	00000000	Miscellaneous options (A).
126	00000000	Miscellaneous options (B).
127	0 0	RTM definition.
132	0000	Alternate base keyboard selection.
136	0000	Standard keyboard layout.
137	0000	Modified keyboard layout.
138	0	Standard keypad layout.
141	A	Magnetic character set.
165	0	Compressed program symbols.
166	A	Attribute select keypad.
168	0	Additional extension; mode key definition.
173	0000	DFT options.
175	000000	DFT password.
179	000	Local format storage.
213	0	Between bracket printer sharing.
215	45362	PU identification.
222	0	Support for command retry.
382	0521	Maximum ring I-frame size; range of values is 265 to 2057.
383	2	Maximum number of I-frames 3174 will transmit before awaiting an acknowledgment (transmit window size).
384	0	Ring speed of the Token Ring network: • 0 = 4 Mbps • 1 = 16 Mbps normal token release • 2 = 16 Mbps early token release

Table C-12 3174-13R Screen 3 Configuration Details

Configuration Line Number	Example Value	Parameter Description and Implementation Notes
500	*0*	CSCM unique
501	TOSFNID	Network identifier
503	TOSFCTLR	LU name

SNA end stations implement Logical Link Control type 2 (LLC2) when attached to a local-area network (LAN). LLC2 implements the following:

- Timers
- Sequencing
- Error Recovery
- Windowing
- Guaranteed delivery
- Guaranteed connection

Figure C-2 illustrates how the T1 reply timer and error recovery operates for a 3174. Assume that the link between the two routers just failed. The following sequence characterizes the error recovery process illustrated in Figure C-2:

1. The 3174 sends a data frame and starts its T1 timer
2. The T1 timer expires after 1.6 seconds
3. The 3174 goes into error recovery
4. The 3174 sends an LLC request (a receiver ready with the poll bit on), which requests the 3745 to immediately acknowledge this frame
5. The 3174 starts its T1 timer
6. The T1 timer expires after 1.6 seconds

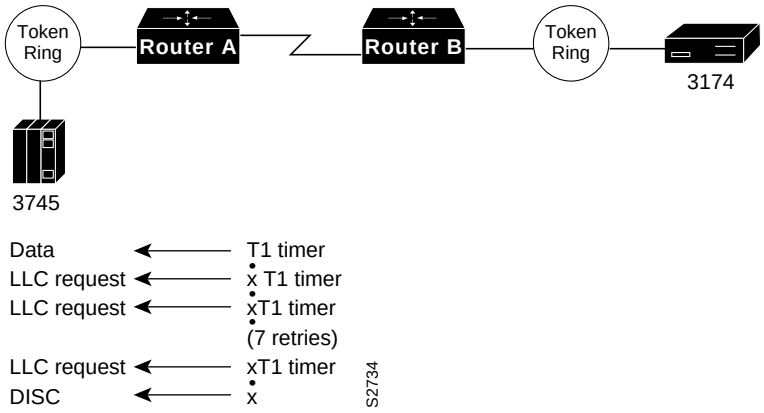
This operation is retried a total of seven times. The total elapsed time to disconnect the session is calculated as follows:

- The first attempt plus 7 retries times 1.6 seconds:

  = 8 x 1.6 seconds

  = 12.8 seconds

Figure C-2 T1 Timer and Error Recovery Process for 3174



APPENDIX D

SNA Host Configuration for SDLC Networks

This appendix outlines router implementation information related to the following topics:

- Front-end processor (FEP) configuration for SDLC links
- 3174 SDLC configuration worksheet example

Table D-1 outlines 3x74 SDLC point-to-point connection for AGS+, MGS, and CGS DCE appliques.

Table D-1 3x74 SDLC Point-to-Point Connection Support for AGS+, MGS, and CGS DCE Appliques

Controller Type	RS-232 DCE	RS-232 NRZI/DCE
3274 1st Generation		
• 3274-1C	Supported	Supported
3274 2nd Generation		
• 3274-21C	Not tested	Supported
3274 3rd Generation		
• 3274-31C	Supported	Not tested
• 3274-51C	Supported	Not tested
3274 4th Generation		
• 3274-41C	Need to tie DSR and DTR together on CU side, break DSR to router	Not tested
• 3274-61C	Same as 3274-41C	Supported
• Telex 274	Supported	Not tested
• Telex 1274	Supported	Not tested
DCA/IRMA 3274 emulation for DOS workstations	Not tested	Supported
DEC SNA gateway	Not tested	Supported
RS 6000 multiprotocol adapter	Not tested	Supported

Controller Type	RS-232 DCE	RS-232 NRZI/DCE
3174 Subsystem CUs		
• 3174-01R	Not tested	3174 ties pin 11 low (-11VDC) which forces the applique into DTE mode (DCE mode is set when pin 11 is set high)
• 3174-03R	Same as 3174-01R	Same as 3174-01R
• 3174-51R	Same as 3174-01R	Same as 3174-01R
3174 Establishment CUs		
• 3174-11R	Not tested	Supported
• 3174-13R	Same as 3174-11R	Not tested
• 3174-61R	Same as 3174-11R	Not tested
• 3174-91R	Same as 3174-11R	Supported
• Telex 1174	Supported	Not tested

FEP Configuration for SDLC Links

Table D-2 through Table D-5 present relevant parameter definitions for an FEP configured to operate within a router-based environment. These parameters are configured as part of the system generation process associated with the Network Control Program (NCP) on an IBM host.

Table D-2 FEP SDLC Configuration Example GROUP Parameter Listing and Definitions

Parameter	Example Values	Description and Implementation Notes
LNCTL	SDLC	Line control parameter that specifies link protocol.
REPLYTO	2	T1 timer; this timer specifies the reply timeout value for LINEs in this GROUP.

Table D-3 FEP SDLC Configuration Example LINE Parameter Listing and Definitions

Parameter	Example Values	Description and Implementation Notes
ADDRESS	(001,HALF)	The value 001 is the physical LINE interface address of the FEP. The second parameter specifies whether half- or full-duplex DATA transfer within the FEP is used. It also has an effect on the DUPLEX parameter. If FULL is specified here, DUPLEX defaults to FULL and attempts to modify this characteristic are ignored.
DUPLEX	HALF	This parameter specifies whether the communication line and modem constitute a half-duplex or full-duplex facility. If HALF is specified, the RTS modem signal is activated only when sending data. If FULL is specified, RTS always remains active. Refer to the ADDRESS parameter in this table.
NRZI	YES	Encoding for this line; options are NRZ or NRZI.

Parameter	Example Values	Description and Implementation Notes
RETRIES	(6,5,3)	Number of retries when REPLYTO expires. Entry options: (*m, t, n*) where *m* = number of retries, *t* = pause in seconds between retry cycles, and *n* = number of retry cycles to repeat. This example would retry 6 times—pausing the value of the REPLYTO between each RETRY (2 seconds per Table D-2), pause 5 seconds, and repeat this sequence 3 times for a total of 63 seconds. At the end of this period, the session is terminated.
PAUSE	2	The delay time in milliseconds between poll cycles. The cycle extends from the time NCP polls the first entry in the service order table to the moment polling next begins at the same entry. During this pause, any data available to send to the end station is sent. If end stations have data to send when polled and the time to send the data extends beyond the PAUSE parameter, the next poll cycle begins immediately.

Table D-4 FEP SDLC Configuration Example PU Parameter Listing and Definitions

Parameter	Example Values	Description and Implementation Notes
ADDR	C1	SDLC address of secondary end station.
MAXDATA	265	Maximum amount of data in bytes (including headers) that the UP can receive in one data transfer; that is, one entire PIU or a PIU segment.
MAXOUT	7	Maximum number of unacknowledged frames that NCP can have outstanding before requesting a response from the end station.
PASSLIM	7	Maximum number of consecutive PIU or PIU segments that NCP sends at one time to the end station represented by this PU definition.
PUTYPE	2	Specifies PU type. PU type 2 and 2.1 are both specified as PUTYPE=2.

Table D-5 FEP SDLC Configuration Example LU Parameter Listing and Definitions

Parameter	Example Values	Description and Implementation Notes
LOCADDR	2	LU address of devices connected to the end station PU.

3174 SDLC Configuration Worksheet

Table D-6 through Table D-8 present a configuration taken from an SDLC-connected 3174-91R cluster controller. The configuration of this 3174-91R involved three specific configuration screens. Table D-6 through Table D-8 list the configuration line numbers, entries used, and descriptions of the configuration lines for each screen. Where applicable, extended descriptions are included for configuration entries that are relevant to the requirements of the routed internetwork.

Table D-6 3174-91R Screen 1 Configuration Details

Configuration Line Number	Example Value	Parameter Description and Implementation Notes
98		Online test password

Configuration Line Number	Example Value	Parameter Description and Implementation Notes
99	TKNRNG	Description field
100	91R	Model number
101	2	Host attachment type: • 2 = SDLC • 5 = SNA (channel-attached) • 7 = Token Ring network

Note Configuration line items 104, 313, 317, 340, and 340 in configuration Screen 2 (refer to Table D-7) are of particular interest when configuring 3174 devices for a router-based SDLC environment. These lines specify the required SDLC address and relevant SDLC options for the cluster controller.

Table D-7 3174-91R Screen 2 Configuration Details

Configuration Line Number	Example Value	Parameter Description and Implementation Notes
104	C2	Specifies the cluster controller SDLC address. It is the same address that you configure on the router's serial line interface. It also represents the PU address of the controller. In multipoint environments, multiple SDLC addresses may be specified on a single serial interface.
108	0045448	Serial number of the cluster controller.
110	0	MLT storage support.
116	0	Individual port assignment.
121	01	Keyboard language.
123	0	Country extended code page support.
125	00000000	Miscellaneous options (A).
126	00000000	Miscellaneous options (B).
127	0 0	RTM definition.
132	0000	Alternate base keyboard selection.
136	0000	Standard keyboard layout.
137	0000	Modified keyboard layout.
138	0	Standard keypad layout.
141	A	Magnetic character set.
150	0	Token Ring network gateway controller.
165	0	Compressed program symbols.
166	A	Attribute select keypad.
168	0	Additional extension; mode key definition.
173	0000	DFT options.
175	000000	DFT password.
179	000	Local format storage.

Configuration Line Number	Example Value	Parameter Description and Implementation Notes
213	0	Between-bracket printer sharing.
215	45448	PU identification.
220	0	Alert function.
310	0	Connect dataset to line operation.
313	0	NRZ = 0; NRZI = 1.
317	0	Telecommunications facility: • 0 = Nonswitched • 1 = Switched (dial-up)
318	0	Full/half speed transmission; 0 = full speed, 1 = half speed. Controls speed of modem; can be used in areas where line conditions are poor.
340	0	RTS control options: • 0 = Controlled RTS (for LSD/MSD operation) • 1 = Permanent RTS (improves performance) • 2 = BSC (not valid for SDLC operation)
365	0	X.21 switched-host DTE connection.
370	0	Maximum inbound I-frame size: • 0 = 265 bytes • 1 = 521 bytes (recommended for better performance)

Table D-8 3174-91R Screen 3 Configuration Details

Configuration Line Number	Example Value	Parameter Description and Implementation Notes
500	*0*	Central Site Change Management (CSCM) unique
501	*xxxxxxxx*	Network identifier
503	*xxxxxxxx*	LU name (for CSCM)

APPENDIX E

References and Recommended Reading

Books and Periodicals

Albrightson, R., "Enhanced IGRP—A Fast Routing Protocol Based on Distance Vectors." *Interop+Networld Engineers* conference proceedings, May 1994.

Apple Computer, Inc. *AppleTalk Network System Overview.* Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.; 1989.

Apple Computer, Inc. *Planning and Managing AppleTalk Networks*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.; 1991.

Black, U. *Data Networks: Concepts, Theory and Practice.* Englewood Cliffs, New Jersey: Prentice Hall; 1989.

Black, U. Physical Level Interfaces and Protocols. Los Alamitos, California: IEEE Computer Society Press; 1988.

Case, J.D., J.R. Davins, M.S. Fedor, and M.L. Schoffstall. "Introduction to the Simple Gateway Monitoring Protocol." IEEE Network: March 1988.

Clark, W. "SNA Internetworking." *ConneXions: The Interoperability Report*, Vol. 6, No. 3: March 1992.

Coltun, R. "OSPF: An Internet Routing Protocol." *ConneXions: The Interoperability Report*, Vol. 3, No. 8: August 1989.

Comer, D.E. *Internetworking with TCP/IP: Principles, Protocols, and Architecture*, Vol. I, 2nd ed. Englewood Cliffs, New Jersey: Prentice Hall; 1991.

De Prycker, Martin. *Asynchronous Transfer Mode Solution for Broadband ISDN*. Prentice-Hall, Inc.; 1994.

Ferrari, Dominico. *Computer Systems Performance Evaluation.* Prentice-Hall, Inc.; 1978.

Davidson, J. *An Introduction to TCP/IP.* New York, New York: Springer-Verlag; 1992.

Garcia-Luna-Aceves, J.J. "Loop-Free Routing Using Diffusing Computations." *IEEE/ACM Transactions on Networking*, Vol. 1, No. 1, Feb 1993.

Green, J.K. *Telecommunications*, 2nd ed. Homewood, Illinois: Business One Irwin; 1992.

IBM. *Token-Ring Problem Determination Guide*. SX27-3710-04, 1990.

Jones, Nancy E.H., and Kosiur, Dave. *Macworld Networking Handbook*. IDG Books Worldwide, Inc.; 1992.

Kousky, K. "Bridging the Network Gap." *LAN Technology*, Vol. 6, No. 1: January 1990.

Lippis, N. "The Internetwork Decade." *Data Communications*, Vol. 20, No. 14: October 1991.

Malamud, C. *Analyzing DECnet/OSI Phase V.* Van Nostrand Reinhold; 1991.

Malamud, C. *Analyzing Novell Networks*. Van Nostrand Reinhold; 1990.

Malamud, C. *Analyzing Sun Networks.* Van Nostrand Reinhold; 1992.

Martin, J. *SNA: IBM's Networking Solution*. Englewood Cliffs, New Jersey: Prentice Hall; 1987.

Medin, M. "The Great IGP Debate—Part Two: The Open Shortest Path First (OSPF) Routing Protocol." *ConneXions: The Interoperability Report*, Vol. 5, No. 10: October 1991.

Meijer, A. *Systems Network Architecture: A tutorial*. New York, New York: John Wiley & Sons, Inc.; 1987.

Miller, M.A. *LAN Protocol Handbook*. San Mateo, California: M&T Books; 1990.

Miller, M.A. *LAN Troubleshooting Handbook*. San Mateo, California: M&T Publishing; 1989.

Minoli, D. and M. Vitella. *ATM and Cell Relay Service for Corporate Networks*. New York, New York: McGraw-Hill, Inc.; 1994.

Perlman, R. *Interconnections: Bridges and Routers.* Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.; 1992.

Perlman, R. and R. Callon. "The Great IGP Debate—Part One: IS-IS and Integrated Routing." *ConneXions: The Interoperability Report*, Vol. 5, No. 10: October 1991.

Rose, M.T. *The Open Book: A Practical Perspective on OSI*. Englewood Cliffs, New Jersey: Prentice Hall; 1990.

Rose, M.T. *The Simple Book: An Introduction to Management of TCP/IP-based Internets.* Englewood Cliffs, New Jersey: Prentice Hall; 1991.

Ross, F.E. "FDDI—A Tutorial." *IEEE Communications Magazine*, Vol. 24, No. 5: May 1986.

Schlar, S.K. *Inside X.25: A Manager's Guide*. New York, New York: McGraw-Hill, Inc.; 1990.

Schwartz, M. *Telecommunications Networks: Protocols, Modeling, and Analysis*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.; 1987.

Sherman, K. *Data Communications: A User's Guide.* Englewood Cliffs, New Jersey: Prentice Hall; 1990.

Sidhu, G.S., R.F. Andrews, and A.B. Oppenheimer. *Inside AppleTalk*, 2nd ed. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.; 1990.

Spragins, J.D. et al. *Telecommunications Protocols and Design*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.; 1991.

Stallings, W. *Data and Computer Communications*. New York, New York: Macmillan Publishing Company; 1991.

Stallings, W. *Handbook of Computer-Communications Standards*, Vols. 1–3. Carmel, Indiana: Howard W. Sams, Inc.; 1990.

Stallings, W. *Local Networks*, 3rd ed. New York, New York: Macmillan Publishing Company; 1990.

Sunshine, C.A. (ed.). *Computer Network Architectures and Protocols*, 2nd ed. New York, New York: Plenum Press; 1989.

Tannenbaum, A.S. *Computer Networks*, 2nd ed. Englewood Cliffs, New Jersey: Prentice Hall; 1988.

Terplan, K. *Communication Networks Management*. Englewood Cliffs, New Jersey: Prentice Hall; 1992.

Zaumen, W.T. and Garcia-Luna-Aceves, J.J. “Dynamics of Link-State and Loop-Free Distance-Vector Routing Alogrithms.” *Journal of Internetworking*, Wiley, Vol. 3, December 1992

Zimmerman, H. “OSI Reference Model—The ISO Model of Architecture for Open Systems Interconnection.” *IEEE Transactions on Communications* COM-28, No. 4: April 1980.

“Dynamics of Shortest-Path Routing Algorithms.” *ACM Computer Communication Review,* Vol. 21, No. 4, Sept. 1991

Technical Publications and Standards

Advanced Micro Devices. *The Supernet Family for FDDI.* Technical Manual Number 09779A. Sunnyvale, California; 1989.

———. *The Supernet Family for FDDI*. 1989 Data Book Number 09734C. Sunnyvale, California; 1989.

American National Standards Institute X3T9.5 Committee. *FDDI Station Management (SMT)*. Rev. 6.1; March 15, 1990.

———. Revised Text of ISO/DIS 8802/2 for the Second DIS Ballot, “Information Processing Systems—Local Area Networks.” Part 2: Logical Link Control. 1987-01-14.

———. T1.606. Integrated Services Digital Network (ISDN)—Architectural Framework and Service Description for Frame-Relaying Bearer Service; 1990.

———. T1.617. Integrated Services Digital Network (ISDN)—Signaling Specification for Frame Relay Bearer Service for Digital Subscriber Signaling System Number 1 (DSS1); 1991.

———. T1.618. Integrated Services Digital Network (ISDN)—Core Aspects of Frame Protocol for Use with Frame Relay Bearer Service; 1991.

Consultative Committee for International Telegraph and Telephone. *CCITT Data Communications Networks—Services and Facilities, Terminal Equipment and Interfaces, Recommendations X.1–X.29.* Yellow Book, Vol. VIII, Fascicle VIII.2; 1980.

———. *CCITT Data Communications Networks—Interfaces, Recommendations X.20–X.32.* Red Book, Vol. VIII, Fascicle VIII.3; 1984.

DDN Protocol Handbook. Four volumes; 1989.

Digital Equipment Corporation, Intel Corporation, Xerox Corporation. *The Ethernet, A Local-Area Network, Data Link Layer and Physical Layer Specifications.* Ver. 2.0; November 1982.

Garcia-Luna-Aceves, J.J. “A Unified Approach to Loop-Free Routing Using Distance Vectors or Link States.” ACM 089791-332-9/89/0009/0212, pp. 212–223; September 1989.

International Business Machines Corporation. ACF/NCP/VS network control program, system support programs: general information. GC30-3058.

———. *Advanced Communications Function for VTAM (ACF/VTAM), general information: introduction*. GS27-0462.

———. *Advanced Communications Function for VTAM, general information: concepts. GS27-0463.*

———. *Network Problem Determination Application: general information* GC34-2010.

———. *NCP/SSP Resource Definition Reference* SC30-3448.

———. *Synchronous Data Link Control: general information*. GA27-3093.

———. *Systems Network Architecture: concepts and products.* GC30-3072.

———. *Systems Network Architecture: technical overview.* GC30-3073-1; 1985.

———. *Token-Ring Network Architecture Reference*. SC30-3374.

———. *Tuning and Problem Analysis for NCP SDLC Devices*.

Novell, Inc. IPX Router Specification, Version 1.10. Part Number 107-000029-001. October 16, 1992.

———. NetWare Link Services Protocol (NLSP) Specification, Revision 0.9. Part Number 100-001708-001. March 1993.

StrataCom. *Frame Relay Specification with Extensions*. 001-208966, Rev.1.0; September 18, 1990.

Xerox Corporation. *Internet Transport Protocols*. XNSS 029101; January 1991.

INDEX

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z