14 ROUTING

"R$\overline{\text{oo}}$'•ting" is what fans do at a football game, what pigs do for truffles under oak trees in the Vaucluse, and what nursery workers intent on propagation do to cuttings from plants. "Rou´•ting" is how one creates a beveled edge on a tabletop or sends a corps of infantrymen into full-scale, disorganized retreat. Either pronunciation is correct for *routing*, which refers to the process of discovering, selecting, and employing paths from one place to another (or to many others) in a network.

The British prefer the spelling routeing, *perhaps to distinguish what happens in networks from what happened to the British in New Orleans in 1814. Since the* Oxford English Dictionary *is much heavier than any dictionary of American English, British English generally prevails in the documents produced by ISO and CCITT; wherefore, most of the international standards for routing protocols use the* routeing *spelling. Since this spelling would be unfamiliar to many readers, we use* routing *in this book, with apologies to our friends in the British Standards Institute.*

A simple definition of routing is "learning how to get from here to there."[1] In some cases, the term *routing* is used in a very strict sense to refer *only* to the process of obtaining and distributing information ("learning"), but not to the process of using that information to actually get

1. This is an application of the classic definition of a route by Schoch (see Chapter 5), augmented by Radia Perlman's (1992a) very practical observation that routes are both source- and destination-*dependent*: knowing how to get there isn't enough; you have to know where you are starting from ("here") as well as where you are going ("there").

from one place to another (for which a different term, *forwarding*, is reserved). Since it is difficult to grasp the usefulness of information that is acquired but never used, this chapter employs the term *routing* to refer in general to all the things that are done to discover and advertise paths from here to there and to actually move packets from here to there when necessary. The distinction between routing and forwarding is preserved in the formal discussion of the functions performed by OSI end systems and intermediate systems, in which context the distinction is meaningful.

Source Routing and "Hop-by-Hop" Routing

The routing operations of finding out how to get from here to there, and then actually getting from here to there, can be done in two (very different) basic ways. In *source routing*, all the information about how to get from here to there is first collected at the source ("here"), which puts it into the packets that it launches toward the destination ("there"). The job of the intervening network (with its collection of links and intermediate systems) is simply to read the routing information from the packets and act on it faithfully. In *hop-by-hop routing*, the source is not expected to have all the information about how to get from here to there; it is sufficient for the source to know only how to get to the "next hop" (perhaps an intermediate system to which it has a working link), and for that system to know how to get to the *next* hop, and so on until the destination is reached. The job of the intervening network in this case is more complicated; it has only the address of the destination (rather than a complete specification of the route by the source) with which to figure out the best "next hop" for each packet.

Consider an example, in which "here" is your home in Hopkinton, Massachusetts (U.S.A.), and "there" is Blueberry Hill Inn in Goshen, Vermont. If you sit down at home with a set of road maps and figure out exactly which roads and highways connect Hopkinton and Goshen, plotting the route you will follow along this road to that interchange to this junction (etc.), and then get in your car and actually drive along precisely that route to Blueberry Hill, you are performing source routing: if you were a packet, an ordered list of identifiers for links (roads) and intermediate systems (junctions and interchanges) would be encoded in your protocol header (see Chapter 13). If, on the other hand, you simply climb into your car and begin driving, stopping at every intersection to ask directions or examine the signposts, you are performing hop-by-hop routing: if you were a packet, the identification of your origin (Hopkinton)

and final destination (Blueberry Hill) would be encoded in your protocol header. In the first case, your ability to actually get to Blueberry Hill depends on the accuracy of the maps you used and whether or not any of the roads you have selected are closed for repairs; in the second case, it depends on finding enough information at every intersection to enable you to pick the right road to follow to the next.

For the most part, routing in OSI and TCP/IP networks today is hop-by-hop. Source routing has recently emerged as an important component of a new set of routing capabilities (for both OSI and TCP/IP networks) that support complex *policies* governing the paths that packets are permitted to take when more than one organization owns or administers the equipment and facilities that intervene between "here" and "there." These issues are, however, beyond the scope of this chapter, which concentrates on the current hop-by-hop architecture of OSI and TCP/IP routing, and which also omits the very different concerns of routing in virtual-circuit (connection-oriented) networks. For a detailed and comprehensive description of the theory and practice of routing, including the topics that are beyond the scope of this book, readers are encouraged to consult *Interconnections: Bridges and Routers* (Perlman 1992a).

Routing Principles

The principal criterion of successful routing is, of course, *correctness* (you do in fact want to get to Blueberry Hill, not Cranberry Bog), but it is not the only criterion. You might prefer to take the most direct route (the one that takes the least time and uses the least fuel), the most reliable route (the one that is not likely to be closed by a heavy snowfall), the most scenic route (the one that follows pleasant country roads rather than busy highways), the least expensive route (the one that follows freeways rather than toll roads), or the safest route (the one that avoids the army's missile testing range). In its most general form, *optimal* routing involves forwarding a packet from source to destination using the "best" path. What constitutes the "best" path can, of course, become quite a complicated question, as this example shows; networks, like the highway system, have variable costs, transit restrictions, delay characteristics, and residual error rates, and all of these can be more or less important in the determination of what "best" means for a particular source and destination or for a particular packet.

The principal objective of an open systems routing architecture is not, therefore, to achieve "optimal" routing—such a thing does not exist

in the abstract. Such an architecture must nevertheless be based on principles that account for what is happening in the real open systems world of today and tomorrow, in which computers are being connected to networks at a rate that more than doubles the number of systems connected to the worldwide (OSI and TCP/IP) Internet each year. These computers will be connected using a variety of local, metropolitan, and wide area networking technologies; the topology of interconnection will change as computers and the links between them are added and deleted; the networks will cross every conceivable national and international boundary; and the computers and networks will be administered by different organizations, both public and private, each of which may impose rules (policies) governing (and safeguarding) their use.

These observations suggest that an open systems routing architecture should

- Scale well
- Support many different subnetwork types and multiple qualities of service
- Adapt to topology changes quickly and efficiently (i.e., with minimal overhead and complexity)
- Provide controls that facilitate the "safe" interconnection of multiple organizations

It is not likely that the manual administration of static routing tables (the earliest medium for the maintenance of internetwork routes, in which a complete set of fixed routes from each system to every other system was periodically—often no more frequently than once a week—loaded into a file on each system) will satisfy these objectives for a network connecting more than a few hundred systems. A routing scheme for a large-scale open systems network must be dynamic, adaptive, and decentralized; be capable of supporting multiple paths offering different types of service; and provide the means to establish "trust, firewalls, and security" across multiple administrations (ISO/IEC TR 9575: 1990).

OSI Routing Architecture

The architecture of routing in OSI is basically the same as the architecture of routing in other connectionless (datagram) networks, including TCP/IP. As usual, however, the conceptual framework and terminology of OSI are more highly elaborated than those of its roughly equivalent peers, and thus, it is the OSI routing architecture that gets the lion's share of attention in this chapter. Keep in mind that most of what is said about the OSI routing architecture applies to hop-by-hop connectionless open systems routing in general.

The OSI routing scheme consists of:

- A set of *routing protocols* that allow end systems and intermediate systems to collect and distribute the information necessary to determine routes
- A *routing information base* containing this information, from which routes between end systems can be computed[2]
- A *routing algorithm* that uses the information contained in the routing information base to derive routes between end systems

End systems (ESs) and intermediate systems (ISs)[3] use routing protocols to distribute ("advertise") some or all of the information stored in their locally maintained routing information base. ESs and ISs send and receive these routing *updates* and use the information that they contain (and information that may be available from the local environment, such as information entered manually by an operator) to modify their routing information base.

The routing information base consists of a table of entries that identify a *destination* (e.g., a network service access point address); the subnetwork over which packets should be forwarded to reach that destination (also known as the *next hop*, or "next-hop subnetwork point of attachment address"); and some form of *routing metric*, which expresses one or more characteristics of the route (its delay properties, for example, or its expected error rate) in terms that can be used to evaluate the suitability of this route, compared to another route with different properties, for conveying a particular packet or class of packets. The routing information base may contain information about more than one "next hop" to the same destination if it is important to be able to send packets over different paths depending on the way in which the "quality of service" specified in the packet's header corresponds to different values of the routing metric(s).

The routing algorithm uses the information contained in the routing information base to compute actual routes ("next hops"); these are collectively referred to as the *forwarding information base*. It is important to recognize that the routing information base is involved in computations that take place in the "background," independent of the data traffic

2. Like the directory information base, the routing information base is an abstraction; it doesn't exist as a single entity. The routing information base can be thought of as the collective (distributed) wisdom of an entire subsystem concerning the routing-relevant connectivity among the components of that subsystem.

3. The terms *end system* and *intermediate system* will appear so frequently in this chapter's discussion of routing that the authors feel justified in resorting for the most part to the use of the acronyms *ES* and *IS* in their place.

flowing between sources and destinations at any given moment; but the forwarding information base is involved in the real-time selection of an outgoing link for every packet that arrives on an incoming link and must therefore be implemented in such a way that it does not become a performance-killing bottleneck in a real-world intermediate system (router).

Figure 14.1 illustrates the decomposition of the OSI routing function as it is represented in ISO/IEC TR 9575.

No system—certainly not an end system, which is supposed to be devoted primarily to tasks other than routing—can maintain a routing information base containing all the information necessary to specify routes from any "here" to any "there" in the entire global Internet. Neither is it possible to design a single routing protocol that operates well both in local environments (in which it is important to account quickly for changes in the local network topology) and in wide area environments (in which it is important to limit the percentage of network bandwidth that is consumed by "overhead" traffic such as routing updates). The OSI routing architecture is consequently hierarchical, and is divided into three functional tiers:

1. *End-system to intermediate-system routing* (host-to-router), in which the principal routing functions are discovery and redirection.
2. *Intradomain intermediate-system to intermediate-system routing* (router-to-router), in which "best" routes between ESs within a single administrative domain are computed. A single routing algorithm is used by all ISs within a domain.

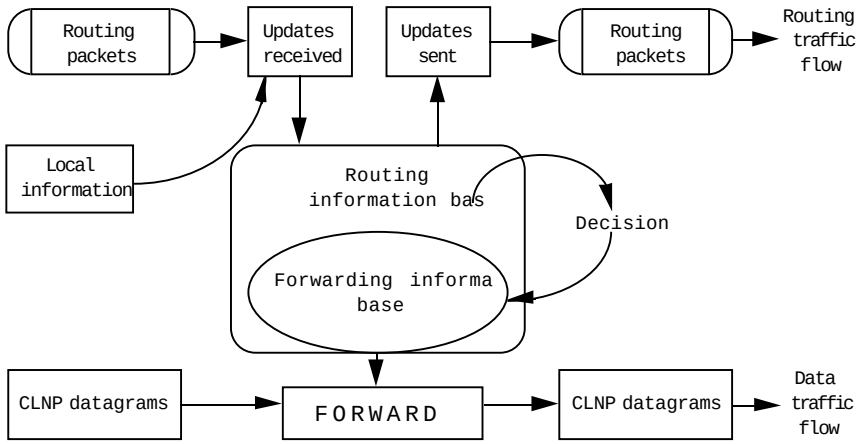

FIGURE 14.1 Decomposition of the OSI Routing Function

3. *Interdomain intermediate-system to intermediate-system routing* (router-to-router), in which routes are computed between administrative domains.

In Figure 14.2, end systems discover and communicate with the intermediate systems to which they are directly connected (by dedicated or dial-up point-to-point links or by multiaccess local or metropolitan area networks) in the outermost level of the hierarchy; intermediate systems communicate with other intermediate systems within a single routing domain in the levels of the hierarchy next closest to the center; and in the center, intermediate systems communicate with other intermediate systems across routing domain boundaries.

The decomposition illustrated in Figure 14.2 is not arbitrary. At each level of the hierarchy, a different set of imperatives governs the choices that are available for routing algorithms and protocols. In the OSI routing architecture, end systems are not involved in the distribution of routing information and the computation of routes, and hence, the participation of end systems in routing is limited to asking and answering the question "Who's on this subnetwork with me?" (On broad-

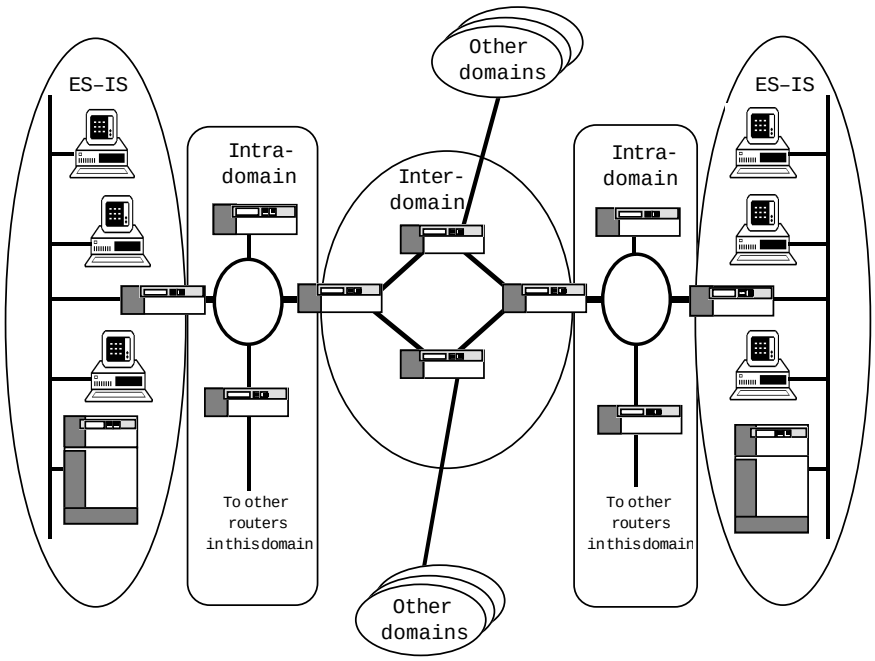

FIGURE 14.2 OSI Routing Architecture

cast subnetworks such as most local area networks, this inquiry typically begins with the more or less Cartesian assertion "I broadcast [multicast], therefore I am . . .")

Within a single routing domain, the hegemony of a single administration (and a correspondingly consistent set of routing policies) argues in favor of using a single routing protocol which provides every intermediate system with complete knowledge of the topology of the routing domain. (See "Intradomain Routing in OSI," later in this chapter.) Between routing domains that may be controlled by different (possibly even antagonistic) administrations, issues of security (including control over the extent to which information about the topology of one domain is propagated to other domains) outweigh most others, and the argument in favor of distributing complete topology information to all intermediate systems, so compelling when selecting an intradomain routing protocol, misfires for the very reason that concealing or witholding information is often as important as distributing it. It is important to recognize that the analysis that leads to one conclusion in the intradomain context does not necessarily hold when it is transplanted to the interdomain context.

Fortunately, with respect to routing, it has not been difficult to refute the simplistic argument that if link-state routing is the best choice for the intradomain level of the routing hierarchy it must be the best choice for the interdomain level. Ten years ago, however, a similarly simplistic argument destroyed the opportunity for OSI to standardize one of the best features of the TCP/IP internetwork architecture—the combination of a connectionless (datagram) internetwork protocol (which could be operated efficiently over any underlying network technology, whether based on datagrams or virtual circuits) with a connection-oriented end-to-end transport protocol (which made everything "come out even" at the hosts, or end systems). The OSI position at that time was that a connection-oriented service at the transport layer "naturally" mapped to a connection-oriented service at the network layer, as if this were something inherent in the very architecture of a layered model. The OSI community wasted years dealing with this red herring, which was intended to divert attention from the fact that a large segment of the OSI community believed that the service provided by the network layer was *an end-to-end transport service. The TCP/IP community, unencumbered by such nonsense, happily expanded to fill the resulting vacuum.*

End System to Intermediate System Routing "ES-IS" routing establishes connectivity and reachability among ESs and ISs attached to the same (single) subnetwork. Limiting the scope of routing in this manner allows

an ES to play a simple role in the overall routing process and leaves most of the ES's resources available to support end-user applications (which is, presumably, the *raison d'être* of an ES). ESs are commonly attached to *multiaccess subnetworks*, such as IEEE 802 local area networks (LANs) and metropolitan area networks (MANs), creating topologies that are both highly connected and densely populated (with ESs); the protocols and algorithms that are appropriate for routing in this environment are very different from those that are appropriate for routing in the wide area environments served by intermediate system to intermediate system routing.

At this level of routing, the two critical (closely related) concerns are discovery (who is out there?) and reachability (with whom is it possible to communicate?). Within a single subnetwork, an ES is one "hop" away from any ES or IS connected to the same subnetwork, so the only information an ES needs in order to reach either destination ESs on the same subnetwork or ISs that will forward packets to destination ESs on other subnetworks is the "hardware interface" or *subnetwork point of attachment* (SNPA) addresses of the ESs and ISs attached to the subnetwork.

Intradomain Intermediate System to Intermediate System Routing "IS-IS" routing establishes connectivity among intermediate systems within a single authority, the *administrative domain*. An administrative domain is composed of one or more *routing domains*. Each routing domain consists of a set of ISs and ESs; ISs within a routing domain use the same routing protocol, routing algorithm, and routing metrics.

At this level of routing, the critical concern is the selection and maintenance of best paths among systems within the administrative domain. ISs are concerned about route optimization with respect to a variety of metrics and about the trade-off between (1) the cost of distributing and maintaining routing information (which increases as the granularity of the information approaches "a separate route for every source/destination pair for every value of the routing metric[s]") and (2) the cost of actually sending data over a particular route (which increases if the available routing information causes data to be sent over a "suboptimal" route).

Interdomain Intermediate System to Intermediate System Routing Interdomain IS-IS routing establishes communication among different administrative domains, enabling them to control the exchange of information "across borders." In most circumstances, it is common to think of routing as something that tries to make it "as easy as possible" for two systems to communicate, regardless of what may lie between them. In-

terdomain routing, on the other hand, plays the paradoxical role of facilitating communication among open systems for which communication is a (politically) sensitive activity, involving issues of cost, accountability, transit authorization, and security that can produce highly counterintuitive answers to what look like simple technical questions.[4]

At this level of routing, the critical concern is the maintenance and enforcement of policies that govern, for example, the willingness of an administrative domain to (1) act as a transit domain for traffic originating from and destined for other administrative domains, (2) receive information from sources outside the administrative domain and deliver them to destinations within the administrative domain, and (3) forward information from within the administrative domain to destinations outside the administrative domain. Policies concerning 1, 2, and 3 can be derived on the basis of cost, access control, and regulatory concerns.

The hierarchical relationship of the OSI routing protocols is depicted in Figure 14.3.

Within the OSI routing framework, it is possible for different routing domains within a single administrative domain to run different

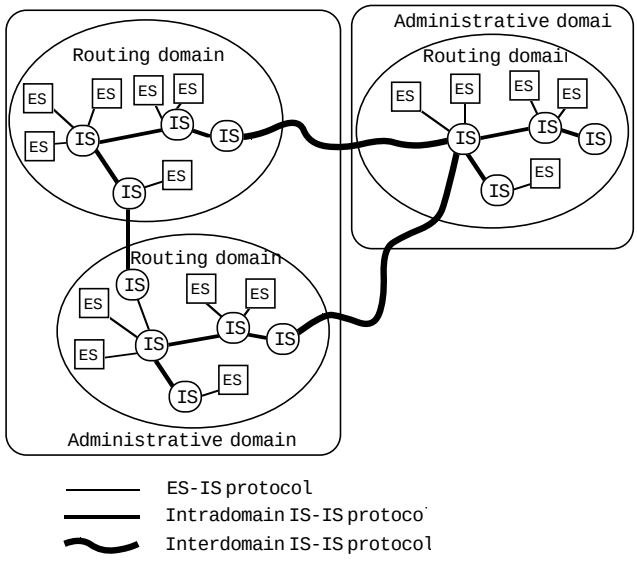

FIGURE 14.3 Hierarchical Relationship of OSI Routing Protocols

4. Within the purview of a single network administration, it is considered to be a very good and useful thing for the network to automatically reconfigure itself to route traffic around a failed link onto an alternate path. In an interdomain configuration involving mu-

intradomain routing protocols, and it is also possible to operate different ES-IS protocols within different areas of the same routing domain. At present, however, OSI defines only one standard routing protocol for each of the three levels of the hierarchy.

TCP/IP Routing Architecture

Through a process of evolution—in which some of the ideas that led to features of the OSI routing architecture originated with TCP/IP, developed into OSI standards, and returned to be adopted by the TCP/IP community—the TCP/IP routing architecture today is almost identical to the OSI architecture. The TCP/IP world began with a single network (which didn't require much in the way of routing), and grew into the "core"-based ARPANET, with individual networks connected to a single backbone (composed of "core gateways") as "stubs" (RFC 888; RFC 904). Multiple organizations began offering IP transit services, and for a time, it was difficult to tell whether the Internet was a "mesh" of backbone networks or a hierarchy. A three-tier hierarchy was gradually introduced as the NSFnet[5] grew and supplanted the ARPANET: the NSFnet served as a national backbone, and midlevel networks (or "regionals") provided transit services to and from the IP networks whose directly connected hosts served as the sources and sinks of Internet traffic.

Today, the TCP/IP routing architecture looks very much like the OSI routing architecture. Hosts use a discovery protocol to obtain the identification of gateways and other hosts attached to the same network (subnetwork). Gateways within *autonomous systems* (routing domains) operate an *interior gateway protocol* (intradomain IS-IS routing protocol), and between autonomous systems, they operate *exterior* or *border gateway protocols* (interdomain routing protocols). The details are different but the principles are the same.

tually suspicious network administrations, however, it may be the worst possible thing for the network to switch traffic to an alternate path "automatically" without first clearing the change with the legal departments of both parties. This conundrum has led one of the authors to claim that the only large-scale interdomain routing protocol that is likely to be deployed in the near future will be implemented as an army of lawyers on bicycles.

5. The National Science Foundation network (NSFnet) is the principal wide area backbone network for the Internet in the United States.

Routing Protocols

Over the past two decades, there have been more than a dozen routing protocols in operation in various parts of the Internet: HELLO, RIP (succeeded by RIP II), EGP, GGP, and BGP (which has gone through several versions, of which the most recent is BGP-4).[6] This is largely due to the fact that the Internet has been so successful that it has encountered—more than once!—a fundamental (architectural) limit to the ability of its prevailing routing technologies to cope with its increased size and extent. OSI benefited enormously from these years of experience with TCP/IP networks and can boast today that its routing protocols have been designed "from the beginning" to scale into the foreseeable future of billion-node internets. The boast is empty, of course, since the same lessons have been factored into a new generation of TCP/IP routing protocols; in some cases, in fact, the two communities are close to adopting the same routing protocol (much to the dismay of protocol-stack isolationists, but much to the relief of network operators and users!).

It is impossible to cover each of the TCP/IP routing protocols in detail in this chapter. However, it is instructive to compare a single TCP/IP routing protocol at each level of the routing hierarchy with an equivalent OSI protocol. The remainder of the chapter considers each of the three levels of the hierarchy (which is a common feature of both the OSI and the TCP/IP routing architectures) in turn, examining the OSI protocol (for which there is just one at each level) in some detail and comparing with it the currently recommended TCP/IP counterpart.

Reachability and Discovery in OSI—The ES-IS Protocol

In OSI, the discovery paradigm is *announcement*. An end system uses the *end system hello message* (ESH) of the ES-IS protocol (ISO/IEC 9542)[7] to announce its presence to intermediate systems (and end systems) connected to the same subnetwork. Any end or intermediate system that is listening for ES hello messages gets a copy; intermediate systems will store the NSAP address and the corresponding subnetwork address pair in routing tables. ESs may do so if they wish, or they may wait to be informed by intermediate systems when they need such information.

The announcement process is most effective when operated over a

6. RIP: routing information protocol; EGP: exterior gateway protocol; GGP: gateway-to-gateway protocol; BGP: border gateway protocol.

7. ISO often carries descriptive naming too far. The full title of the ES-IS standard is *End System to Intermediate System Routeing Exchange Protocol for Use in Conjunction With the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)*. No wonder everyone refers to it simply as "ES-IS"!

multiaccess, connectionless subnetwork such as an IEEE 802 local area network. ISO/IEC 9542 refers to such facilities as "broadcast subnetworks." ES-IS can, however, operate quite nicely over X.25 public (or private) data networks and over newer wide area network services such as switched multimegabit data service (SMDS; Perlman 1992b).

The protocol operates over an IEEE 802 local area network as follows. An end system periodically composes an ES hello—inserting its NSAP address(es), subnetwork point of attachment address, and a *holding time* into the ES hello packet—and sends this packet to the 48-bit multicast medium access control (MAC) address whose value has been defined to mean *all intermediate system network entities* (Figure 14.4). This aspect of the protocol is called the *report configuration* function.[8]

The frequency of these announcements is determined by the value of a locally administered *configuration timer* (CT). Why periodically? On LANs especially, computer systems are powered on and off frequently, sometimes for hours or more. LAN-based computers are also moved from office to office, and depending on how LANs are administered, NSAP addresses may remain associated with the computer or the office. Since resources for storing addressing information at intermediate systems are precious, it is important for ISs to know whether such information remains useful and accurate. Periodic transmission of ES hellos also

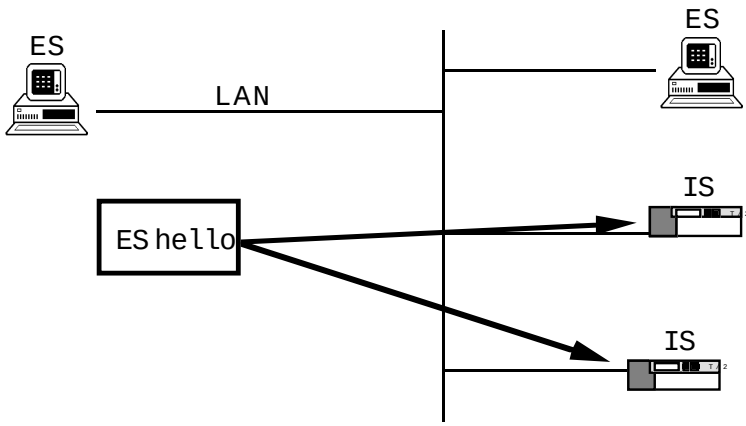

FIGURE 14.4 Report Configuration by End Systems

8. Over multidestination subnetworks such as SMDS, in which a form of multicast or group addressing is offered, configuration information can be announced to any set of individual subnetwork addresses that are collectively identified by a "group address" (see Chapter 15, "Emerging Digital Technologies").

allows a new router on a subnetwork to obtain configuration information about ESs on the subnetwork without explicitly asking for it and allows a router that has "crashed," lost state, and reinitialized to do so as well. The configuration time is used by the originators of hello messages to compute a *holding time* (HT). The holding time is encoded in the hello messages and tells recipients of hello messages how long the addressing information encoded in this hello message should be treated as accurate. Clearly, the shorter the configuration time (and accordingly, the holding time), the more accurate the addressing information, but the accuracy has to be weighed against the amount of traffic on the LAN and the resources consumed by recipients who must process each ES hello.

Upon receiving an ES hello, an intermediate system records the configuration information (NSAP addresses, subnetwork point of attachment address, holding time) contained in the message and holds the addressing information for the period of time indicated in holding time. If holding time elapses and no ES hello messages are received from this ES, the intermediate system discards the configuration information it has maintained for this ES, and assumes that the ES is no longer reachable. This aspect of the ES-IS protocol is called the *flush old configuration* function. Holding time is always greater than configuration time and is typically set at the source to at least twice the configuration time (with this value, even if every other ES hello is lost, ISs should hear a hello message, and thus, the presence of the ES on the LAN will be known and its configuration information will continue to be maintained by ISs).

An intermediate system performs nearly the same report configuration function (see Figure 14.5). An IS composes an *intermediate system hello*

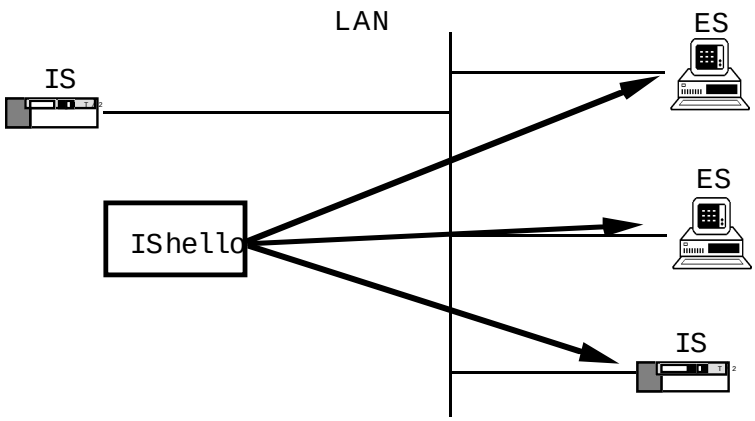

FIGURE 14.5 Report Configuration by Intermediate Systems

message (ISH) and announces its configuration information to intermediate systems and end systems alike connected to the same broadcast subnetwork. Like the ES hello, the IS hello message contains the addressing information for the IS (network entity title, subnetwork point of attachment address) and a holding time; optionally, IS hellos may encode a *suggested ES configuration time*, which recommends a configuration timer to end systems. This may be used by a LAN administrator to coordinate (distribute) the announcement process across all ESs that will report configuration to an IS. The announcement process over IEEE 802 LANs is again accomplished by issuing the IS hello message to a 48-bit multicast destination MAC address, but here, the value indicates *all OSI network entities* (so that both ESs and ISs may listen for the messages).

Upon receiving an ISH, an end system records the configuration information contained in the message and holds that information for the period of time identified in the holding time parameter. Again, if the holding time elapses and no IS hello messages are received from this IS, the end system discards the configuration information it has maintained for this IS and assumes that the IS is no longer reachable. If this was the only (or last remaining) IS the end system had discovered, it may no longer be able to communicate with end systems other than those attached to its own subnetwork. ISs also listen for ISH messages for initial configuration information prior to operating the IS-IS intradomain routing protocol.

Redirection Both IP and OSI have a redirection capability, and the two are functionally the same. In IP, redirection is part of the Internet control message protocol (ICMP, see Chapter 13, "Internet and OSI Control Messages"); in OSI, the redirect function, and the *redirect message* (RD), are part of the ES-IS protocol.

In OSI, redirection proceeds as follows. When an IS receives a CLNP data packet from an ES, it processes the packet and forwards it to the next hop toward its destination. If the IS determines that the destination is on the same LAN as the originator of the datagram, it returns a redirect message to the end system, indicating the *better next-hop subnetwork point of attachment address* (BNSPA) to that destination, which is the subnetwork point of attachment address of the destination NSAP address itself (Figure 14.6). Note that a holding time is also associated with redirect configuration information.[9]

Absent any further information in the redirect message, an end sys-

9. ISO/IEC 9542 Annex B describes important timer considerations and optimizations for redirection that are too detailed to describe here. These are discussed extensively in Perlman (1992b).

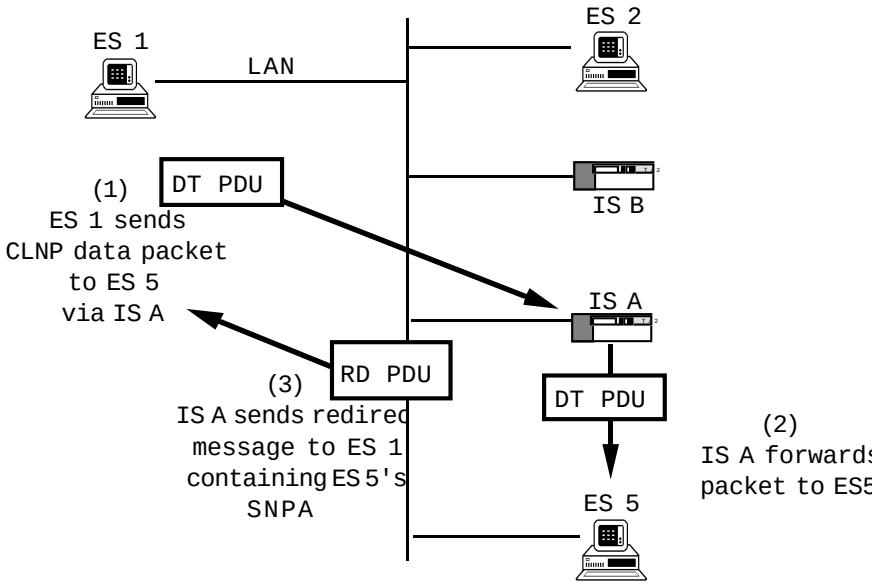

FIGURE 14.6 Redirection of One IS to Another IS

tem can only conclude that a better path to a given destination NSAP address exists through an IS identified by the network entity title and BSNPA encoded in the redirect message. For those situations in which the identified IS provides a better route to a larger set of destinations—or as ISO/IEC 9542 says, "an equivalence class of NSAP addresses to which the same redirection information applies"—an IS can include an *address mask* parameter, which indicates just how widely the redirection information can be applied. The address mask is composed from the destination NSAP address that caused the redirection, with binary 1s identifying how much of the NSAP address is the mask. For example, the bits representing the entire initial domain part and the first 16 bits of the domain-specific part of the mask might be set to binary 1s, and the remaining bits of the domain-specific part set to binary 0s. This signals to the end system that any time it receives a request to send a data packet to a destination that has the identical initial domain part and the same initial 16 bits of domain-specific part as this NSAP address, it should forward the packet to the IS identified in the redirect message.

One can't help but notice the similarity between the actions of ESs and ISs in this protocol and the actions of the citizens of Whoville in Dr. Seuss's Horton Hears a Who, *who, attempting to save their*

world, proclaim loudly and frequently, "We are here! We are here! We are HERE!" (Giesel 1971).

Reachability and Discovery in TCP/IP — ARP and Friends

The original *Address Resolution Protocol* (ARP; RFC 826) was designed to be used over a single Ethernet to solve a very simple routing problem: determine the 48-bit Ethernet hardware address associated with a specific Internet address. It has since been extended to operate over many local area network and metropolitan area network technologies and services, including SMDS (RFC 1209), frame relay (RFC 1293), and FDDI (RFC 1188).

The paradigm for the address resolution protocol is *request/reply*. To discover the binding between IP addresses and the interface addresses of other hosts, a host issues address resolution protocol requests over the Ethernet "broadcast" address (or its equivalent for non-Ethernet networks). All hosts listening for the broadcast address will receive a copy of the request; in its simplest form, the host whose IP address corresponds to that encoded in the request will return a reply to identify its hardware address. In another form, a router may listen for address resolution protocol requests for a set of IP addresses (i.e., addresses of hosts for which it provides routing service) and, acting as an agent or "proxy," will reply on their behalf ("proxy ARP").

In either case, ARP processing proceeds as follows. A TCP segment is submitted to IP for forwarding. The IP packet is created, and the routing process looks up the destination IP address in a routing table to determine the Internet address of the next hop (the destination address itself or the IP address of a router). Associated with this IP address is an address for the interface over which the packet is to be forwarded. This is called the *hardware address*—in this case, a 48-bit Ethernet address. When the routing process looks for the hardware address associated with a "next hop" and finds none, it builds an *ARP request* packet, placing the destination IP address from the IP packet into the request packet.[10] The request packet is "broadcast" over the local area network (step 1 in Figure 14.7). A copy of the packet is received by each station on the local area network that is listening for broadcasts. If the destination address in the ARP request packet doesn't match any of the local IP addresses, it is discarded. If a match is found, a reply packet is composed from the request; in particular, the source and destination IP address fields are reversed, the source hardware address is copied into the destination hardware address field, and the local hardware address is copied into the source hardware

10. In most instances, the routing process attempts to hold the IP packet until the ARP request is resolved; a system can drop the IP packet under circumstances enumerated in the Host and Router Requirements RFCs (RFC 1122, 1989), RFC 1123, 1989).

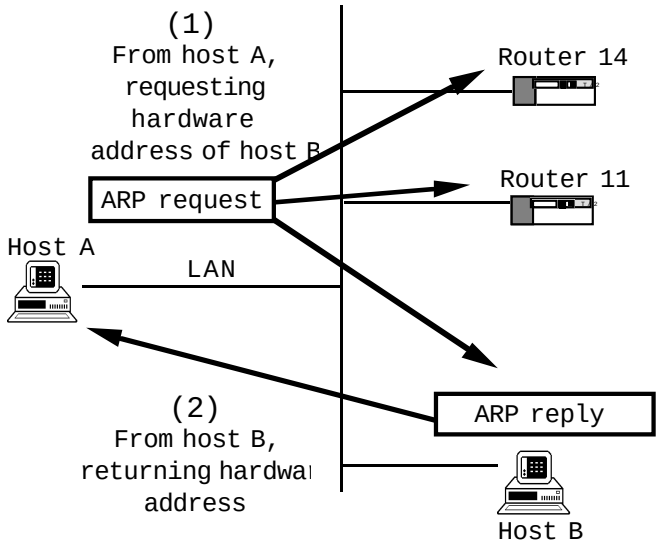

FIGURE 14.7 Address Resolution Protocol Request/Reply Sequence

address field of the reply packet. The ARP reply packet is then returned directly to the originator of the ARP request packet (step 2 in Figure 14.7). The requester now has the necessary information (IP address, hardware address pair) for routing to this destination.

As for the ES-IS protocol, a configuration or lifetime is associated with address information obtained via the address resolution protocol. Few details on how to implement this are provided in the address resolution protocol RFCs; Comer (1991) provides an example of how to implement the address resolution protocol and associated cache management.

Which Is Better? There are pros and cons to both discovery protocols. ES-IS was designed to be media-independent; all ESs and ISs use it, irrespective of the medium to which they are connected. The address resolution protocol has been extended to accommodate network interfaces other than the original Ethernet. The address resolution protocol is used "on demand," whereas ES-IS is operated periodically. The address resolution protocol uses a broadcast address (partly a consequence of its having been developed for the experimental Ethernet rather than the IEEE 802.3 local area network). This causes interrupts to occur at systems that are not IP-based, so all end systems must look at the packet. ES-IS uses multicast addresses, which restricts interrupts (and packet processing) to only those systems that are listening to OSI-specific 48-bit multicast addresses. In any event, both of these discovery protocols are a whole lot better than static tables.

Intradomain Routing in OSI

The OSI intradomain IS-IS routing protocol (ISO/IEC 10589) operates within a routing domain to provide every IS with complete knowledge of the topology of the routing domain. Generally speaking, within a single routing domain, small convergence time and simplicity of operation are more important than trust and fire walls. Thus, the main design goals of IS-IS are to

- *Accommodate large routing domains*: IS-IS is designed to handle 100,000 NSAP addresses and 10,000 ISs in a single routing domain.
- *Converge quickly*: When a change in topology in the routing domain occurs, IS-IS converges to correct routes quickly and without oscillation.
- *Provide operational "simplicity"*: In the OSI routing architecture, the role of ESs in routing is extremely simple, and with IS-IS, the configuration and parameter tuning of ISs is also relatively simple. IS-IS is relatively simple to maintain as well.
- *Support multiple subnetwork types*: IS-IS is designed to operate over local area networks, metropolitan area networks, point-to-point links, and X.25 networks. These subnetworks have widely differing delay, bandwidth, and operational characteristics (e.g., some offer connections, and some run datagrams), yet IS-IS is immune to the differences.

To accommodate large routing domains, IS-IS is organized as a two-level hierarchy. The subdomains that comprise level 1 of the hierarchy are called *areas*, and *level-1 ISs* within an area maintain routes only to destinations within their area. Routes to destination areas within the same routing domain (and routes to areas in other routing domains absent any policy restrictions—e.g., shortest paths to destination routing domains) are maintained at the second level of the hierarchy, by *level-2 ISs* (note that ISs designated as "level-2 ISs" maintain both level-1 and level-2 routes).

To converge quickly on the best (shortest) routes, IS-IS uses both a link-state routing algorithm based on the "new ARPANET" algorithm developed by McQuillan, Richter, and Rosen (1980), and the fault-tolerant broadcast mechanism developed by Radia Perlman (1983). IS-IS computes routes using the Dijkstra *shortest path first* (SPF) algorithm. Link-state routing avoids the "count-to-infinity," oscillatory behavior exhibited by many distance-vector algorithms such as the *routing information protocol* (RIP; RFC 1058, 1988). In a *distance-vector routing* environment, each router passes to its neighbors information about routes to destinations that the router has computed based on information it has received

from other neighbors; since the information being distributed is filtered through the route-computation process of each router, each router has only a partial map of the topology of a routing domain. In a *link-state routing* environment, each router broadcasts to every other router in the domain complete information about all of its links to adjacent routers; every router thereby acquires information describing the complete topology of the routing domain and uses that information to compute its own best routes to every destination. Paul Francis aptly described the difference between the two protocols by saying that "distance-vector routing is rumor-based, while link-state routing is propaganda-based."[11] Consider the topology of a single routing domain illustrated in Figure 14.8, in which all the links in the topology conveniently have equal costs, and the routing metric is hop count. Conceptually, distance-vector routing information received by router Dave from router Deborah would tell Dave, "I

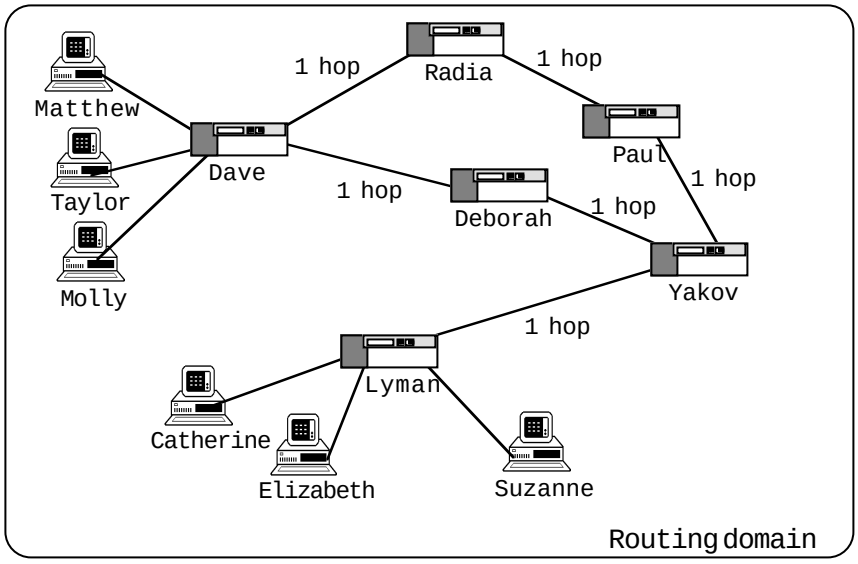

FIGURE 14.8 Example of a Routing Domain

11. To appreciate how link-state routing might be related to propaganda, readers must first refrain from assuming that propaganda is bad; in fact, propaganda is "the dissemination and the defense of beliefs, opinions, or actions deemed salutary to the program of a particular group." Routing information fits this definition nicely. Each router announces (disseminates information describing) its own neighbors, links, and link costs, effectively saying "it is so" (the "defense of beliefs" component); and indeed, all the routers that listen to the announcements believe them to be true. Each router is therefore a *propagandist*, devoting itself to "the spread of (a) system of principles or set of actions" (Webster's , 1977).

(Deborah) hear that I can get to Lyman, so you can get to Lyman through *me*, but it will cost you 2 hops"; the routing packet would be a bit more succinct, saying perhaps in this case that Deborah has a 2-hop route to Lyman. Dave would add the cost of the link he shares with Deborah (1 hop) and conclude that the cost of the path through Deborah to Lyman is 3 hops. Dave, hearing from Radia that she hears she can reach Lyman in 3 hops, would add the cost of the link he shares with Radia and conclude that the best path to Lyman is through Deborah (3 hops) rather than through Radia (4 hops).

Link-state routing information is distributed quite differently, and paints a *very* different conceptual picture of the topology. Deborah would announce ("advertise") a list of neighboring routers (Dave, Yakov) over each of her links; Yakov would announce his list (Deborah, Paul, Lyman), Radia her list (Dave, Paul), Paul his list (Radia, Yakov), and Lyman his list (Suzanne, Catherine, and Elizabeth—perhaps collectively referred to as "the Chapins", and Yakov). Dutifully, Dave would announce his list as well (Molly, Matthew, and Taylor—the "Piscitellos"—Deborah, and Radia).[12]

This information is flooded through the entire routing domain; everyone hears about everyone else's links, neighbors, and link costs. Dave waits until it's quiet, then deduces from the collection of announcements that "I can get to Deborah (1 hop); Deborah can get to Yakov (1 hop); Yakov can get to Lyman (1 hop); therefore *I* can get to Lyman over the link to Deborah in 3 hops!" (Having the entire set of announcements, Dave also deduces that he can get to Lyman over the link to Radia in 4 hops, and elects to use the path through Deborah; of course, since Radia was kind enough to offer Dave an *alternative* path, he keeps this in the back of his mind—just in case Deborah goes away, or becomes very busy).

The selection of a standard IS-IS protocol for OSI began in the United States in 1987 and is remembered as the "late, great, OSI routing debate." Two routing protocols were introduced as candidates—one from Digital Equipment Corporation and one from Burroughs (Unisys). Both protocols scaled well and offered hierarchical, multiple–quality of service routing with partition repair. The principal difference was that the DEC protocol was a link-state or "new ARPANET" protocol, and the Burroughs protocol

12. Distance-vector and link-state routing share the notion of *neighbor greetings exchange*, in which each router identifies itself to its neighbors when a link comes up or its status changes.

was a distance-vector or "old ARPANET" protocol.

The most damning fault of distance-vector routing is that it converges too slowly following topology changes. The Burroughs protocol, based on the Burroughs Integrated Adaptive Routing Algorithm System (BIAS) used in the Burroughs Network Architecture (BNA) (Piscitello and Gruchevsky 1987; Rosenberg, Piscitello, and Gruchevsky 1987), incorporated many improvements (Jaffe and Moss 1982; Kamoun and Kleinrock 1977; Tajibnapis 1977) to the original Bellman-Ford algorithm that dramatically reduced the seriousness of the problem, but it could not converge as quickly as the DEC link state protocol, and this difference proved to be the deciding factor.

A generic description of the IS-IS protocol follows. Initially, routers begin by learning about other routers to which they share direct connectivity; i.e., a router X learns first about those routers that are attached to the same subnetworks as X, often by using the IS hellos described earlier. This process is called neighbor greeting, or neighbor initialization. Each router then constructs a *link-state packet* (LSP), which contains a list of the names of its *neighbors* and its *cost*[13] to reach each of those neighbors. Routers then distribute these link-state packets to all the other routers. When all the link-state packets have been propagated to all the routers, each router will have received a complete map of the network topology in the form of link-state packets; each router uses these link-state packets to compute routes to every "destination" in the network using Dijkstra's *shortest path first* algorithm.[14]

All of the OSI NSAP address formats (see Chapter 13) can be used with the IS-IS routing protocol. For the purpose of computing routes, an NSAP address is always interpreted by IS-IS as having three fields: an area address, a system identifier, and an NSAP selector. The *area address* identifies an area within a routing domain; the *system identifier* unambiguously identifies a system or "host" within that area; and the *NSAP selector* identifies an entity within the system (a transport entity in an ES or a network entity in an IS; the NSAP selector is the OSI functional equivalent of the protocol identifier in IP). The relationship between the gen-

13. IS-IS uses non–traffic-based *metrics* to compute route costs. The metrics are insensitive to traffic, so IS-IS is able to avoid unnecessary recomputation of routes when traffic patterns on links oscillate. The default metric is capacity (a measure roughly equal to bits per second); however, IS-IS computes routes based on transit delay, cost (incremental expense, such as a charge per packet), and error (a measure of the probability of undetected errors on a circuit) so that CLNP datagrams originated with the QOS maintenance bits set (see Chapter 13) can be forwarded as requested by the originator.

14. Details of link-state packet distribution schemes and computation of Dijkstra's SPF algorithm as applied in IS-IS routing are described in Perlman (1992a).

eral structure of OSI NSAP addresses and the way in which NSAP addresses are interpreted by IS-IS for routing purposes is illustrated in Figure 14.9.[15]

The IS-IS interpretation of the domain-specific part (DSP) of the NSAP address, which has been standardized in the United States as American National Standard X3.216-1992, allows 1 octet for the NSAP selector, 6 octets for the system ("host") identifier,[16] and the remaining (leftmost) octets for the low-order part of the area address. The system identifier must be unique among all the NSAPs in the same area; it need not be globally unique, and it need not be an IEEE 802 medium access control address (it is, however, often convenient for a network administrator to use actual MAC addresses for the values of the system identifier as a simple way to ensure uniqueness). The IS-IS standard refers to the leftmost domain-specific part octets (the ones that form the low-order part of the area address) as the *LOC-AREA* field. The LOC-AREA value must be unique among NSAPs with the same initial domain part. In general, the procedures associated with level-1 routing look only at the contents of the system identifier field; the procedures associated with level-2 routing look only at the contents of the area address field.[17]

This address structure reflects the two-level hierachy of IS-IS routing, illustrated in Figure 14.10. In this figure, the notation "<number>.

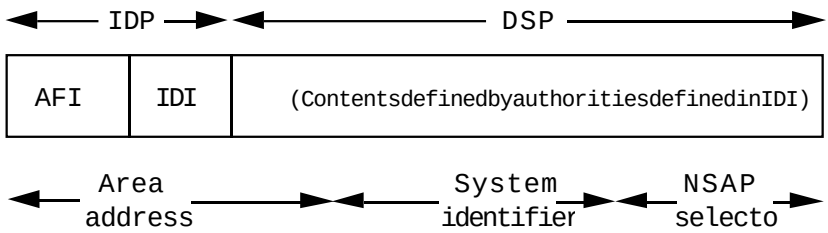

FIGURE 14.9 IS-IS Interpretation of NSAP Addresses

15. See Chapter 13 for a detailed description of the way in which the IS-IS interpretation of the NSAP address structure has been formalized as American National Standard X3.216-1992.

16. Readers who read "6 octets" and immediately thought, "I'll bet that the system identifier is expected to be a 48-bit IEEE 802 medium access control (MAC) address," are right—but they might also think about consulting a good obsessive/compulsive disorder specialist . . . The 6-octet size of the system identifier field was indeed chosen so as to make it possible to use a MAC address as a system identifier field value, but none of the ISO/IEC standards requires that this be the case.

17. An intermediate system that is configured to operate as a level-2 IS may also be configured to participate in level-1 routing in areas to which it is directly connected; that is, a single hardware component (what one would ordinarily recognize as a "router box") may participate in both level-1 and level-2 routing.

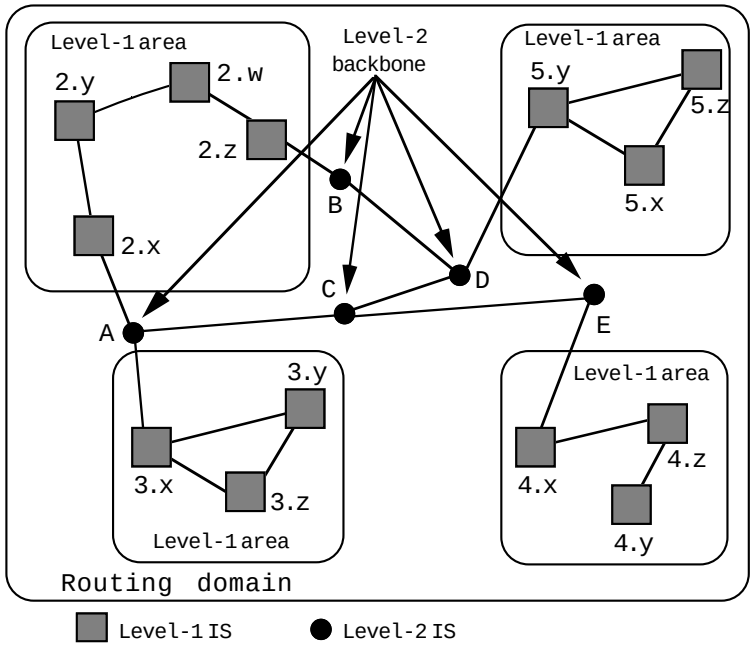

FIGURE 14.10 Two-Level Hierarchy of IS-IS Routing

<lowercase letter>" is used for the area address and system identifier parts of the network entity titles of level-1 ISs; the notation "<uppercase letter>" is used for the area address part of the network entity titles of level-2 ISs.

A level-1 IS needs to know only about the ESs and other level-1 ISs in its own level-1 area and about the "nearest" level-2 IS which it can use to forward traffic out of its own area. The level-1 "view" of the routing domain depicted in Figure 14.10 is illustrated in Figure 14.11.

A level-2 IS needs to know only about other level-2 ISs in its own routing domain, the location of level-1 areas, and the "best" exit level-2 IS to use for traffic destined for other routing domains. The level-2 "view" of the routing domain depicted in Figure 14.10 is illustrated in Figure 14.12.

The existence of and distance to other routing domains is administratively configured into level-2 ISs that share links with ISs in other routing domains. The other routing domains are included in level-2 link-state updates and propagated to all other level-2 ISs. When an interdomain routing protocol is present (see "Interdomain Routing in OSI," later in the chapter), it may be possible for border level-2 ISs to learn about other routing domains dynamically.

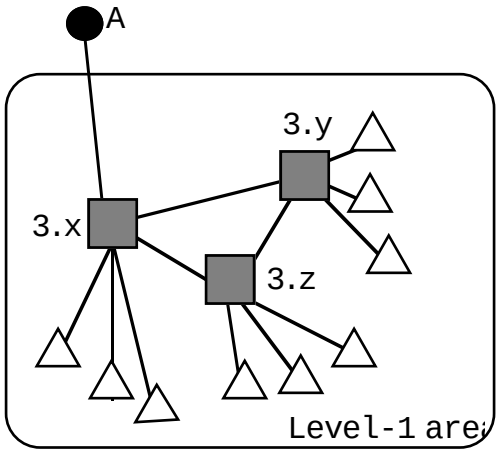

FIGURE 14.11 Level-1 View of a Routing Domain

Once NSAP addresses and routing metrics are configured, IS-IS runs automatically: "best" routes are automatically calculated and recalculated upon topology change without manual intervention.

The IS-IS Link-State Algorithm: A Closer Look The idea behind link-state routing is that each IS obtains a full topology map of the network—a complete description of how the ISs and ESs are connected. Using this map, each IS can generate a next-hop forwarding table by calculating the shortest path to all ESs. As long as all ISs have identical topology maps

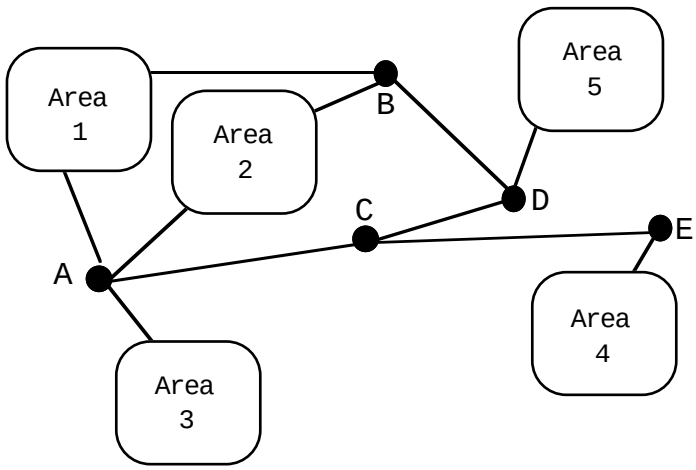

FIGURE 14.12 Level-2 View of a Routing Domain

and use the same algorithm to generate the next-hop forwarding table, routes will be computed correctly.

The topology maps constructed by each IS are distributed among and collected by ISs using a directed form of flooding. Each IS generates a link-state update consisting of the list of ISs (in the case of a level-2 IS) or of ESs and ISs (in the case of a level-1 IS) to which it is connected, along with the metric associated with each link to an ES or IS in the list. The link-state update—consisting of a set of link-state packets (LSPs)—is flooded to all neighbor ISs, which flood it to their neighbor ISs, and so on (with the proviso that a link-state packet is *not* propagated back to the neighbor from which it was received). ISs designated as level-1 ISs propagate link-state packets to other level-1 ISs in their area; level-2 ISs propagate level-2 link-state packets to other level-2 ISs throughout the routing domain. Sequence numbers are used to terminate the flood and to distinguish old link-state updates (duplicates) from new ones. Every IS receives link-state updates from every other IS, and from these, it builds a full topology database. When the connectivity of an IS changes, it floods another link-state update.

The Two-Level Hierarchy Level-2 ISs form a backbone that connects different level-1 areas. The IS-IS protocol assumes that the level-2 ISs are *connected*; that is, that any level-2 IS can reach any other level-2 router by going through only level-2 routers. Note that this does not imply that "mesh" connectivity must exist as depicted in Figure 14.10; in fact, any of the three links between the level-2 ISs in Figure 14.10 could be removed and the notion of "connected" would be preserved. As depicted, the three links simply provide resiliency of the level-2 "backbone."

Although routing hierarchies can be built without this sort of backbone configuration, it simplifies the operation of both level-2 and level-1 ISs. The connection of level-2 ISs frees routers designated as "level-1 ISs only" from having to know anything more than how to route to the nearest level-2 IS (see Figure 14.10). Once a packet reaches a level-2 IS, it can reach its destination level-1 area (or another routing domain entirely) via level-2 IS routing exclusively. If this were not true, then level-1 ISs would need to know the location of all other level-1 areas to prevent loops as the packet was sent from level-2 ISs to level-1 ISs and back to level-2 ISs. Another advantage of the connected level-2 "backbone" is that it simplifies routing protocol evolution. Since level-1 ISs need not know the level-2 routing protocol or topology, changes can occur at level 2, or in other level-1 areas, without affecting the operation of other level-1 ISs.

Two penalties are incurred as a consequence of the "backbone"

simplification. The first is the restriction on the topologies that can be constructed (since all the level-2 ISs must be directly connected without using level-1 paths). Considering that many routing domains have or require a backbone topology anyway, and that any level-1 IS can be administratively upgraded to a level-2 IS if necessary to connect the backbone, this restriction is at most a minor inconvenience.

The second penalty is a potential increase in path length. To illustrate, consider again the topology depicted in Figure 14.10, and imagine that all the links have the same metric value. Assume that a message from an ES attached to level-1 IS {2.w} is destined for an ES attached to {3.x}, outside its local area. Level-1 IS {2.w} must forward the packet to its nearest level-2 IS; based on the topology information that {2.w} keeps, the packet will be forwarded to level-2 IS B via {2.z}. Level-2 IS B will forward it over the level-2 backbone to IS A, which will deliver it to level-1 IS {3.x}. The path chosen is six hops long—two hops more than the shortest path via level-1 IS {2.y}.

Area Partitions One of the problems associated with an area hierarchy is the possibility of an area *partition*. For instance, if the link between level-1 ISs {2.y} and {2.w} in Figure 14.10 is lost (see Figure 14.13), then level-1 IS {2.y} will not be able to deliver packets to {2.z} even though a physical path exists (via level-2 ISs). Why? As a level-1 IS, {2.y} keeps routing information only about the area in which it resides and about the nearest level-2 IS. Knowing from previous link-state updates that {2.z} is in its area because of the area address part of its address and that {2.w} provided a route to {2.z}, and knowing also that its link to {2.w} is lost,

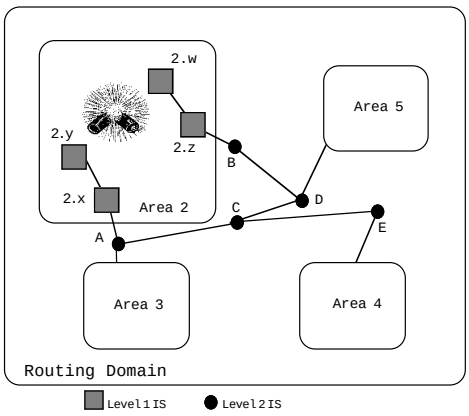

FIGURE 14.13 Area Partition

{2.y} would not be able to route packets within the area to {2.z} because of the partition. In fact, some packets forwarded to {2.z} coming from the backbone would be delivered to {2.y} because they had entered the "wrong" partition segment of the area. This is called a level-1 partition.

Level-2 partitions are also possible. For instance, if the link between level-2 ISs C and D breaks, packets coming from area 3 will not get to area 5 even though a physical path exists (through area 2; see Figure 14.14). IS-IS dynamically repairs level-1 partitions but not level-2 partitions. The partition-repair mechanism works entirely in level-2 ISs. Level-2 ISs discover the partition because they get inconsistent information from level-1 ISs, telling which level-2 ISs are attached to an area, and from level-2 ISs, telling which level-1 areas can be reached (recall that all level-2 ISs that are attached to an area are also level-1 ISs in that area, giving them access to the level-1 IS information required to know which ESs are in their partition segment). When a partition is discovered by level-2 ISs, all of the level-2 ISs "elect" a *partition designated level-2 IS* in each partition segment to execute a repair (the winner of the election is the "partition-repair-capable" level-2 IS with the numerically lowest system identifier). Partition designated level-2 ISs build a level-1 repair path between them by establishing a virtual level-1 link between them using level-2 ISs and passing level-1 routing updates over the level-1 virtual link. CLNP datagrams, error reports, and link-state updates travel across the virtual link as encapsulated CLNP datagrams.[18] The level-1 ISs do not know that the virtual level-1 link passes through level-2 ISs—they see it

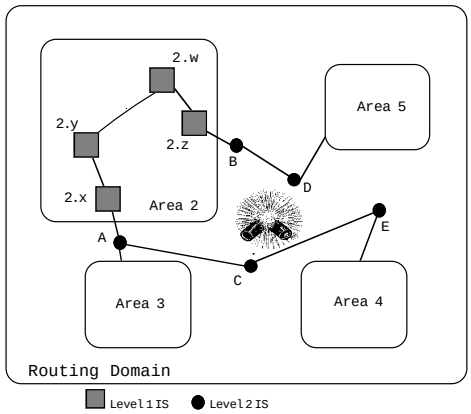

Figure 14.14 Level 2 Partition

18. Every packet that must traverse the level-1 repair path is encapsulated within another CLNP packet with its own header. The source and destination address information in the

as no different from a "real" level-1 link.

For example, consider again the case in which the level-1 link in Figure 14.10 between {2.y} and {2.w} is lost. Level-2 ISs A and B will discover the partition and, following the election, will be recognized as partition designated level-2 ISs. A and B will establish a virtual link between themselves and pass level-1 IS updates between themselves encapsulated as CLNP datagrams. Now assume that an ES attached to level-1 IS {4.x} has a packet destined for an ES attached to level-1 IS {2.z}. The packet will initially be routed to level-2 IS A, since it is the shortest path to that area (again assuming that all links shown have identical metric values). Level-2 IS A will recognize that it cannot deliver the packet via level-1 links and so will encapsulate the packet in a CLNP header addressed to level-2 IS B. The packet will backtrack to level-2 IS B, which will decapsulate the packet and deliver the original CLNP packet from {4.x} to level-1 IS {2.z}. The level-1 repair path for this example is illustrated in Figure 14.15.

No IS other than level-2 ISs A and B is aware that a partition exists and was repaired. The penalties for the advantage of localizing the repair effort are suboptimal paths and the extra burden of encapsulation and decapsulation. The gain, of course is, the preservation of connectivity.

Routing Metrics In the examples used so far, all of the metric values have been assumed to be the same. Typically, metric values are assigned to help control the flow of traffic over individual links. The default metric for IS-IS is a 1-octet parameter that represents a measure of capacity/throughput of a link. The "length" of a path is the sum of the metric

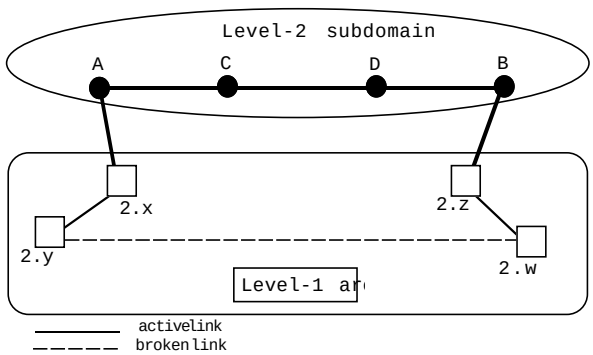

FIGURE 14.15 Level-1 Repair Path

additional header is populated with the network entity titles of the level-2 routers providing the partition repair.

values for the links on that path. One octet was chosen because it gives adequate granularity and allows for an efficient implementation of the shortest-path-first route calculation—i.e., it allows ISs to generate the forwarding table from the topology map efficiently and quickly.[19]

IS-IS optionally computes routes based on metrics for transit delay, cost, and error rate, so that CLNP datagrams originated with the quality of service maintenance bits set can be forwarded as requested by the originator over appropriate routes. ISs that compute routes based on optional metrics report these metric values for links in link-state packets (in which a separate octet is identified for each optional metric). ISs compute quality of service routes by using only link-state packets that contain values for a given metric; for example, to compute a "cost" route, an IS will run the shortest-path-first algorithm using only those link-state updates that indicate a value for the cost metric and only if the IS itself has a link with a cost metric associated with it.

Partial Routing Updates Since the purpose of a routing protocol is to assist in directing traffic, and not to consume resources that might otherwise be employed to forward user data, it is always desirable to keep routing updates small while still keeping track of which neighbors are reachable and which are not. IS-IS routing updates originally conveyed which neighbors were reachable by listing them and which neighbors were not reachable by not listing them; if a neighbor was listed in a previous routing update but was not in the latest routing update, that neighbor was assumed to be unreachable. As a result, all existing neighbors were listed in all routing updates.

The more neighbors a router had, the larger its routing updates. Large updates had to be fragmented and reassembled by CLNP, which slowed down the entire update process (since the fragments had to be reassembled at each hop during flooding). IS-IS now uses *partial updates* to circumvent this problem.

With partial updates, the otherwise complete routing update is split into multiple updates, and each update indicates the numerical range of neighbors that are covered by the update. This way, the semantics of explicitly listing reachable neighbors and implicitly listing unreachable neighbors is preserved. If a partial routing update does not list a neighbor that is in the range covered by the update, then it is assumed that the neighbor is unreachable. The various partial updates each maintain their

19. An implementation of the shortest path first algorithm used by IS-IS is likely to assume that an IS maintains an intermediate table of entries, ordered by distance; by limiting distance values to 1 octet, this intermediate table is bounded to a reasonable size.

own sequence numbers. If the status of a neighbor changes, only the partial update covering that neighbor must be sent.

Pseudonodes IS-IS defines another method of limiting the size of routing updates for those configurations in which many ISs populate a broadcast subnetwork (such as a local area network). On these subnetworks, and using the general case principles, each IS is a neighbor of all the other ISs attached to the same subnetwork. If each of the N ISs on a broadcast subnetwork were to send full routing updates, then N ISs would each send a routing update describing $N - 1$ neighbors, burdening the subnetwork with the exchange of order $N^{(N-1)}$ routing information. To prevent this, one IS on the subnetwork is elected as the *pseudonode* or LAN desigated router (see Perlman, 1992a). The pseudonode advertises all the other ISs (including itself) as its neighbors, but all the other ISs advertise *only* the pseudonode as their neighbor. With the pseudonode, a logical star configuration with N links is formed rather than the original fully connected "mesh" configuration with $N^{(N-1)}$ links. To maintain the correct metric values, the metric value from the pseudonode to the (real) ISs is 0.

Operation over X.25 Subnetworks When operating over an ISO/IEC 8208 (X.25) subnetwork, IS-IS distinguishes virtual circuits established by administrative procedure (static) from virtual circuits established because of the receipt of a CLNP packet (dynamically allocated, as described in the CLNP standard). Among the dynamically allocated circuits, IS-IS also distinguishes those for which the NSAP addresses reachable via the neighbor are learned from routing updates from those for which the NSAP addresses reachable via the neighbor are locally administered. The latter case allows the use of X.25 subnetworks without requiring that the virtual circuits be left open solely for the purpose of exchanging routing updates. Depending on the tariffs associated with a particular X.25 network service, call duration may be an expensive proposition, and this optimization leaves such links open only when they are needed.

"Integrated IS-IS" The OSI standard for IS-IS (ISO/IEC 10589: 1992) deals only with the provision of routing information to support the forwarding of CLNP packets. Internet RFC 1195 augments the base standard by describing how IS-IS can be used to route both CLNP *and* IP datagrams. RFC 1195 specifies how ISs distribute information about the TCP/IP destinations they can reach in addition to the OSI destinations they can reach, simply by adding more information to the IS-IS link-state packets. The mechanism extends naturally to environments that include other protocol stacks as well; for example, the techniques described in

RFC 1195 could be used to build routes to XNS/IPX[20] destinations by adding IPX-specific information to link-state packets.

The term used to describe the operation of a single intradomain routing protocol to support the forwarding of traffic associated with multiple protocol stacks is *integrated routing*. The alternative to integrated routing, of course, is to run a separate intradomain routing protocol for each protocol stack (OSI, TCP/IP, XNS/IPX, etc.). Advocates of integrated IS-IS cite the advantages of fewer protocols to operate, fewer resources devoted to the routing process, and simpler management. Advocates of the alternative (which is called "ships in the night," or "S.I.N.," by members of the Internet community) cite the advantages of intradomain protocols that are custom-tailored to their corresponding protocol stacks, noninterference of one protocol stack (and its routing concerns) with others, and the ability to configure ISs with only those intradomain routing functions that are needed for the stack(s) that will be supported.

Intradomain Routing in TCP/IP

A variety of different intradomain routing protocols are operated in the Internet today. The most commonly used is the *routing information protocol* (RIP; RFC 1058). RIP is the routing protocol "we love to hate"; its main problem is that today's Internet is far too big and diverse for it to handle. The routing information protocol operates as an application over the user datagram protocol (UDP), computes routes using a distance-vector algorithm, and uses a hop-count metric that reflects distance but not link speed, efficiency, or congestion. The routing information protocol is vulnerable (as are all distance-vector routing protocols) to loops; to avoid them, it uses a variety of more or less effective techniques, including a maximum hop count of 16 to indicate "no path" or "link down." This limits the network diameter to a maximum of 15 hops, which is small by today's standards.[21]

HELLO is another open routing protocol that has seen better days; but seriously, how can you hate something called "HELLO"? It's simply easier to rip RIP.

The new kid on the block is an intradomain routing protocol called the *Open Shortest Path First Protocol* (OSPF; RFC 1247). Open shortest path

20. XNS: Xerox Network Systems; IPX: Internet packet exchange.
21. A version 2 of RIP that deals with these and other limitations has been published as a proposed standard (RFC 1388).

first is a close relative of the OSI intradomain IS-IS routing protocol, tailored specifically for the TCP/IP (only) environment.

How is OSPF like IS-IS? Both OSPF and IS-IS are link-state routing protocols that compute routes using Dijkstra's shortest path first algorithm and distribute link-state information (OSPF calls this information "link-state advertisements," or LSAs) using Perlman's fault-tolerant broadcast technique. OSPF has a two-level hierarchy: a backbone and attached areas. It is capable of providing multiple types of service routing as indicated in the IP header. It handles area partitions and provides pseudo-node (designated router) optimization over local area networks.

How does OSPF differ from IS-IS? Mostly, in the way in which some of the detailed operations of the protocol are performed. OSPF is encapsulated in IP datagrams; IS-IS operates directly over the individual underlying data link (or subnetwork) protocols. The two protocols also differ in how they deal with link-state updates that are very large and may require fragmentation. IS-IS puts all the link-state update information into a single link-state packet with a single header, and ISs fragment the link-state packet if it is too large, using a single fragment number to identify and order the fragments of the link-state packet. OSPF builds separate link-state advertisements for each destination (i.e., multiple packets) and combines these into a single IP datagram. The OSPF encoding is optimized for a scenario in which incremental updates may be frequent, and hence, the savings in link utilization will be great. The tradoff is an increased consumption of memory to accommodate the overhead of many separate link-state advertisements rather than one link-state packet.

OSPF and IS-IS also differ in their philosophies of "route granularity." OSPF propagates link-state advertisements between areas, so that a level-1 IS can choose which level-2 IS offers the best path to destinations outside its own area. In the OSPF scheme, the advantage of having more refined routes is traded off against the disadvantage of increased usage of memory and link resources. When used for routing CLNP packets, IS-IS has no way of telling ISs about addresses in other areas, and hence, it is assumed that IS-IS imposes a strict separation of area information. When integrated IS-IS is used, however, IP addresses look the same, regardless of whether they are from inside or outside the area, so the same granularities can be achieved using either integrated IS-IS or OSPF.

OSPF and IS-IS also handle level-1 (area) partitions differently. Level-1 partitions are repaired automatically by IS-IS; they are not repaired at all in OSPF, except by manual reconfiguration of OSPF's level-2 address summaries after a partition has occurred. IS-IS requires that

level-2 ISs be connected only through other level-2 ISs, and therefore, partition repair always involves encapsulation (tunneling). OSPF provides network administrators with the ability to manually configure routes or "virtual links" to circumvent level-2 partitions.

It is possible to debate the pros and cons of IS-IS and OSPF much more passionately than the authors have done here (see, for example, Coltun [1989]; Tsuchiya [1989]; Perlman [1991a, 1991b]; Medin [1991]). The fact that IS-IS can be used in both OSI-only and dual-stack roles in parts of the Internet may be important, perhaps increasingly so as CLNP deployment increases and as the Internet community deals with scaling issues such as IP address exhaustion (see Chapter 16). However, a recently published Internet applicability statement[22] (RFC 1370) specifies the use of OSPF for intradomain routing in TCP/IP-only domains of the Internet, leaving the future of "integrated IS-IS" in doubt. There is nevertheless at least one thing on which the proponents of OSPF and the proponents of IS-IS strongly agree—*both* of these protocols are better than their predecessors!

Interdomain Routing in OSI

The OSI *interdomain routing protocol* (IDRP;[23] ISO/IEC 10747) views the global OSI internetwork as an arbitrary interconnection of routing domains (RDs) connected to each other by subnetworks and by "border" intermediate systems (BISs), which are located in routing domains and attached to these subnetworks. Each border IS resides in a single routing domain and may simultaneously participate in both the interdomain routing protocol and an intradomain routing protocol of the domain (e.g., IS-IS).

IDRP calculates interdomain routes as a sequence of *path segments*. A path segment consists of a pair of border ISs and a link that connects them. If a pair of border ISs are attached to a common subnetwork, then the link between them is called a *real link*. Links between border ISs in different routing domains are always real. Within a single routing domain, however, a link that connects two border ISs may be constructed and maintained by intradomain routing protocol procedures; such links

22. *Applicability statements* are a type of Internet standard, distinct from technical specifications in that they describe the way in which a protocol or mechanism shall be used in a particular configuration or context, not the details of the protocol or mechanism itself; see Chapter 2.

23. The curse of "acronymania" has fallen with particularly cruel force on the field of interdomain routing. Two very different interdomain routing protocols—the OSI interdomain routing protocol (IDRP) and the Internet's interdomain policy routing (IDPR) protocol—are commonly referred to by their almost identical acronyms, neither of which is readily "pronounceable"!

are called *virtual links*.

To illustrate the role of IDRP, consider how a packet leaves an end system in one domain and travels to an end system in some other domain. First, the packet is forwarded by the ES to the nearest IS in its "home" domain. The packet will be forwarded by this and possibly other ISs via intradomain routing to a border IS in the home domain. The border IS uses information obtained via IDRP to determine a path to a border IS in an adjacent domain lying on a route to the destination and forwards the packet across the path segment toward the first-"hop" border IS in the interdomain path, which lies in the next domain. When the packet arrives at the next domain, the border IS in that domain forwards it through that domain toward a border IS located in the next hop or domain along the interdomain route. (Note that the path segment within this or any domain transited may be constructed either by IDRP alone or by a combination of IDRP and the intradomain routing procedures.) The process will continue on a hop-by-hop basis (with hops being expressed in terms of domains) until the packet arrives at a border IS in the domain that contains the destination end system. Here, the packet will be forwarded to its destination via the intradomain routing procedures of the destination routing domain.

As an illustration, consider the interdomain topology shown in Figure 14.16, and assume that the path between ESs in routing domains A and F goes through routing domains B, C, and E. A packet originated by

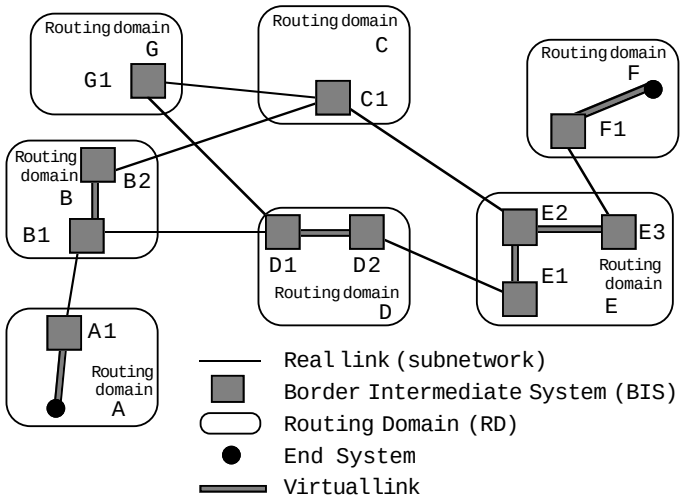

FIGURE 14.16 Example of Interdomain Topology.

an ES located in routing domain A and destined for an ES located in routing domain F will first be routed by means of routing domain A's intradomain routing procedures to border IS A1. Border IS A1 uses information obtained via IDRP to determine that the next border IS on the path is B1, which is located in the next domain along the route. Since border ISs A1 and B1 share a common subnetwork, border IS A1 can forward the packet directly to border IS B1 over a real link. Border IS B1 uses information obtained via IDRP to select border IS B2 as the next border IS along the path. (Note that the link between B1 and B2 is a virtual link; hence, the forwarding of the packet along this link is accomplished using domain B's intradomain routing procedures.) The process continues through border ISs C1, E2, and E3, until the packet finally reaches border IS F1, which is located in the routing domain (F) that contains the destination end system. The packet is forwarded to the destination ES within routing domain F by means of domain F's intradomain routing procedures.

The IDRP Routing Algorithm The routing algorithm employed by IDRP bears a certain resemblance to distance-vector (Bellman-Ford) routing. Each border IS advertises to its neighboring border ISs the destinations that it can reach. The IDRP routing algorithm augments the advertisement of reachable destinations with *path attributes*, information propagated in IDRP routing control packets that describes various properties of the paths to these destinations. To emphasize the tight coupling between the reachable destinations and the properties of the paths to these destinations, IDRP defines a route as a unit of information that consists of a *pairing* between a destination and the attributes of the path to that destination. A path, expressed in terms of the domains that the routing information has traversed so far, is carried in the *RD_PATH* attribute of the IDRP packets that distribute the routing information. Therefore, one may call IDRP a *path-vector* protocol, in which a border IS receives from each neighboring border IS a vector that contains paths to a set of destinations.[24]

The IDRP routing algorithm is distributed: each border IS maintains partial routing information, and routes are determined through the exchange of information among border ISs and through local computations carried out in each border IS. The IDRP routing algorithm assumes that each routing domain has a globally unique *routing domain identifier* (RDI), which is simply a network entity title,[25] and that the NSAP addresses

24. The term *path-vector protocol* bears an intentional similarity to the term *distance-vector protocol*, in which a border IS receives from each of its neighbors a *vector* that contains *distances* to a set of destinations.

25. A routing domain identifier always identifies a particular domain for the purposes

assigned to ESs within the routing domains that participate in IDRP exchanges are unique and unambiguous as well. Although this is a general characteristic of NSAP addressing, and is also assumed by the OSI intradomain routing protocol, it is mentioned explicitly here because IDRP has no mechanism to enforce either of these two assumptions or even to detect whether they have been violated (which may result in incorrect protocol operation).

Loop-free Routing in IDRP IDRP is designed to ensure that, during steady-state operation (i.e., no interborder IS topological changes have occurred), routes computed by the border ISs participating in IDRP are loop-free. Since data packets flow in a direction opposite to the direction in which IDRP routing information flows, one way to satisfy this design objective is to ensure that routing information does not loop. Because the RD_PATH attribute contains a list of routing domains expressed in terms of the routing domain identifiers of the domains that routing information has traversed so far, suppressing routing information looping is straightforward: a border IS that receives a route must examine the RD_PATH attribute of the route and check whether the routing domain identifier of the border IS's own routing domain is present in the attribute. If it is, the border IS must not use this route.

As an illustration of how the loop-suppression mechanism works, consider a flow of routing information (route) that advertises reachable destinations within routing domain F (see Figure 14.17). Assume that routes are selected in such a way that routing domain C prefers to reach destinations in routing domain F via routing domain E, routing domain B prefers to reach destinations in routing domain F via routing domain C (then via routing domain E), and routing domain D prefers to reach destinations in routing domain F via routing domain B (then via routing domains C and E). Denote the routing domain identifiers of the participating domains by B, X, Δ, E, and Φ. Border IS F1 originates the route and propagates it to border IS E3. At this point, the RD_PATH attribute associated with the route contains only one routing domain identifier—namely, the routing domain identifier of F (RD_PATH is < Φ >). Border IS E3 propagates this information to the other border ISs within its own domain—namely, E1 and E2. As E2 propagates the route to border IS C1, it updates the RD_PATH attribute by appending to it the routing domain identifier of its own routing domain (RD_PATH becomes < Φ, E >). Likewise, when C1 propagates the route to border IS B2, it appends the routing domain

of IDRP but does not necessarily convey any information about the NSAP addresses of the end systems within the domain.

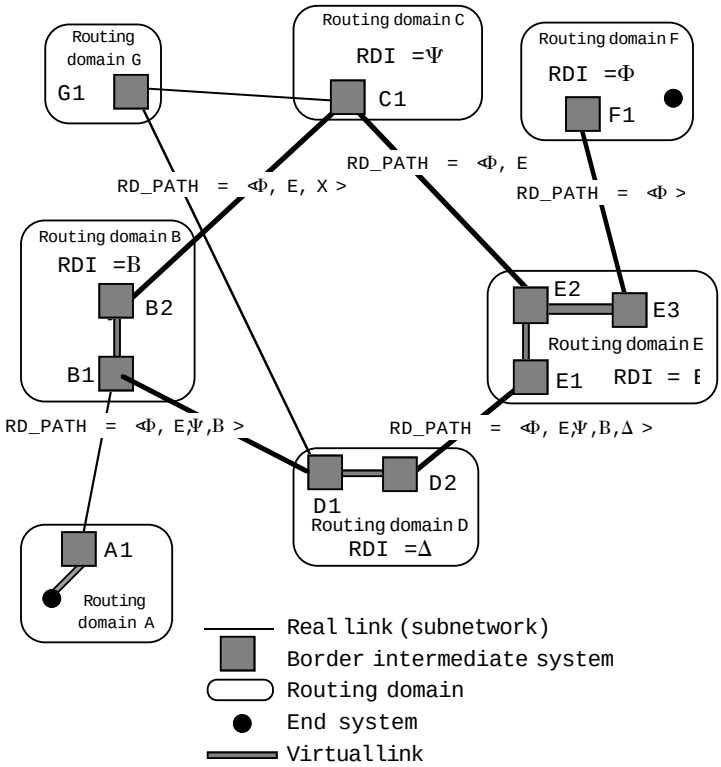

FIGURE 14.17 Interdomain Topology with Routing Domain Identifiers

identifier of its own routing domain, C, to the RD_PATH attribute (RD_PATH becomes < Φ, E, X >). Since border ISs B2 and B1 reside within the same routing domain, propagation of the route from B2 to B1 does not result in any changes to the RD_PATH attribute. Border IS B1 propagates the route to border IS D1 and appends routing domain B's own routing domain identifier to the RD_PATH attribute (RD_PATH becomes < Φ, E, Ψ, B >). Border IS D1 propagates the route to border IS D2 (without any modification to the RD_PATH attribute). However, when border IS D2 propagates the route to border IS E1, E1 detects that it cannot use it, since the RD_PATH attribute of the route already contains routing domain E's own routing domain identifier, E (RD_PATH of the route that border IS D2 propagates to border IS E1 is now < Φ, E, C, B, Δ >).

Even when connectivity between border ISs F1 and E3 is disrupted, thus making the original route unusable, none of the border ISs within domain E can switch to a route that was advertised by border IS D2

(since such a route would involve routing information looping). (Note that we have traced only one flow of routing information in this example. Readers are encouraged to trace other flows to acquire a feel for how the routing process converges.)

Selecting an Optimal Route IDRP does not require all routing domains to have identical criteria (policies) for route selection. IDRP does not even have a notion of a globally agreed upon metric that all the domains must use for route selection. Moreover, the route-selection policies used by one routing domain are not necessarily known to any other routing domain. To maintain the maximal degree of autonomy and independence among routing domains, each routing domain that participates in IDRP may have its own view of what constitutes an optimal route. This view is based solely on local domain route-selection policies and the information carried in the path attributes of a route.

Note that when presented with exactly the same set of possible routes to a given destination, different routing domains may choose different routes from the set. As an example, consider again the interdomain topology shown in Figures 14.16 and 14.17 and assume that routing domains C and D both have a route to destinations in routing domain F (both of which go through domain E). When routing domains C and D advertise this information to routing domains B and G, routing domain B may select the route offered by C, whereas routing domain G may select the route offered by D, despite the fact that C advertises exactly the same route to both B and G.

The following essential properties of the IDRP algorithm make it possible to provide loop-free steady-state routing even in the presence of independent route-selection policies that need not be known outside a domain:

- Routing information distribution is preceded by the computation over the information.
- Only the results of the computation are distributed to other border ISs.
- Suppression of routing information looping does not depend on the presence of a globally agreed upon, monotonically increasing metric.

Thus, the only thing that is needed to provide consistent operation across multiple routing domains is to standardize the mechanism used to exchange the results of the route-selection procedure (the results of the route computation) while allowing the policies that affect route selection to be of local significance only (nonstandard).[26]

26. This elegant and succinct observation was made by Ross Callon.

Restricting Transit Traffic Providing ubiquitous connectivity within the global OSI internetwork implies that a domain may need to share its own transmission and switching facilities (links, subnetworks, and ISs) with other domains, as well as rely on the willingness of other domains to share theirs. Of course, once sharing exists, the ability to control it becomes an important issue; there is always a cost associated with deploying and managing transmission and switching facilities, and the administrator responsible for these facilities must be able to control and account for their use, particularly by parties outside the scope of his or her own local user base.

One such control mechanism is the ability to restrict transit traffic, or prohibit it entirely, for a particular set of domains or for all other domains. In such cases, a domain is said to employ *transit restrictions* (*transit policies*) as a way of controlling the use of its facilities.[27]

Transit Policies Supported by IDRP The fundamental technique used by IDRP in support of transit policies is controlled distribution of routing information. This technique is based on a simple observation: when a packet's flow is determined by the routing information present in border ISs, the direction of this flow is opposite to the direction of the routing information flow between the border ISs. Controlling routing information flows allows restriction of packet flows, which, in effect, imposes transit policies.[28] (In the extreme case in which there are no routes over which a packet may be forwarded, the packet is discarded.)

Selective Advertisement of Routing Information IDRP provides several mechanisms with various degrees of granularity and sophistication that allow a routing domain to control the distribution of routing information, thus providing control over its own transit traffic. The simplest mechanism is to have a border IS within a domain announce only a subset (or possibly none) of the routes that it uses to any border ISs in adjacent domains. Note that a border IS may announce a different subset to border ISs in different adjacent domains, so as to grant to some routing domains a specific set of transit privileges and to others none.

To illustrate how this mechanism can be applied in the context of the topology illustrated in Figure 14.16, consider the following example. If border IS B1 has a route to destinations in routing domain F (which

27. Examples of domains that rely heavily on transit policies are the backbone networks sponsored by agencies of the U.S. federal government, such as the Energy Sciences Network (ESnet), the NASA Science Internet (NSI), and the National Science Foundation Network (NSFnet).

28. This technique was pioneered in the NSFnet backbone phase II (RFC 1104; RFC 1092).

may traverse, for example, routing domains C and E), but does not announce this route to border IS A1, then no packets originated from an ES in routing domain A and destined for an ES in routing domain F will traverse the facilities that are under the control of routing domain B. This imposes a transit policy that prevents transit traffic from routing domain A to routing domain F from flowing through routing domain B. The same border IS B1 may, however, announce a route to destinations in routing domain F to border IS D1, thus allowing transit traffic flowing between routing domains D and F to traverse routing domain B. Furthermore, by making the distribution of routing information depend on the contents of the RD_PATH attribute, this mechanism may be extended to take into account not only the origin and the destination of traffic but the intervening path (in terms of routing domains) as well.

This mechanism works well if one wants to restrict transit traffic between sources that are within routing domains immediately adjacent to one's own routing domain and destinations that are in some other (and not necessarily adjacent) routing domain. However, to impose a more general transit policy, IDRP provides another mechanism, one that explicitly controls the distribution of routing information, thereby enabling a domain to restrict the scope of potential recipients of the routing information that flows through it. The scope may be specified by using a *distribution inclusion list* (DIST_LIST_INCL) path attribute, which allows the enumeration of the potential recipients of the routing information (identified by their routing domain identifiers), or a *distribution exclusion list* (DIST_LIST_EXCL) path attribute, which allows the enumeration of the domains that shall be excluded from receiving the routing information. The mechanism provided by the DIST_LIST_INCL and DIST_LIST_EXCL path attributes imposes no restrictions on the relative placement of the source and destination domains of the transit traffic that must be restricted.

Referring again to Figure 14.16, suppose that routing domain C is willing to carry transit traffic destined for any ES in routing domain F, as long as the traffic does not originate from any of the ESs in routing domain A. To impose such a transit policy, border IS C1 must attach a DIST_LIST_EXCL path attribute that contains A's routing domain identifier to the route that it advertises to border IS B2 (DIST_LIST_EXCL of the advertised route shall be < A >). Border IS B2 will propagate this route, together with the DIST_LIST_EXCL attribute, to border IS B1. Although B1 may now propagate this route to border IS D1, it may not send it to border IS A1, since A1 belongs to the routing domain A, which is listed in the DIST_LIST_EXCL attribute of the route.

The ability to control the distribution of routing information provides an organization with corresponding control over the way in which its resources may be used, while facilitating the interconnection of multiple organizations—thereby satisfying a major objective of OSI routing. It remains to be seen whether the controls that are available in IDRP are sufficient or whether the deployment of a more elaborate control system, such as that provided by the source-routed interdomain policy routing (IDPR) protocol, will be necessary.

Routing Domain Confederations To achieve good scaling characteristics, IDRP provides a mechanism that allows a set of connected routing domains to be grouped together and treated as a unit. Such a grouping of domains, called a *routing domain confederation* (RDC), is defined and established by means that are beyond the scope of IDRP.[29] From the outside, a routing domain confederation (hereafter, simply "confederation") looks like a single routing domain: routing domains outside the confederation cannot tell whether the confederation is just a single domain or a collection of domains. This mechanism is inherently recursive, so that a member of a confederation may be either a routing domain or a set of routing domains that themselves form a confederation. Confederations may be nested, disjoint, or overlapped. The protocol does not place any topological restrictions on the way in which confederation boundaries are defined, nor on the number of confederations to which a single domain may belong.

Routing domain confederations are identified by *routing domain confederation identifiers* that are taken from the same address space as the routing domain identifiers used to identify individual routing domains. The protocol assumes that all the border ISs that belong to a confederation are statically preconfigured with both the routing domain identifier of their routing domain and the routing domain confederation identifier of the confederation. If a routing domain belongs to more than one confederation, then the static information must include information about the nested/disjoint/overlap relationships that exist between each pair of confederations. No other requirements are imposed on the confederation members. For example, different confederation members may advertise different routes to a given destination—there is no need to have consistent route-selection policies for all the members of a confederation. Different members may also impose different transit policies—there is no

29. Specifically, IDRP does not suggest or recommend any principles, operating procedures, or policies that might be used by administrative domains which attempt to establish bilateral or multi-organizational routing relationships.

need for an *a priori* agreement on the transit policies that will be imposed by individual members.

Note that *how* one establishes a routing confederation is outside the scope of the protocol; however, the protocol explicitly recognizes a confederation's boundaries by taking specific actions when routing information enters and exits a confederation. Specifically, routing information that enters a confederation is marked by appending a special RD_PATH path segment (either ENTRY_SET or ENTRY_SEQUENCE), followed by the routing domain confederation identifier of the entered confederation, to the RD_PATH attribute.[30] As routing information exits a confederation, the RD_PATH attribute is scanned in the reverse direction: the protocol searches for the ENTRY_SET or ENTRY_SEQUENCE path segment that contains the routing domain confederation identifier of the confederation that is about to be exited. Exiting a confederation results in removing all the domains within the confederation that the routing information traversed, leaving only the routing domain confederation identifier, thereby reducing the amount of information carried in the RD_PATH attribute.

The ability to group routing domains into confederations provides a powerful mechanism for routing information aggregation and abstraction, which, in turn, makes IDRP scalable to a practically unlimited number of routing domains. The formation of confederations can keep the RD_PATH information from growing beyond a manageable size (and from preserving topological details that may be irrelevant in a very large internetwork). Similarly, transit policies may be expressed in terms of confederations rather than individual domains.

Using Confederations to Support Transit Policies Relations between confederations—expressed in terms of nesting, disjoint, or overlap—may be used to impose certain types of transit policies that otherwise would be difficult to support. Such policies are imposed via a "two-phase" rule that states that once routing information enters a confederation, it cannot exit the confederation. To record the fact that routing information has entered a confederation, IDRP uses the HIERARCHICAL_RECORDING path attribute. The distribution of the routing information is determined by the value of the attribute and by the flow of the information. Specifically, if a

30. Recall that the RD_PATH attribute is defined as a sequence of path segments. IDRP supports four distinct types of path segments: RD_SEQ (an ordered set of domains or confederations), RD_SET (an unordered set of domains or confederations), ENTRY_SEQ (an ordered set of entered but not exited confederations), and ENTRY_SET (an unordered set of entered but not exited confederations).

given border IS chooses to advertise routing information whose HIERARCHICAL_RECORDING attribute is equal to 1, it may advertise it to a border IS located in any adjacent routing domain as follows:

- If it is necessary to enter a confederation in order to reach the adjacent border IS, then the advertising border IS shall set the value of the HIERARCHICAL_RECORDING to 0.
- If the adjacent border IS can be reached without entering any confederation, then the advertising border IS shall not change the value of the HIERARCHICAL_RECORDING.

If a border IS chooses to advertise routing information whose HIERARCHICAL_RECORDING attribute has a value of 0, it may advertise this information only to border ISs that can be reached without exiting any confederation to which the advertising border IS belongs.

To illustrate how this mechanism can be applied, consider the topology shown in Figure 14.18, in which solid dots represent individual routing domains and rectangular boxes represent the boundaries of routing domain confederations. Routing domains E and F form confederation X; routing domains G and H form confederation Y; and routing domains A through H form confederation Z. Routing domain A's transit policy is such that it is willing to carry transit interdomain traffic that originates in L, D, C, B, or A and is destined for K, provided that the traffic does not traverse any of the domains in either confederation X or Y. Routing domain A is also willing to carry transit interdomain traffic that originates in any domain within confederations X or Y and is destined for K. Such a transit policy may be supported by requiring border ISs in A that are connected to border ISs in B and E to attach the HIERARCHICAL_RECORDING path attribute with the value of 1 to the routing in-

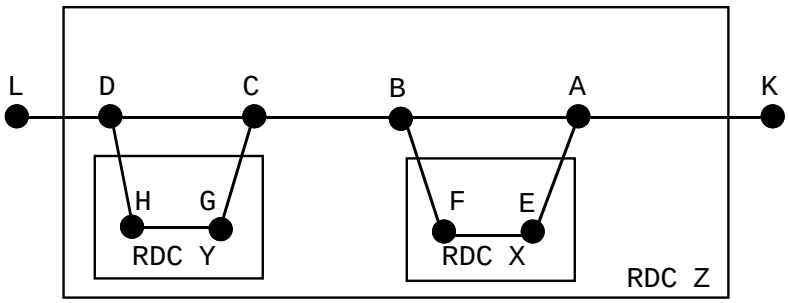

FIGURE 14.18 Example of the use of the HIERARCHICAL_RECORDING Attribute

formation that they receive from border ISs in K prior to propagating this information to border ISs in routing domains B or E. As a result, routing domain F would not be allowed to propagate routing information to routing domain B that traversed routing domains A and E. Likewise, routing domain H would not be allowed to propagate routing information to routing domain D that traversed routing domains A, B, C, and G. That leaves routing domain B with routing domain A as the only possible route to K, and leaves routing domain C as the only possible choice that goes via B and A, and routing domain D as the only possible choice that goes via C, B, and A.

Using the HIERARCHICAL_RECORDING path attribute in conjunction with the formation of routing domain confederations as a way of implementing transit policies creates an environment that significantly enhances an administrator's ability to assess the effect of changes in interdomain connectivity on interdomain routing. For example:

- Changes in the connectivity between domains that are located in a region in which two or more routing domain confederations overlap will not affect routing between domains outside the overlap. This is because border ISs located in routing domains within the overlap cannot propagate routes between domains located outside the overlap.
- Changes in the connectivity between routing domains within a confederation cannot affect routing between domains outside the confederation, since routing domains that form a confederation cannot be used for transit traffic between routing domains outside the confederation.
- Changes in the connectivity between a border IS that belongs to a domain within a confederation and a border IS that belongs to a domain outside the confederation have no effect on routing between the domain outside the confederation and any other domain outside the confederation.

Handling Reachability Information Recall that in the context of IDRP, a route is defined as a pairing between a set of reachable destinations and the attributes of the path to those destinations. The set of reachable destinations is called *network layer reachability information* (NLRI).

IDRP expresses network layer reachability information in terms of NSAP address prefixes. Such an encoding provides a flexible and concise mechanism for describing a set of reachable destinations. A set covered by an NSAP address prefix may be as small as a single end system or as large as a collection of all end systems within a confederation consisting

of multiple routing domains. If NSAP addresses are assigned in a hierarchical fashion, and the address assignment hierarchy is congruent with interdomain topology, the ability to express network layer reachability information in terms of NSAP address prefixes provides a powerful tool for reducing the amount of reachability information that must be carried and processed by the protocol. This reduction, accomplished by IDRP's route-aggregation procedures, allows IDRP to combine several routes into a single route. Part of the route-aggregation procedure involves the aggregation of network layer reachability information, in which a set of longer address prefixes is aggregated into a single shorter address prefix.

Supporting Multiple Routes to a Destination IDRP allows a border IS to support multiple routes to a destination, based on the ability of CLNP to carry quality of service and security parameters (see Chapter 13). A pair of border ISs may exchange not one, but multiple "path vectors" to the same destination, with each vector being tagged by its distinguishing path attributes. This tagging allows a border IS to construct multiple forwarding tables, one for each set of distinguishing attributes associated with individual path vectors. When a border IS needs to forward a packet, it checks for the presence of quality of service or security parameters in the packet and, if any are present, maps them into a corresponding set of distinguishing attributes. To determine the next hop on a path to a destination, the border IS uses the forwarding table tagged with the appropriate distinguishing attributes. To ensure consistent forwarding across multiple domains, IDRP standardizes on the mapping between quality of sevice and security parameters supported in CLNP and the distinguishing attributes that are defined in IDRP.

Conceptually, one may think of a set of distinguishing attributes defined in IDRP as a set of "route colors." Similarly, one may think of the quality of service and security parameters defined in CLNP as a set of "packet colors." Tagging path vectors with appropriate distinguishing attributes may be viewed as "route coloring"; specifying quality of service or security parameters in CLNP packets may be viewed as "packet coloring." Consistent mapping between a route's colors and a packet's colors across domains ensures correct forwarding.

Like the quality of service and security parameters in CLNP—which permit the use of source-specific, destination-specific, and global parameter values—route and packet colors also have a scope. Specifically, the meaning of a color may be defined only within the scope of a source or destination ES, or a color may have global significance. The scope of the source- and destination-specific quality of service and security *distin-*

guishing attributes in IDRP with respect to their significance for forwarding is determined by the source addresses and destination addresses, respectively. The scope of such distinguishing attributes as cost, residual error rate, and capacity is global; that is, their significance for forwarding does not depend on the source or destination addresses in the CLNP packets.

IDRP requires all the border ISs within a single domain to agree on the supported distinguishing attributes. However, the protocol does not require all the domains within the global OSI environment to support the same set of distinguishing attributes. In other words, the decision to support or not support a particular set of distinguishing attributes is left to a domain's local administration.

Having the ability to support multiple routes to a destination provides IDRP with a flexible and powerful scheme for supporting various forwarding granularities, including source- and destination-sensitive forwarding.

Support for multiple routes to a destination allows IDRP to take into account factors such as the cost or quality of service associated with different routes; because "someone is paying," support for such features is critical to interdomain routing.

Routing Information Exchange IDRP packets (IDRP PDUs) are carried in the data field of CLNP packets. A border IS that participates in IDRP exchanges routing information (routes) with its neighboring border ISs in the form of *IDRP update packets*.

The set of neighbors is partitioned into two categories: internal and external. The internal neighbors are the border ISs that belong to the same routing domain as the local border IS. The external neighbors are the border ISs that belong to different routing domains but share a common subnetwork with the local border IS.

IDRP requires all border ISs within a domain to maintain pairwise connections with each other, in effect creating a complete mesh of border-IS to border-IS connections. Although this appears to be a significant burden, careful analysis shows that such a mesh has no impact on overhead, other than maintaining state information for each of the border ISs within a domain. This overhead will usually be negligible compared to the amount of interdomain routing information that a border IS needs to handle.

Routing information exchange in IDRP may be modeled as a two-phase procedure. A border IS first selects the best route among the routes

received from all its external neighbors and advertises this route to all its internal neighbors. In the second phase, a border IS selects the best route from all the routes received from all its neighbors, both external and internal, and advertises this route to all its external neighbors (subject, of course, to constraints on dissemination of routing information). To achieve consistency between border ISs within a domain, a border IS that sends a route to an internal neighbor attaches the ROUTE_SEPARATOR path attribute to the route. This attribute carries the metric that the border IS assigns to that route. The receiving border IS may use this information to check whether its own route selection policies are consistent with the one of the sending border IS.

The IDRP routing information dissemination algorithm that controls the exchange of routing information (update packets) between border ISs is based on the technique of *incremental* updates, in which after an initial exchange of complete routing information, a pair of border ISs exchange only changes to that information. This technique helps to conserve a significant amount of resources (processor cycles and bandwidth), which is critical when dealing with an environment consisting of a large number of domains. Support for incremental updates requires reliable information exchange between participating border ISs. IDRP supports this reliability by a combination of sequence numbers assigned to update packets, explicit acknowledgments, and retransmission of unacknowledged update packets. In addition to incremental updates, IDRP provides a mechanism to perform a complete update of the routing information in either solicited or unsolicited mode (this is done by using refresh packets).

The IDRP routing information dissemination procedure is completely event-driven and dynamic. The two possible events that may trigger the dissemination of (new) routing information are the establishment of a new session with a neighboring border IS and the selection of new routes by the local border IS as a result of either receiving an update packet from one of its neighbors or the deletion of all the routes received from a neighbor due to loss of the neighbor.

IDRP in Large Public and Private Data Networks For domains that employ general mesh topology subnetwork technology, such as X.25 or Switched Multimegabit Data Service networks (see Chapter 15), IDRP avoids the use of border ISs within such domains for packet forwarding altogether; rather, IDRP uses border ISs solely as a mechanism to put transit and route-selection policies in place. To accomplish this, a border IS is allowed to pass a route to another border IS located in an adjacent

domain that contains the NSAP of the next hop on a path to a destination that is different from the NSAP of the border IS that sends the update packets (this information is conveyed in an update packet and encoded in the NEXT_HOP path attribute). In such circumstances, the border IS acts as a *route server*, whose sole responsibility is to participate in the protocol without forwarding any packets.

Using this technique, a routing domain that forms a large public or private data network may need to deploy only a handful of border ISs that would maintain border-IS to border-IS connections with a large number of external border ISs located in domains that are attached to that network. Packet forwarding between these domains through the network will be accomplished solely by the border ISs in these domains without any direct participation of border ISs within the network. Moreover, in order to receive routing information from all the domains attached to such a large data network, an external border IS needs to maintain border-IS to border-IS connection only with the border ISs that act as route servers rather than with all other external border ISs.

Figure 14.19 illustrates IDRP operation in a large data network (straight lines denote border-IS to border-IS connections). Routing domain X, which forms the network, has only two border ISs deployed within its boundaries. All the border ISs of the domains that use routing domain X as a transit have border-IS to border-IS connections with either of these two border ISs. Some of them may be connected to only one border IS in routing domain X; some, for redundancy, may be connected to

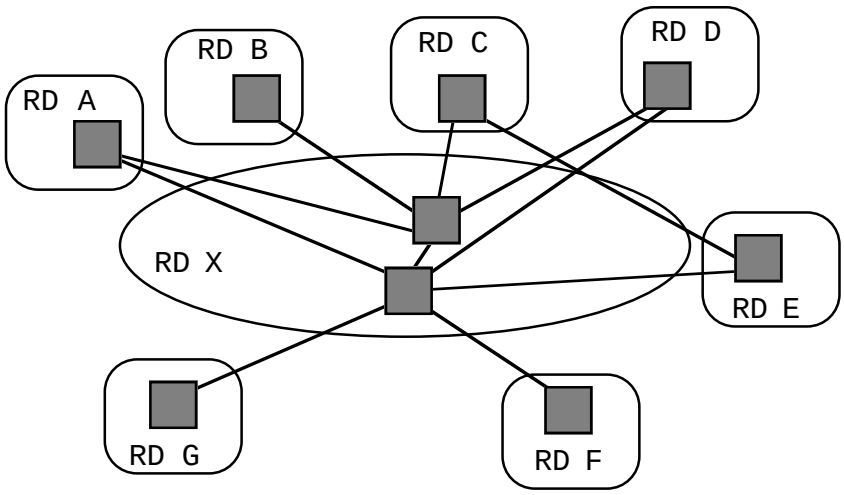

FIGURE 14.19 Example of IDRP in a Large Data Network

both. If a border IS in routing domain A advertises to a border IS in routing domain X a route to a destination, then the border IS in routing domain X may readvertise this route to border ISs in other domains, such as border ISs in routing domains C or G, with the NSAP of the border IS in routing domain A, rather than its own, as the next hop. Thus, when the border IS in routing domain G needs to forward a packet to that destination, it may send it directly to the border IS in routing domain A. To facilitate forwarding, the information about the next-hop NSAP may be further augmented with the list of SNPAs associated with that NSAP. Observe that although border ISs in the domains attached to routing domain X have sufficient routing information to send packets directly to each other through routing domain X, distribution of this information does not require these border ISs to have border-IS to border-IS connections with each other. Instead, the distribution is accomplished as a result of maintaining border-IS to border-IS connections with only border ISs in routing domain X.

Interdomain Routing in TCP/IP

In the terminology of TCP/IP routing, "interdomain" routing becomes "interautonomous system" routing. The first *exterior gateway protocol* (RFC 904) assumed a tree structure for the Internet: lots of midlevel networks branching off the core or backbone autonomous system, and lots of networks branching off the midlevel networks. In principle, leaf networks forwarded IP packets to midlevels if they didn't know how to get to a destination, midlevels forwarded IP packets to the core if they didn't know how to get to a destination, and the omniscient core dealt with forwarding IP packets out through the branches to the leaves.

All well and good, right? Well, not exactly. First, the deployed Internet topology did not conform to the model: links were deployed and used to route packets between the leaf networks and between midlevels, and (horrors!) more backbones were introduced! (So much for the notion of a "core.") Second, and equally disconcerting, the EGP model required that the core know everything about the connectivity of the topology (a major shortcoming of EGP was the fact that it functioned as both an intra- and interdomain routing protocol). EGP always distributed complete updates: as the Internet grew, the size of EGP "updates," which are sent in single IP packets, became very large, and IP reassembly became a bottleneck.[31] EGP really didn't address the issue of policy; "policies" were manually introduced by operators and applied as *filters* to EGP

31. Recall that both OSPF and integrated IS-IS try to avoid IP fragmentation and reassembly by allowing incremental (partial) updates.

routing updates as they were received. Finally, and most damaging of all, EGP couldn't detect loops (it insisted that the topology have none). In short, EGP simply could not cope, nor could the network administrators using it.

Enter the *border gateway protocol* (first specified in RFC 1105; subsequent versions, denoted BGP-1 through BGP-3, have appeared in RFC 1163 and RFC 1267). Incrementally, versions of BGP have freed and will free the Internet from the restrictive notion of a tree-based topology for inter-AS routing. BGP versions 3 and 4 (an RFC in preparation) offer a subset of the features for IP that IDRP offers for CLNP (not particularly surprising, since it is the ancestor of the ISO IDRP [Rekhter 1991; Tsuchiya 1991]). Like IDRP, BGP uses a path/distance-vector method of route computation and distribution: at start-up, BGP *speakers* (the equivalent of border ISs in IDRP) exchange their complete routing information bases and subsequently distribute incremental updates of in-use autonomous system paths only. Like IDRP's RD_PATH, which is composed of routing domain identifiers of real and virtual paths between border ISs, an *autonomous system path*, composed of Internet network numbers, may be real or virtual; however, the terms used in BGP are *external* or *internal paths*. As BGP speakers distribute reachability information, including path attributes, they compose and forward a complete list of autonomous systems that have already forwarded this routing information; like IDRP's compilation of the RD_PATH, this is done to avoid looping. Whereas IDRP operates point-to-point between border ISs over CLNP and supports reliability by a combination of sequence numbers, explicit acknowledgments, and retransmission of unacknowledged update packets in IDRP proper, BGP speakers operate pairwise over TCP connections and leave reliability to TCP.

Final Comments on Policy Routing

Interdomain routing protocols like BGP and IDRP are fine for most forms of traffic enforcement, especially those that can be generally applied to a majority of users to facilitate routing domain interconnection; in such cases, the routing overhead—interdomain distribution and route computation—is more than justified, since these policies are widely used. Both BGP-4 and IDRP are expected to scale well enough to accommodate general needs for the foreseeable future.

There are other types of interdomain routes, however, that merit special consideration. For example, a research branch of one company may collaborate with another, perhaps even a competitor, to advance a particular technology. In cases such as these, it may be important to offer connectivity between the research laboratories of the two companies, but

it remains equally important to both administrations that they retain the ability to prohibit access to other parts of their enterprise networks. Another example is one in which a company maintains networks in several countries; quite often, international law frowns on such companies' providing intertransit services to third parties. In small doses, policies such as these can be managed by IDRP or BGP, but it is quite obvious that dealing with every policy eccentricity of every interdomain pairing of communicating hosts with hop-by-hop routing will quickly tend toward unmanageability; IDRP and BGP don't scale **that** well! In other words, there is a need for "exception-case" interdomain routing.

The *interdomain policy routing* (IDPR) architecture and protocol (RFCs in preparation) provide the means whereby special restrictions on interdomain routes may be specified or "demanded" at the source; hence, the name *source-demand routing*. The current version of IDPR uses link-state routing to distribute interdomain reachability and policy information between *policy gateways*, which, like IDRP's border ISs and BGP's speakers, are routers that are directly connected across administrative domains (here, the relationship is called a *virtual gateway*). The policy gateways in each domain receive interdomain link-state advertisements and calculate routes based on the topology map they have received according to whatever policies the administration wishes to enforce. When a packet is to be forwarded with certain policy considerations, it is forwarded to a policy gateway within the routing domain of that packet's source. This "source" policy gateway calls upon a route server to assist in setting up a route determined by the source policy gateway; a *route-setup* packet that describes the interdomain path selected by the source policy gateway is forwarded to all the policy gateways that must maintain the route in the transit (and destination) routing domains. A path or flow identifier is assigned to this interdomain route, and all packets that are to be forwarded across this route are encapsulated in an IP datagram that carries the flow identifier as additional header information.

"Unified" Interdomain Routing Note that these two different methods of interdomain routing overlap, in the sense that nearly the same types of controls can be imposed on interdomain traffic using either method (RFC 1322, 1992). However, it is not necessary to choose between the two methods of interdomain routing; rather, each has its respective merits. RFC 1322, A *Unified Approach to Inter-domain Routing*, suggests that in a robust interdomain routing architecture, these two methods are complementary. "Hop-by-hop" or *node-routing* interdomain protocols are important because they allow administrations to set policies for the largest percent-

age of interdomain traffic and from the largest number of sources; i.e., policies for the masses. Of course, since the node-routing protocols are expected to support the masses, they must maintain routes that are operational and under some administrative control, and they must also be able to adapt to topology changes, to keep the interdomain traffic flowing in the face of failures (IDRP and BGP have these characteristics).

Source-demand routing protocols like IDPR are important because they allow administrations to build special routes that could be supported by the "hop-by-hop" protocols, but at considerable routing expense/overhead. Having both methods available allows administrations to put in place policies that are generally needed while accommodating exceptional circumstances. Balancing between the two is an important operational matter. Since the routes maintained by source-demand routing protocols are supposedly "special," there is an assumption that they will be used sparingly; indeed, if they become widely used, then the routing information should be distributed by node-routing protocols.

Conclusion

This chapter has discussed the routing principles, architecture, and protocols of OSI and TCP/IP—in the process, illustrating how OSI routing has benefited from what has been learned from the considerable experimentation and field experience with IP routing. The IP community has developed a hierarchy of routing functions and protocol in an evolutionary fashion, by necessity (a consequence of its success) increasing the robustness and reach of its protocols and architecture to scale to millions of hosts. The resulting IP routing architecture locates discovery and reachability among hosts and gateways in one functional tier, providing best paths between hosts within a single routing domain in a second tier, and providing methods for routing between domains in a third tier. The OSI community readily adopted this functional hierarchy, defining the three functional tiers of ES-IS, intradomain IS-IS, and interdomain IS-IS routing. The benefit to the OSI community is that it won't have to undergo many of the growing pains that have been endured by the IP community and that it can support a routing environment that scales at least as well as that of IP. The OSI community—and of course, the "tweeners"—have attempted to return the favor by defining some useful improvements in OSI routing protocols that can be applied in IP-only or integrated routing protocols as well.