CHAPTER 2

Routing Basics

Background

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

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

Routing Components

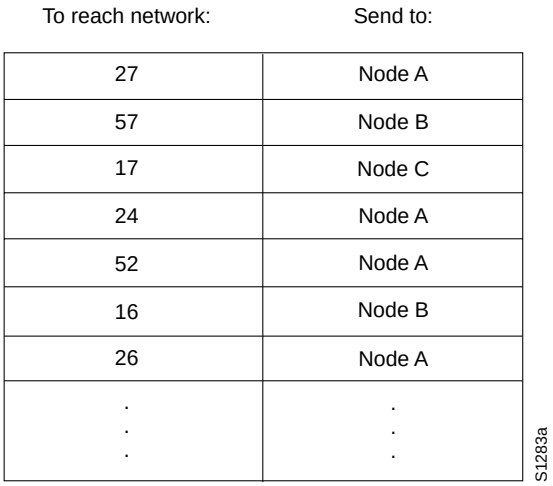
Routing involves two basic activities: determination of optimal routing paths and the transport of information groups (typically called *packets*) through an internetwork. In this publication, the latter of these is referred to as *switching*. Switching is relatively straightforward. Path determination,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

Path Determination

A *metric* is a standard of measurement—for example, path length—that is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain *routing tables*, which contain route information. Route information varies depending on the routing algorithm used.

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

Figure 2-1 Destination/Next Hop Routing Table



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

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

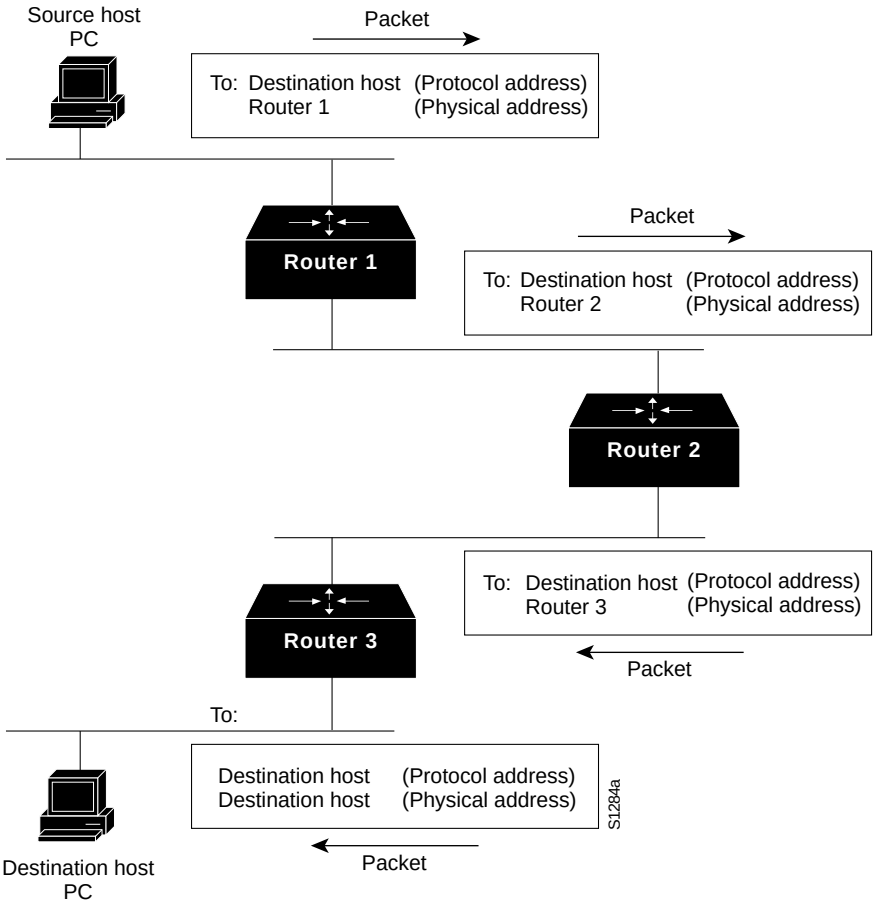
Switching

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

On examining the packet's destination protocol address, the router determines that it either knows or does not know how to forward the packet to the next hop. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, it changes the destination physical address to that of the next hop and transmits the packet.

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

Figure 2-2 Switching Process



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

Routing Algorithms

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

Design Goals

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

- Optimality
- Simplicity and low overhead
- Robustness and stability
- Rapid convergence
- Flexibility

Optimality

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

Simplicity

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

Robustness

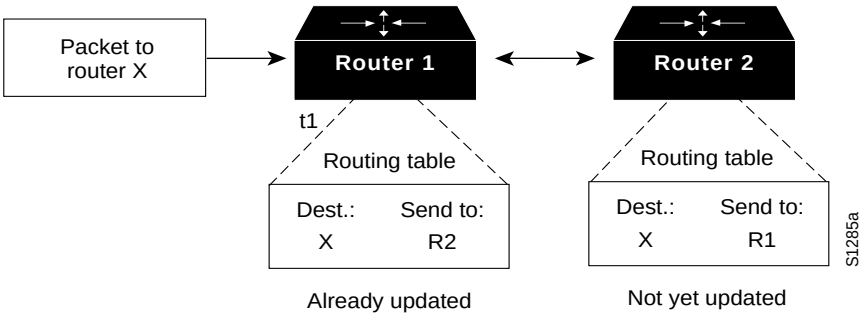
Routing algorithms must be robust. In other words, they should perform correctly in the face of unusual or unforeseen circumstances such as hardware failures, high load conditions, and incorrect implementations. Because routers are located at network junction points, they can cause considerable problems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

Rapid Convergence

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

Figure 2-3 shows a routing loop. In this case, a packet arrives at Router 1 at time t1. Router 1 has already been updated and so knows that the optimal route to the destination calls for Router 2 to be the next stop. Router 1 therefore forwards the packet to Router 2. Router 2 has not yet been updated and so believes that the optimal next hop is Router 1. Router 2 therefore forwards the packet back to Router 1. The packet will continue to bounce back and forth between the two routers until Router 2 receives its routing update or until the packet has been switched the maximum number of times allowed.

Figure 2-3 Slow Convergence and Routing Loops



Flexibility

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

Types

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

- Static or Dynamic
- Single-Path or Multipath
- Flat or Hierarchical
- Host-Intelligent or Router-Intelligent
- Intradomain or Interdomain
- Link State or Distance Vector

Static or Dynamic

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

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

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

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

Single-Path or Multipath

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

Flat or Hierarchical

Some routing algorithms operate in a flat space, while others use routing hierarchies. In a flat routing system, all routers are peers of all others. In a hierarchical routing system, some routers form what amounts to a routing backbone. Packets from nonbackbone routers travel to the backbone routers, where they are sent through the backbone until they reach the general area of the destination. At this point, they travel from the last backbone router through one or more nonbackbone routers to the final destination.

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

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

Host-Intelligent or Router-Intelligent

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

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

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

Intradomain or Interdomain

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

Link State or Distance Vector

Link state algorithms (also known as *shortest path first* algorithms) flood routing information to all nodes in the internetwork. However, each router sends only that portion of the routing table that describes the state of its own links. Distance vector algorithms (also known as *Bellman-Ford* algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers.

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

Metrics

Routing tables contain information used by switching software to select the best route. But how, specifically, are routing tables built? What is the specific nature of the information they contain? How do routing algorithms determine that one route is preferable to others?

Routing algorithms have used many different metrics to determine the best route. Sophisticated routing algorithms can base route selection on multiple metrics, combining them in a single (hybrid) metric. All of the following metrics have been used:

- Path Length
- Reliability
- Delay
- Bandwidth
- Load
- Communication Cost

Path Length

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

Reliability

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

Delay

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

Bandwidth

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

Load

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

Communication Cost

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

Routed vs. Routing Protocols

Confusion about the terms *routed* protocol and *routing* protocol is common. Routed protocols are protocols that are routed over an internetwork. Examples of such protocols are the *Internet Protocol* (IP), *DECnet*, *AppleTalk*, *NetWare*, *OSI*, *Banyan VINES*, and *Xerox Network System* (XNS). Routing protocols are protocols that implement routing algorithms. Put simply, they route routed protocols through an internetwork. Examples of these protocols include *Interior Gateway Routing Protocol* (IGRP), *Enhanced Interior Gateway Routing Protocol* (EIGRP), *Open Shortest Path First* (OSPF), *Exterior Gateway Protocol* (EGP), *Border Gateway Protocol* (BGP), *OSI Routing*, *Advanced Peer-to-Peer Networking*, *Intermediate System to Intermediate System (*IS-IS), and *Routing Information Protocol* (RIP). Routed and routing protocols are discussed in detail later in this publication.