letting it remain the best choice for others. The slopes, offsets, and breakpoints for the curves in Figure 4.21 were arrived at by a great deal of trial and error, and they were carefully tuned to provide good performance.

We end our discussion of routing metrics with a dose of reality. In the majority of real-world network deployments at the time of writing, metrics change rarely if at all, and only under the control of a network administrator, not automatically as was described above. The reason for this is partly that conventional wisdom now holds that dynamically changing metrics were too unstable, even though this probably need not be true. Perhaps more significantly, many networks today lack the great disparity of link speeds and latencies that prevailed in the ARPANET. Thus, static metrics are the norm. One common approach to setting metrics is to use a constant multiplied by \( (1/\text{link\_bandwidth}) \).

**Monitoring Routing Behavior**

Given the complexity of routing packets through a network of the scale of the Internet, we might wonder how well the system works. We know it works some of the time because we are able to connect to sites all over the world. We suspect it doesn't work all the time, though, because sometimes we are unable to connect to certain sites. The real problem is determining what part of the system is at fault when our connections fail: Has some routing machinery failed to work properly, is the remote server too busy, or has some link or machine simply gone down?

This is really an issue of network management, and while there are tools that system administrators use to keep tabs on their own networks—for example, see the Simple Network Management Protocol (SNMP) described in Section 9.1.4—it is a largely unresolved problem for the Internet as a whole. In fact, the Internet has grown

**4.2.5 Routing for Mobile Hosts**

Looking back over the preceding discussion of how IP addressing and routing works, you might notice that there is an implicit assumption about the mobility of hosts, or rather the lack of it. A host's address consists of a network number and a host part, and the network number tells us which network the host is attached to. IP routing algorithms tell the routers how to get packets to the correct network, thus enhancing the scalability of the routing system by keeping host-specific information out of the routers. So what would happen if a host were disconnected from one network and connected to another? If we didn't change the IP address of the host, then it would become unreachable. Any packet destined for this host would be sent to the network that has the appropriate network number, but when the router(s) on that network tried to deliver the packet to the host, the host would not be there to receive it.

The obvious solution to this problem is to provide the host with a new address...
when it attaches to a new network. Techniques such as DHCP (described in Section 4.1.6) can make this a relatively simple process. In many situations this solution is adequate, but in others it is not. For example, suppose that a user of a PC equipped with a wireless network interface is running some application while she roams the countryside. The PC might detach itself from one network and attach to another with some frequency, but the user would want to be oblivious to this. In particular, the applications that were running when the PC was attached to network A should continue to run without interruption when it attaches to network B. If the PC simply changes its IP address in the middle of running the application, the application may not continue to function correctly, because the remote end has no way of knowing that it must now send the packets to a new IP address.

The question of whether an application will continue to operate correctly following a change of IP address of one endpoint is a complex one. For example, if the application is a client-server interaction (such as web browsing) and the client’s address changes, it is likely that nothing would break, except perhaps a response from the server that was partially complete at the time of the move. By contrast, a peer-to-peer application, such as a voice over IP telephone call, would quite likely fail as a result of the address change unless some special action was taken by the application.

Ideally, we would like all applications to keep working correctly when an endpoint moves and for this process to be reasonably transparent to the applications. The proce-
dures that are designed to address this problem are usually referred to as “Mobile IP” (which is also the name of the IETF working group that defined them).

The Mobile IP working group made some important design decisions at the outset. In particular, it was a requirement that the solution would work without any changes to the software of nonmobile hosts or the majority of routers in the Internet. This sort of approach is frequently adopted in the Internet. Any new technology that requires a majority of routers or hosts to be modified before it can work is likely to face an uphill battle for acceptance.

While the majority of routers remain unchanged, mobility support does require some new functionality in at least one router, known as the home agent of the mobile node. This router is located on the “home” network of the mobile host. The mobile host is assumed to have a permanent IP address, called its home address, which has a network number equal to that of the home network, and thus of the home agent. This is the address that will be used by other hosts when they send packets to the mobile host; since it does not change, it can be used by long-lived applications as the host roams.

In many cases, a second router with enhanced functionality, the foreign agent, is also required. This router is located on a network to which the mobile node attaches itself when it is away from its home network. We will consider first the operation of mobile IP when a foreign agent is used. An example network with both home and foreign agents is shown in Figure 4.22.

Both home and foreign agents periodically announce their presence on the networks to which they are attached using agent advertisement messages. A mobile host may also solicit an advertisement when it attaches to a new network. The advertisement by the home agent enables a mobile host to learn the address of its home agent before it leaves its home network. When the mobile host attaches to a foreign network, it hears an advertisement from a foreign agent and registers with the agent, providing the address of its home agent. The foreign agent then contacts the home agent, providing a care-of address. This is usually the IP address of the foreign agent.

Figure 4.22 Mobile host and mobility agents.
At this point, we can see that any host that tries to send a packet to the mobile host will send it with a destination address equal to the home address of that node. Normal IP forwarding will cause that packet to arrive on the home network of the mobile node, on which the home agent is sitting. Thus, we can divide the problem of delivering the packet to the mobile node into three parts:

1. How does the home agent intercept a packet that is destined for the mobile node?

2. How does the home agent then deliver the packet to the foreign agent?

3. How does the foreign agent deliver the packet to the mobile node?

The first problem might look easy if you just look at Figure 4.22, in which the home agent is clearly the only path between the sending host and the home network, and thus must receive packets that are destined to the mobile node. But what if the sending node were on network 18, or what if there were another router connected to network 18 that tried to deliver the packet without its passing through the home agent? To address this problem, the home agent actually impersonates the mobile node, using a technique called proxy ARP. This works just like ARP as described in Section 4.1.5, except that the home agent inserts the IP address of the mobile node, rather than its own, in the ARP messages. It uses its own hardware address, so that all the nodes on the same network learn to associate the hardware address of the home agent with the IP address of the mobile node. One subtle aspect of this process is the fact that ARP information may be cached in other nodes on the network. To make sure that these caches are invalidated in a timely way, the home agent issues an ARP message as soon as the mobile node registers with a foreign agent. Because the ARP message is not a response to a normal ARP request, it is termed a gratuitous ARP.

The second problem is the delivery of the intercepted packet to the foreign agent. Here we use the tunneling technique described in Section 4.1.8. The home agent simply “wraps” the packet inside an IP header that is destined for the foreign agent and transmits it into the internetwork. All the intervening routers just see an IP packet destined for the IP address of the foreign agent. Another way of looking at this is that an IP tunnel is established between the home agent and the foreign agent, and the home agent just drops packets destined for the mobile node into that tunnel.

When a packet finally arrives at the foreign agent, it strips the extra IP header and finds inside an IP packet destined for the mobile node. Clearly the foreign agent cannot treat this like any old IP packet because this would cause it to send it back to the home network. Instead, it has to recognize the address as that of a registered mobile node. It
then delivers the packet to the hardware address of the mobile node (e.g., its Ethernet address), which was learned as part of the registration process.

One observation that can be made about these procedures is that it is possible for the foreign agent and the mobile node to be in the same box, that is, a mobile node can perform the foreign agent function itself. To make this work, however, the mobile node must be able to dynamically acquire an IP address that is located in the address space of the foreign network. This address will then be used as the care-of address. In our example, this would have to be an address with a network number of 12. We have already seen one way in which a host can dynamically acquire a correct IP address using DHCP (Section 4.1.6). This approach has the desirable feature of allowing mobile nodes to attach to networks that don’t have foreign agents; thus, mobility can be achieved with only the addition of a home agent and some new software on the mobile node (assuming DHCP is used on the foreign network).

What about traffic in the other direction (i.e., from mobile node to fixed node)? This turns out to be much easier. The mobile node just puts the IP address of the fixed node in the destination field of its IP packets, while putting its permanent address in the source field, and the packets are forwarded to the fixed node using normal means. Of course, if both nodes in a conversation are mobile, then the procedures described above are used in each direction.

**Route Optimization in Mobile IP**

There is one significant drawback to the above approach, which may be familiar to users of cellular telephones. The route from sending node to mobile node can be significantly suboptimal. One of the most extreme examples is when a mobile node and the sending node are on the same network, but the home network for the mobile node is on the far side of the Internet. The sending node addresses all packets to the home network; they traverse the Internet to reach the home agent, which then tunnels them back across the Internet to reach the foreign agent. Clearly it would be nice if the sending node could find out that the mobile node is actually on the same network and deliver the packet directly. In the more general case, the goal is to deliver packets as directly as possible from sending node to mobile node without passing through a home agent. This is sometimes referred to as the triangle routing problem since the path from sender to mobile node via home agent takes two sides of a triangle, rather than the third side that is the direct path.

The basic idea behind the solution to triangle routing is to let the sending node know the care-of address of the mobile node. The sending node can then create its own tunnel to the foreign agent. This is treated as an optimization of the process just described. If the sender has been equipped with the necessary software to learn the care-of address and create its own tunnel, then the route can be optimized; if not, packets just follow the suboptimal route.
When a home agent sees a packet destined for one of the mobile nodes that it supports, it can deduce that the sender is not using the optimal route. Therefore, it sends a binding update message back to the source, in addition to forwarding the data packet to the foreign agent. The source, if capable, uses this binding update to create an entry in a binding cache, which consists of a list of mappings from mobile node addresses to care-of addresses. The next time this source has a data packet to send to that mobile node, it will find the binding in the cache and can tunnel the packet directly to the foreign agent.

There is an obvious problem with this scheme, which is that the binding cache may become out-of-date if the mobile host moves to a new network. If an out-of-date cache entry is used, the foreign agent will receive tunneled packets for a mobile node that is no longer registered on its network. In this case, it sends a binding warning message back to the sender to tell it to stop using this cache entry. This scheme works only in the case where the foreign agent is not the mobile node itself, however. For this reason, cache entries need to be deleted after some period of time; the exact amount is specified in the binding update message.

Mobile routing provides some interesting security challenges. For example, an attacker wishing to intercept the packets destined to some other node in an internetwork could contact the home agent for that node and announce itself as the new foreign agent for the node. Thus, it is clear that some authentication mechanisms are required. We discuss such mechanisms in Chapter 8.

Finally, we note that there are many open issues in mobile networking. The security and performance aspects of mobile networks might require routing algorithms to take account of several factors when finding a route to a mobile host; for example, it might be desirable to find a route that doesn't pass through some untrusted network. There is also the problem of ad hoc mobile networks—enabling a group of mobile nodes to form a network in the absence of any fixed nodes. These continue to be areas of active research.

4.2.6 Router Implementation

In Section 3.4 we saw a variety of ways to build a switch, ranging from a general-purpose workstation with a suitable number of network interfaces to some sophisticated hardware designs. In general, the same range of options are available for building routers, many of which look something like Figure 4.23. The control processor is responsible for running the routing protocols discussed above, among other things, and generally acts as the central point of control of the router. The switching fabric transfers packets from one port to another, just as in a switch; and the ports provide a range of functionality to allow the router to interface to links of various types (e.g., Ethernet or SONET).

A few points are worth noting about router design and how it differs from switch design. First, routers must be designed to handle variable-length packets, a constraint that