16 Using NAT and IP Masquerading

Many residential and low-end business broadband accounts provide just one IP address, but an increasing number of users have multiple computers they'd like to connect to the Internet. For these users, a conventional router, as described in Chapter 15, is inadequate, because such a router works only when every computer on the local network has a legal IP address on the Internet as a whole. In order to get around this limitation, another approach is required. This approach goes by various names but is most commonly called *Network Address Translation (NAT)* or *IP masquerading*. Although these terms technically refer to techniques with somewhat different characteristics, for the purposes of this chapter they're pretty much identical, and I often use the term *NAT* as a stand-in for both terms. Both procedures let one computer, which I refer to as the *NAT router*, appear as a "front" for an entire network. The outside world knows about just one machine, but in reality there can be two, ten, a hundred, or more computers all accessing the Internet simultaneously.

It should come as no surprise that NAT is phenomenally popular among low-end broadband users, and this chapter describes its implementation. Before proceeding, though, you should read Chapter 14, which describes NAT in broad strokes and fits it into the overall broadband-sharing picture. Chapter 15, which describes conventional routing, is also helpful, because NAT is a form of routing, albeit an unusual one.

This chapter begins with an overview of NAT, including its capabilities, its limitations, and how to work around some of its limitations. The chapter then proceeds to describe NAT as implemented in Windows, MacOS, Linux, and dedicated hardware routers.

WARNING

Some ISPs explicitly prohibit the use of NAT with their services. You should therefore read your ISP's acceptable use policy document to be sure NAT is allowed. Although NAT is essentially impossible to detect with certainty by passive means, ISPs can do statistical analyses of your network traffic that would be likely to detect NAT in use. If you're found out as violating your service agreement, you may find yourself without a broadband connection to share.

UNDERSTANDING NAT AND IP MASQUERADING

Chapter 14 introduced the basics of NAT: It's a tool that allows an entire network to masquerade behind a single computer's IP address. Before you configure your network to use NAT, though, you should know something more about it, including details about the technique's functioning and some of its limitations. These limitations make NAT unsuitable for use in some situations and may restrict *which* NAT products you may use.

FUNCTIONS OF NAT AND IP MASQUERADING

The basic function of NAT is to translate one IP address to another. The technique is described in the Internet *Request for Comments (RFC)* document 1631 (`http://RFC.net/rfc1631.html`, among other places). The primary motivation for the development of NAT has been to conserve IP addresses, which are a limited resource. For you as a broadband user, though, NAT can conveniently extend your broadband connection's benefits to all the computers on a local network.

Hiding a Network Behind One IP Address

NAT is a feature that's implemented on a router. Normally, a router doesn't modify the packets it transfers from one network to another, except in limited ways. (Routers modify the *time-to-live [TTL]* value, for instance, which causes a packet to be dropped if it has made too many hops on the network.) NAT, however, makes more substantial changes. Specifically, a TCP/IP packet contains two addresses: the source address and the destination address. Suppose you have three computers but only one IP address.

You can operate an internal network with your three computers, but these machines require their own IP addresses from within a range of private addresses. If you configure one of your systems as a conventional router, the system will send packets on to a destination outside of your local network, but the source address in the packet will indicate a private IP address as the return address. Because private IP addresses are not officially part of the Internet, the receiving system will be unable to send a reply. This rather defeats the purpose of setting up a router.

NOTE

Three blocks of IP addresses are set aside for private networks: 192.168.0.0–192.168.255.255, 172.16.0.0–172.31.255.255, and 10.0.0.0–10.255.255.255. I normally use addresses from these ranges as example addresses for both internal and external networks, to avoid accidentally specifying somebody's real IP address in an example. In this chapter, I use the 192.168.0.0–192.168.255.255 address range for private internal networks and the other address ranges as fictitious external addresses.

With NAT, the router modifies the return address of the outgoing packets to indicate a valid return address—one that the NAT router will itself receive. Figure 16.1 illustrates one possible configuration. When 192.168.1.2 sends a packet to the Internet, the NAT router replaces that IP address with its own external address (172.19.16.203). The ultimate recipient therefore sees a valid return address and so can send a reply that reaches the NAT router.

The return packet also requires modification and housekeeping. The outgoing packet was sent from a particular *port* on 192.168.1.2. (A port is a virtual construct that allows a computer to keep track of connections, similar to extensions on a business telephone line.) The NAT router also modifies the source port number to correspond to a port on the router's outside interface. Replies are therefore directed to this same port, which is how the NAT router knows to which internal system to send the reply—the NAT router remembers which external port is associated with which internal computer and port. When a reply packet arrives, the NAT router can alter its destination IP address and send it on to the correct internal system.

The NAT system itself can also participate in networking. In this case, though, there's no need for masquerading; the NAT router can use its true external IP address (or its true internal address, when communicating with internal systems). It's therefore possible to use a system that serves other duties as a NAT router. This practice, however, increases the security risks because it exposes a normal system directly to the Internet. It's

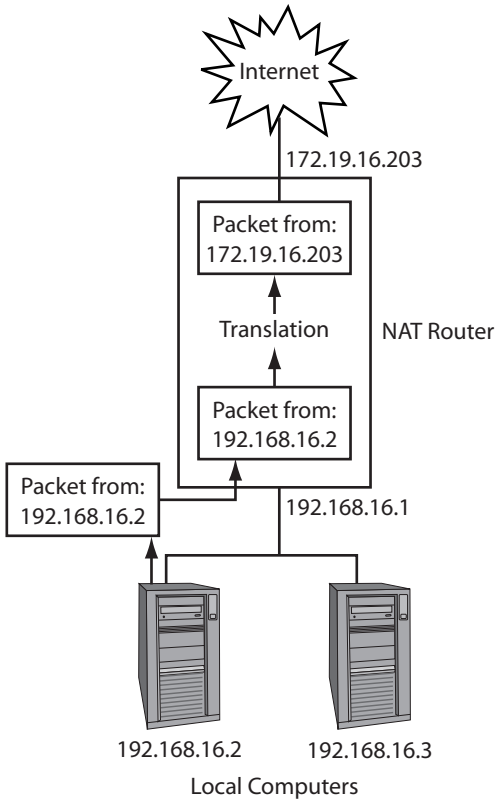

Figure 16.1 The NAT router tampers with outgoing packets so that they contain return addresses that are more desirable in one way or another.

often desirable to protect your main work systems behind a dedicated NAT router or firewall, as described shortly.

If all this works correctly, the internal computers see the NAT router as an ordinary router; they don't realize that their packets are being manipulated. Similar comments apply to outside systems; these computers see only a single computer. Unlike a conventional router, a NAT router doesn't require that the ISP's router know that the NAT router is a router for other systems; there's just one IP address involved, so the ISP's router treats the NAT router as an ordinary computer. These characteristics make NAT extremely appealing to broadband users with small local networks. There are, however, some caveats, which are described shortly, particularly in the sections Running Servers Behind a NAT Router and When to Use NAT.

Changing One IP Address to Another

The process I've just described is the way that NAT is most often used with broadband accounts. As described, this is a many-to-one mapping—many internal IP addresses are linked to a single external IP address. NAT, however, is capable of more complex configurations, as illustrated in Figure 16.2. In this network, the NAT router is recognized as a router to the outside world, or at least is configured with multiple external IP addresses. The computers on the internal network are known to each other by their private IP addresses, but the NAT router assigns outgoing packets a specific valid external IP address, as indicated in the figure. The address 192.168.1.2 receives no external address (it's not allowed to communicate with the outside world); 192.168.1.3 receives the 172.19.16.101 external address, and all external communications directed to this address are sent on to this system, and 192.168.1.4 receives the NAT router's own external address, as in a simpler NAT configuration.

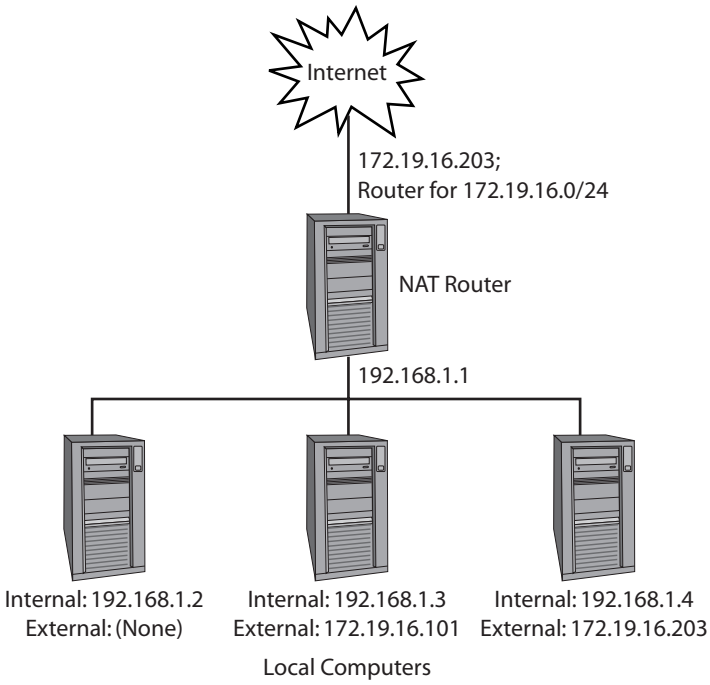

Figure 16.2 You can use NAT with multiple external IP addresses to customize how computers can be addressed.

This more complex form of NAT can be used to help control access to systems that may be configured as servers. For instance, in Figure 16.2, 192.168.1.3 might be running a server that should be accessible to the outside world, but the other systems on the internal network should not be so accessible. It's even possible to use NAT as a simple form of load balancing; the NAT router can redirect alternate incoming requests to different internal systems, to spread the effort of serving a popular Web site or the like.

Different NAT implementations have different capabilities when it comes to many-to-many connections. Many NAT tools are designed around the assumption of a many-to-one configuration, but others can be made to support these more complex configurations. Consult your tool's documentation for details.

As a general rule, you won't be using NAT in this complex way on a typical broadband account. A simpler many-to-one mapping is quite adequate for most uses. It's even possible to run servers on such a configuration, as described shortly.

Security Implications of NAT

NAT is inherently asymmetrical, at least as used in a many-to-one configuration. Computers on your local network can initiate connections with outside systems, because the NAT router keeps track of which ports are in use by which internal systems, and so can redirect return packets appropriately. To the outside world, though, there's just one computer—the NAT router. This fact provides an important security advantage for NAT. Because outsiders can directly access only the NAT router, systems on the private local network are protected from outside attack. An outside cracker cannot send packets directly to any internal system. Thus, even if the internal computers are all running buggy servers with well-known security exploits, a basic NAT configuration can keep the network from being invaded by outsiders. In fact, a NAT router closely resembles a firewall computer in this respect—so much so, in fact, that many hardware and software NAT products are sold as firewalls. (In truth, the line between NAT and at least certain types of firewall is blurry, because the same networking tools can be used to implement both.)

These facts should *not*, however, be taken as an excuse for lax internal security. The security benefits of NAT are more than theoretical, but they're less than perfect. A NAT-protected network can be compromised in several ways, including:

- Port-forwarded servers—If a NAT router is configured to allow access to an internal server, as described shortly, and if that server is buggy or

misconfigured, it may be possible to break into the internal system despite the NAT router.

- Many-to-many mappings—If you use NAT with a many-to-many mapping, any system that's completely exposed to the Internet, even if under an IP address other than the one it uses internally, can be compromised as easily as if it were directly connected to the Internet.
- NAT router compromise—The NAT computer itself may be attacked. This is particularly likely to be successful if you run servers on the NAT router, which is why such a practice is inadvisable. In fact, the simpler the NAT router, the better.
- Trojans—A *Trojan horse*, or *Trojan* for short, is a program that appears to be one thing but in fact is a cracker's tool. Trojans sometimes initiate outgoing connections that crackers can then use to access your system or attack other systems. A NAT router is unlikely to provide any protection against Trojans.

On the whole, then, a NAT router can be an important security measure on a broadband connection. It should not, however, be considered a guarantee of safety. If you're lax about your internal security, a NAT router could conceivably do more harm than good, by providing you with an inflated sense of security. This is particularly true if the NAT router itself does double duty as a normal computer or a server. Such systems are often loaded with tools that can help a cracker, either by presenting security holes through which the computer can be compromised or by providing tools the cracker can use to invade the rest of your systems. If you must use a normal system as a NAT router, the very least you should do is to take basic security precautions on it (as described in Chapter 19) and run a firewall package on that system (as described in Chapter 20).

RUNNING SERVERS BEHIND A NAT ROUTER

There are two main ways to run servers for external use on a broadband connection that uses a NAT router:

- On the NAT router—The NAT router itself can host the server. This practice, although easy to configure, is potentially dangerous. As just noted, a bug or misconfiguration of the server can lead to a compromise of the entire NAT router system. If you're running a server on this system, it's presumably a regular computer with tools that a cracker could use to compromise the rest of your network.
- On a forwarded port—It's possible to configure a NAT router to *forward* one or more ports from the router's external interface to an internal

computer. When the NAT router receives a request on that port, the router passes the request on to a specified internal computer. In some sense, this is all that a many-to-many configuration is—the NAT router simply forwards *all* the ports directed to a specific IP address to another system. Because most servers listen to a single port, though, forwarding just one port is sufficient to handle most server types.

Clearly, the second option is generally preferable. It usually works, too. For a few types of servers, though, forwarded ports don't work well, or at least have drawbacks. For instance, Secure Shell (SSH) encodes IP addresses in an encrypted way that NAT can't touch. Therefore the IP address claimed in the TCP/IP packet headers doesn't match that encoded by SSH itself. Although you can configure your SSH servers and clients to ignore this discrepancy, at the very least you're likely to get a warning about possible tampering whenever you try to make a connection.

Actually telling your NAT software or hardware to enable port forwarding requires you to know something about the system you're using. This detail is covered for each of several systems in the following sections. In order to forward a port, though, you must know what port to forward. Table 16.1 summarizes the ports used by some common network protocols, but this list is far from complete. A more complete list can be found at `http://www.isi.edu/in-notes/iana/assignments/port-numbers/`, among other places. Note that there are actually parallel sets of ports, the most common being TCP and UDP. Most servers use the TCP port. A few servers use the UDP port. Most servers, whichever port they use, are assigned the same number for both ports. A few servers—particularly X and VNC—use multiple ports. Usually only the first of a given type is necessary; subsequent ports are used for multiple connections or instances of the server.

NOTE

With rare exceptions, you do *not* need to enable port forwarding to use *client* software from within your network. You need only to perform port forwarding when you want to run *server* software. Keep in mind that the client/server relationship for X servers is unusual; the X server runs on the computer at which you sit, which is backward from the way most servers operate. You therefore need to enable port forwarding of port 6000 (or higher) if you intend to sit behind a NAT router and run X programs based on systems on the Internet at large.

One important point to consider about running servers behind a NAT router is that port forwarding usually requires that the server system have

Table 16.1 Port Numbers Likely to Be Used by Broadband Servers

Port Number	Use
20	FTP data channel
21	FTP server control
22	SSH
23	Telnet
25	Simple Mail Transfer Protocol (SMTP)
53	Domain Name System (DNS) server
80	Hypertext Transfer Protocol (HTTP; Web server)
109	Post Office Protocol 2 (POP-2; mail retrieval)
110	Post Office Protocol 3 (POP-3; mail retrieval)
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol (NTP)
137	NetBIOS name service
138	NetBIOS datagram service
139	NetBIOS session service
143	Interactive Mail Access Protocol (IMAP; mail retrieval)
177	X Display Manager Control Protocol (XDMCP; remote X login)
220	Interactive Mail Access Protocol 3 (IMAP3; mail retrieval)
5800–5899	Virtual Network Computing (VNC) remote Web browser access
5900–5999	Virtual Network Computing (VNC) remote GUI login
6000–6063	X servers

a static IP address on your local network. Many NAT implementations, though, include DHCP servers that aren't guaranteed to provide the same IP address to a computer every time it boots. You may therefore need to run the server with a static IP address, ignoring the DHCP server, or even to run all your internal systems without using the NAT package's DHCP server. You can usually get away with this by assigning the server system an IP address at the high end of the range used by the NAT program's DHCP server or by configuring the NAT DHCP server to assign addresses from within a restricted range. Some DHCP servers, though, can assign the same IP address to a computer every time it boots.

WHEN TO USE NAT

For many broadband users, NAT really is the best thing since sliced bread. NAT neatly solves the problem of providing access to the broadband link from multiple computers. When you have an office with a handful of systems

or when you want to provide access from computers belonging to various family members, NAT can do the job—*if* the job is to use any of the dozens of supported network protocols in client mode or even to allow a server to run from behind a NAT router. There are, however, some caveats and limitations to what NAT can accomplish. These include:

- Unsupported protocols—NAT alters the data packets going into and out of a network. Some protocols react badly to this or require servers on both sides of a connection. In theory, it's often possible to modify the NAT software to handle a new protocol. In practice, any given NAT implementation may or may not work with any given protocol. All versions of NAT support common protocols like SMTP, HTTP, and FTP, but when you get into the realm of specific game servers, videoconferencing tools, and so on, you may need to hunt even to discover whether the protocol works with a given NAT implementation. (Videoconferencing tools are particularly prone to incompatibilities with NAT.) Note that NAT servers are not a monolithic bunch in this respect; one tool may support a given protocol, but another may not.
- Bandwidth issues—As with other broadband-sharing options, NAT doesn't increase the bandwidth you have available; it is a *sharing* tool. If ten people try to use a 1,500Kbps link simultaneously, each will get only 150Kbps. On the other hand, you can split the link ten ways, but if only two of those people try to use it *simultaneously*, each will get 750Kbps.
- Servers—Although it's possible to run servers from behind a NAT router, doing so requires extra configuration and can sometimes cause problems, as already noted. If you want to run two different servers of the same type (say, two different mail servers), you may not be able to do it with the standard many-to-one NAT configuration. Sometimes you can work around this limitation by running one server on a nonstandard port or by configuring one server to handle the duty you had planned for both. Some NAT implementations also support *triggered* port forwarding, in which the NAT router uses additional rules to decide to which internal system to send a given packet.

On the whole, NAT's main drawback for most broadband users is its incompatibility with some protocols. If you plan to use any network tools beyond the standard set (Web browsing, e-mail, FTP, Telnet, and Usenet news), you should investigate your NAT product's ability to handle your more unusual protocols. Still, most NAT products do a good enough job that compatibility isn't a major problem. In fact, many people buy NAT routers for use as firewalls with a single computer and experience no problems.

NAT TOOLS IN WINDOWS

As the most popular operating system in the world, Microsoft Windows is often pressed into service as a platform for NAT. This is particularly true on very small networks of just two or three computers when the expense of dedicated hardware is an issue. Windows also makes an excellent choice when you're stuck with an internal or USB-interfaced modem, because Windows may be the only OS with drivers for such a device. Fortunately, Windows is up to the task of handling NAT, and in fact recent versions of the OS ship with NAT support.

TOOLS FOR PERFORMING NAT IN WINDOWS

Windows supports several different NAT products. Because NAT must interface closely with the OS, many NAT products are specific to one line of Windows (that is, Windows 9*x*/Me or Windows NT/2000). Many NAT products also support firewall features, and in fact many of these programs are marketed primarily as firewalls. Examples include:

- Microsoft ICS—Microsoft's *Internet Connection Sharing (ICS)* is the NAT feature that's built into Windows 98SE (but not the original release of Windows 98), Windows Me, and Windows 2000. Because it's built into the OS, it's quite popular, and it's well worth trying before you buy something else. ICS includes a simple DHCP server, DNS proxy, and other helpful features but lacks the extensive firewall-like controls of several other NAT products. Microsoft maintains a description of ICS, including several helpful links, at `http://support.microsoft.com/support/kb/articles/Q234/8/15.ASP`. The ICS implementations of Windows 98SE/Me and Windows 2000 are not identical; Windows 2000's ICS is more capable than that in Windows 98SE/Me, although you can extend the Windows 98SE/Me version, as described shortly.
- All Aboard!—InterNetShare's AllAboard! product supports all versions of Windows since Windows 95, including NT and 2000. This product comes in two versions (plus another version for Linux). Only the Linux version supports firewall features. AllAboard! has a good reputation for dealing with PPPoE connections, so you may want to investigate it if you have problems with ICS that you believe to be PPPoE-related. The low-end version costs $50–$160 for 3–10 users, and the midrange product costs $220–$350 for 6–25 users. (You can obtain licenses for additional users at extra cost.) Read more at `http://www.internetshare.com`.
- Sygate Home Network and Office Network—These products are designed for home or office use and are licensed on a per-client basis, starting at $40

for 3 users and $450 for 25 users, respectively. Sygate emphasizes its capacity to handle NAT through just one Ethernet card, but as I stated earlier, I recommend against such a configuration. These products work with any version of Windows since Windows 95. You can read more at `http://www.sygate.com`.

- Internet Manager Firewall—This product, from Elron Software (`http://www.elronsoftware.com`), is marketed as a firewall for Windows NT supporting fairly large networks. It also supports NAT and VPN features, however.
- CheckPoint Firewall-1 and VPN-1 Gateway—These two products both support NAT features in Windows NT, although they're marketed as a firewall and a VPN solution, respectively. You can read more at `http://www.checkpoint.com`.

This is just a sampling of the available NAT solutions for Windows. A more complete list is available from `http://www.uq.net.au/~zzdmacka/the-nat-page/nat_windows.html`. Chances are good that ICS will satisfy your needs; however, it's available only in recent versions of Windows. If you're running the original Windows 98 or earlier, or Windows NT 4.0 or earlier, you'll have to either upgrade your OS or use another product.

CONFIGURING WINDOWS FOR NAT

Because Windows now comes with ICS, this is the Windows NAT solution I describe here. The Windows 2000 version is more readily configured to perform port forwarding than the Windows Me version. Other products require installation and have different configuration procedures, but at least some of the principles are the same. You should be aware that Windows insists on using the 192.168.0.1 address when it's configured to do ICS. If your internal network currently uses something other than the 192.168.0.0/24 network, you'll have to reconfigure it.

Configuring Windows Me for NAT

After you've made any necessary changes to your local network, such as removing any other system that might be using the 192.168.0.1 address, follow these steps to configure ICS in Windows Me:

1. Install two network interfaces and the drivers for both. Configure the broadband network interface as you normally would. Windows will configure your local network's interface during subsequent steps.

2. ICS may or may not be installed by default. If it's not, you can install it using the Add/Remove Programs icon in the Control Panel. Click the Windows Setup tab, select Communications, and click Details. ICS is one of the options in the resulting dialog box. If you need to install it, do so.
3. After installing ICS, you can configure it through the Home Networking Wizard. This wizard may have started after you installed ICS. If not, start it by choosing Start ➢ Programs ➢ Accessories ➢ Communications ➢ Home Networking Wizard.
4. Click Next in the introductory window. This produces a display in which you select the network card that's connected to your ISP, as shown in Figure 16.3.
5. Click Next. The wizard lets you enable ICS. Select Yes to the question, and select the network card that connects to your internal network.

NOTE If it doesn't ask you if you want other computers to use the connection, chances are ICS isn't installed on your computer. Go back and check this detail.

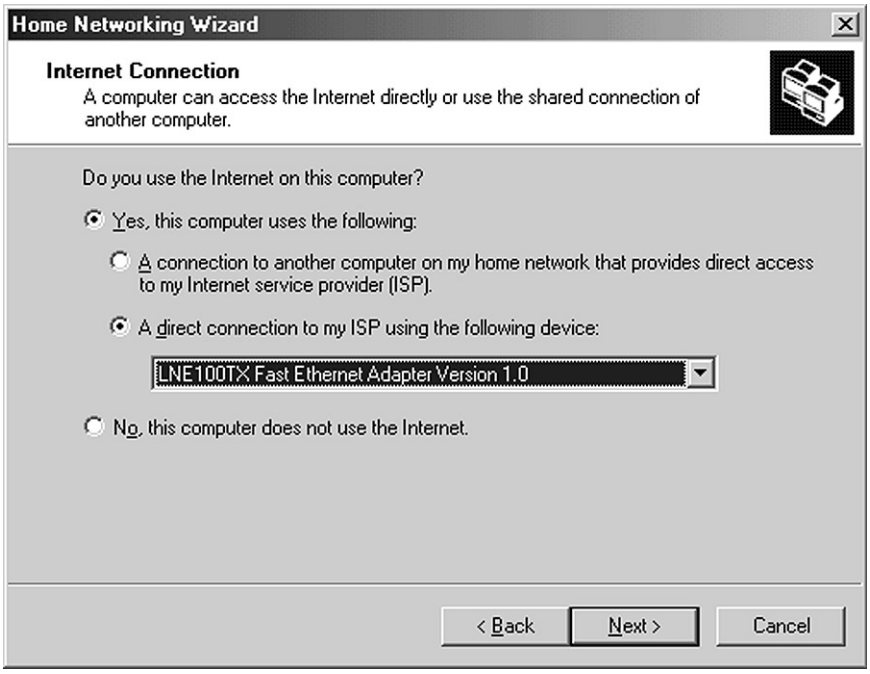

Figure 16.3 Microsoft's Home Networking Wizard helps you configure ICS and other networking features.

6. Click Next. Windows lets you enter a computer and workgroup name for use with SMB/CIFS networking. (This and subsequent steps relate to network details that don't directly affect ICS.)
7. Click Next. You can set up file and printer sharing with this dialog box.
8. Click Next. Windows lets you create a home networking setup disk, which contains files to help configure other Windows systems to use the same network settings you've just configured.

TIP

Although ICS works fine with some OSs without special configuration on their part, Windows 95 and 98 may not work well (or at all) with an ICS gateway unless they're configured with the tools Windows Me makes available at this point. It's therefore best to create this floppy and store it in a safe place.

9. Click Finish. Windows prompts you to reboot the system. Do so.

This procedure enables ICS on the computer, including running a small DHCP server on the local network side and configuring that network card to use the 192.168.0.1 network address. If you configure another computer to use DHCP on that network (using the floppy disk created in step 7, if necessary), when you next boot that computer, it will obtain an address from the ICS system and will be able to access the Internet.

Unfortunately, ICS as delivered with Windows 98/Me doesn't support port forwarding. Fortunately, at least two tools add port-forwarding support to ICS:

- ICSConfig—This is a freeware utility available from `http://lynx.neu.edu/a/amccombs/`. It's configured through GUI tools similar to those used in Windows 2000. ICSConfig also lets you change the internal network address or disable the internal DHCP server.
- ICS Configuration—This is a shareware ($10) utility, available from `http://www.practicallynetworked.com/sharing/ics/icsconfiguration.htm`. This tool is configured through "mapping files" for specific applications. Many samples are available from the ICS Configuration Web site.

Configuring Windows 2000 for NAT

In Windows 2000, the procedure for configuring ICS is somewhat different from that in Windows Me. ICS is installed by default in Windows 2000, and there's no wizard to configure it. To enable ICS, follow these steps:

1. Install two network cards and the drivers for both. Configure your broadband interface as you normally would. You can configure the local network interface if you like, but Windows will reconfigure it during this procedure.
2. Open the Network and Dial-up Connections window from the Control Panel.
3. Right-click the icon that corresponds to your *external* (broadband) network connection, and click Properties from the resulting pop-up menu. This brings up the connection's Properties dialog box.

WARNING

If you select the icon for your internal network, you'll reconfigure your system to perform NAT in the wrong direction, as if the internal network were the Internet.

4. Click the Sharing tab in the Properties dialog box.
5. Check the Enable Internet Connection Sharing for this Connection check box and click OK. Windows warns that the IP address for the internal network will be set to 192.168.0.1 and that other systems on the local network should be configured to use DHCP to obtain their IP addresses. Note that this refers to the IP address for the network adapter that you did *not* select in step 3, which can be confusing.

At this point, Windows 2000 should be configured as a NAT system. Basic connections from your local network to the outside should work, at least once you've reconfigured your local systems to use DHCP and rebooted them or restarted their networking systems.

If you want to make a local server available to the outside world, you can configure Windows 2000's port-forwarding abilities as follows:

1. Follow steps 2–3 in the preceding instructions to get the Sharing tab in the Properties dialog box. Click the Settings button. Windows displays the Internet Connection Sharing Settings dialog box.
2. Click the Services tab in the Internet Connection Sharing Settings dialog box. It should now resemble the one shown in Figure 16.4.
3. Locate the name of the server you want to share, and click the check box next to its name. Windows displays a dialog box similar to that shown in Figure 16.5, in which you can enter the IP address or hostname of the computer that sports the server.

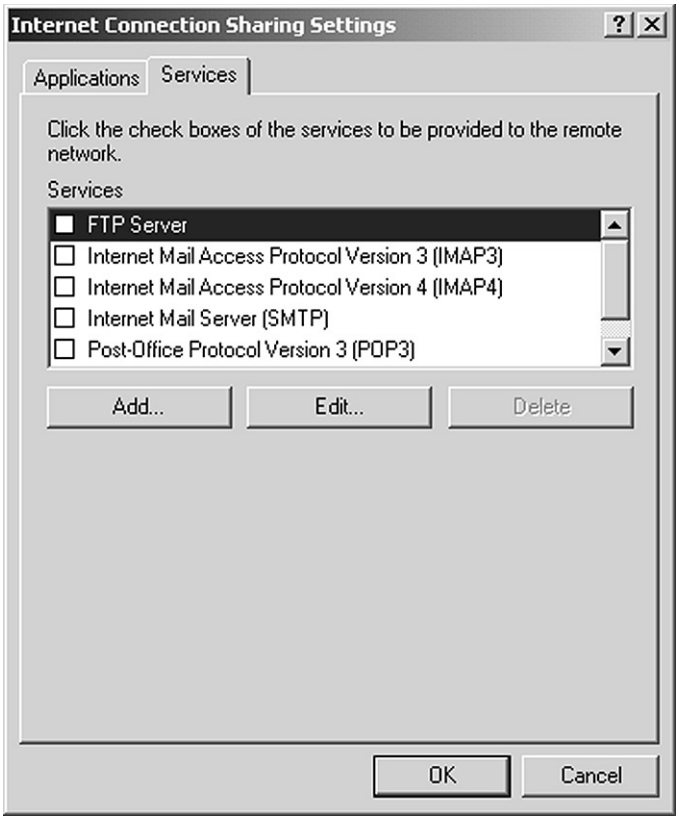

Figure 16.4 Windows 2000 comes with the ability to share any of several services from inside a NAT network.

TIP

The Windows 2000 ICS DHCP server has no configurable provisions to guarantee that a given computer receives the same IP address on every reboot. You may therefore want to configure your server system manually. To avoid conflicts with the DHCP-provided addresses, give this system the address 192.168.0.254, which is the last address the Windows 2000 system would normally allocate. This should prevent conflicts. Alternatively, you can configure your internal network to ignore the DHCP server and configure all your internal IP addresses manually.

4. Click OK in the three dialog boxes. ICS now forwards attempts to reach it to the internal computer you've specified.

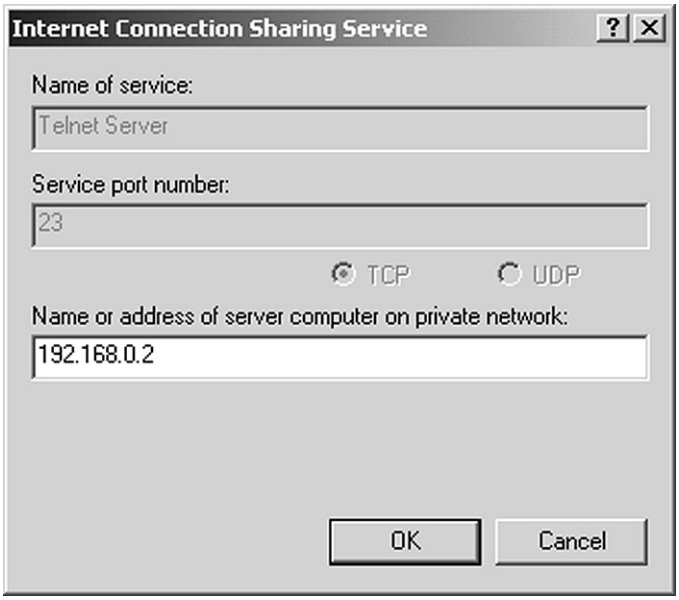

Figure 16.5 Enter the name or address of the computer that hosts the server.

If your service isn't predefined in the Internet Connection Sharing Services window (see Figure 16.4), you can add it by following the preceding steps; however, you must click Add in step 3 rather than click the check box next to the name of the service. Windows responds by allowing you to enter all the information in the Internet Connection Sharing Service window (see Figure 16.5), not just the destination address.

NAT TOOLS IN MACOS

MacOS Classic doesn't ship with any NAT tools, unlike Windows. MacOS X ships with NAT tools, but in mid-2001 they lack GUI front ends, so you'll need to configure some UNIX-style scripts to get NAT working. It's likely that somebody (perhaps one of the players in the MacOS Classic NAT arena) will provide an easier-to-use NAT interface for MacOS X.

TOOLS FOR PERFORMING NAT IN MACOS

There are four major NAT packages for MacOS:

- IPNetRouter—This package, from Sustainable Softworks (`http://www.sustworks.com`), is marketed as a general-purpose router package for

MacOS Classic. (The Web page indicates that the software runs in the Classic environment of MacOS X but is unsupported when run in this way.) This package includes a DHCP server, DNS forwarding, support for some firewall features, and PPPoE support. It costs $89.

- Internet Gateway—This Vicomsoft (`http://www.vicomsoft.com`) product is a MacOS Classic product that's marketed for businesses and costs $99 or more, depending on the number of users. It's similar to IPNetRouter in overall capabilities. This product is available with several different modules to suit different needs. It formerly went by the name *SoftRouter*.
- SurfDoubler—This MacOS Classic product is a low-cost ($35) relative of Vicomsoft's Internet Gateway. It's designed for residential use and supports only two IP addresses behind the NAT router.
- `natd` and `ipfw`—These are the names of the NAT tools that ship with MacOS X. As text-based programs, they're more intimidating to new users than the preceding tools are.

If your needs are modest, SurfDoubler may do the job. If you need to connect more than two systems, though, you'll be better served by IPNetRouter or Internet Gateway.

NOTE

The MacOS Classic products can also function as conventional (non-NAT) router control facilities, allowing you to configure a Macintosh to perform fairly advanced router functions.

CONFIGURING MACOS CLASSIC FOR NAT

As an example of MacOS Classic NAT configuration, this section describes IPNetRouter. Configuring Internet Gateway or SurfDoubler is conceptually similar, but of course many of the details do differ.

Enabling Basic NAT Features

To set up NAT, follow these steps:

1. Install and configure two network cards, as described in Appendix B. The card connected to your local network should be configured using a static IP address, and you should leave the gateway address field blank for this card. The card connected to your broadband modem should be configured in whatever way is appropriate for your ISP.

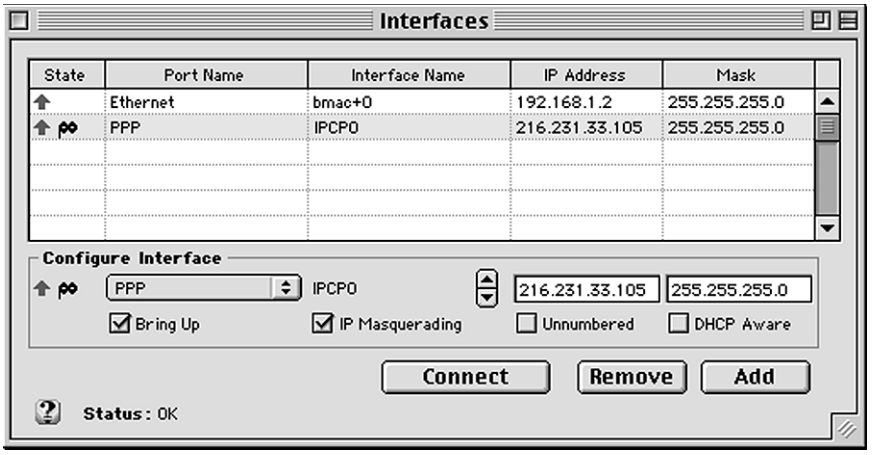

Figure 16.6 The IPNetRouter configuration panel lets you control multiple network interfaces, whether or not the system acts as a router.

2. Install the IPNetRouter software. The result is a folder that contains the IPNetRouter program and several documentation files.
3. Double-click the IPNetRouter icon to launch the program. IPNetRouter displays a list of interfaces in its configuration panel, as shown in Figure 16.6.
4. Select the interface that links to your ISP, check the IP Masquerading check button, and click Add. This enables the NAT feature for the interface. You should see a mask icon appear in the State field, as for the PPP interface in Figure 16.6.
5. Choose File ➢ Save As to save the configuration you've created. You can then start IPNetRouter with this configuration by double-clicking the configuration file.

You may need to quit from IPNetRouter and restart it or possibly even restart the computer, before it functions correctly as a NAT router. Once you've done this, IPNetRouter should provide basic NAT features.

Enabling DHCP in IPNetRouter

IPNetRouter does not provide DHCP services by default, unlike the ICS feature of Windows. You can enable this function as follows:

1. Choose Window ➢ DHCP to open the DHCP control panel.

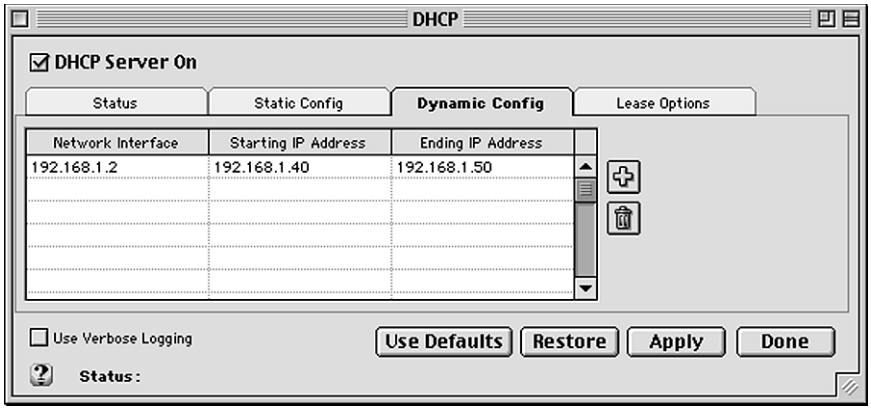

Figure 16.7 IPNetRouter's DHCP server provides options that let you control how it assigns IP addresses.

2. Click the Dynamic Config tab. Alternatively, you can use the Static Config tab to configure the system to provide the same IP address to a machine every time it requests one, but this requires that you enter each computer's network card *Media Access Control (MAC)* address. Figure 16.7 shows the DHCP control panel's Dynamic Config tab.
3. Enter the IP address of the interface on which you want the DHCP server to run and the range of IP addresses the server will deliver. For instance, Figure 16.7 shows the system configured to deliver IP addresses between 192.168.1.40 and 192.168.1.50, inclusive.
4. Click the Lease Options tab and enter the information requested there. In particular, you must enter the IP address of the DHCP server's network interface, the network mask (probably 255.255.255.0), the gateway IP address (probably the same as the DHCP server's network interface), and the IP address of any name servers.
5. Click Done to dismiss the DHCP window and activate the changes. The system should now respond to DHCP queries on the local network.

Configuring IPNetRouter Port Mapping

If you want any servers on your internal network to be accessible to the outside, you must configure IPNetRouter to perform port mapping. This can be done by choosing Window ➢ Port Mapping from the program's menu. This action produces the Port Mapping dialog box (see Figure 16.8). For each port you want mapped, you must enter several pieces of information in the Configure Entry area:

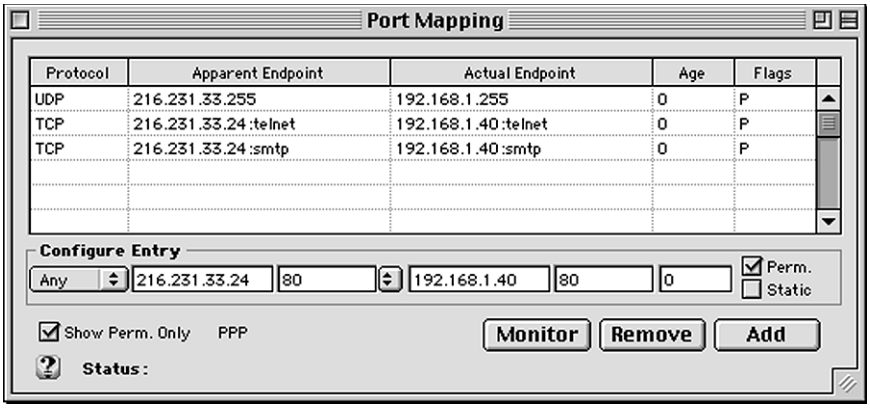

Figure 16.8 The IPNetRouter Port Mapping dialog box lets you enter information on specific external ports to be linked to specific internal IP addresses.

- Port type—The leftmost list selector lets you choose the port type, which will usually be TCP or UDP.
- External IP address—Enter the IP address of the external (broadband) interface in the next field.
- External port number—You can either enter the port number directly if you know it or select from a list of many common servers by clicking the list selector next to the port number field.
- Internal IP address—Enter the IP address of the internal system that's to receive the connection in the next field (which lists 192.168.1.40 in Figure 16.8).
- Internal port number—The internal port number usually corresponds to the external port number, but in some cases it might not—say, if you want to run a server on an unusual port number either internally or externally.
- Flags—The Perm and Static check boxes are used to indicate permanent and static entries, respectively. Chances are you want to check the Perm box. If you don't, the entry may be deleted after it has been unused for a specified period of 30-second intervals (you can specify this value with the preceding text-entry field).

When you're done entering this information, click the Add button to add the information to the port-mapping table. Thereafter, you should be able to reach an internal system from outside by specifying the Macintosh system's IP address.

IPNetRouter is fairly intelligent when it comes to dealing with dynamic external IP addresses. For instance, when entering the external IP address for port mapping, you can list the current address, but the system will automatically update this information if your external IP address changes.

If you want the system to start IPNetRouter at boot time, you can create a shadow of it or its configuration file in the `System Folder:Startup Items` folder. This causes IPNetRouter to run when you start the computer.

CONFIGURING MACOS X FOR NAT

To configure MacOS X for NAT, you'll need to dig into text files or wait for a GUI configuration tool to appear on the market. If you're at least somewhat comfortable with MacOS X's UNIX underpinnings and want to try the former, you can do so, but be sure you do all of this as `root` or some other administrative user. Follow these steps to get NAT running:

1. Configure your system to work correctly with both network interfaces using the normal MacOS X configuration tools.
2. Create a text file called `/etc/rc.natd`. This file should resemble the one shown in Listing 16.1. You may need to alter the broadband interface name (`en0`), particularly if you use PPPoE. (You can determine your interface name by typing **`ifconfig -a`**, which returns information on all the network interfaces. Ignore `lo0`; the remaining interfaces are your broadband and local network interfaces.)
3. Modify the owner and permissions on the `rc.natd` file so that it's executable. Typing **`chmod a+x /etc/rc.natd`** should do this.
4. Edit `/etc/hostconfig`, and change the value of `IPFORWARDING` from `-NO-` to `-YES-`.

Listing 16.1 Sample MacOS X NAT Script

```
/usr/sbin/natd -dynamic -interface en0
/sbin/ipfw -f flush
/sbin/ipfw add divert natd all from any to any via en0
/sbin/ipfw add pass all from any to any
```

You've now created a script that enables NAT on a MacOS X system. Before it will work, you'll need to enable the changes implemented in `/etc/hostconfig` in step 3. Unfortunately, MacOS X doesn't provide an easy way

to do this aside from rebooting, so you should probably reboot. Afterward, type **/etc/rc.natd** at a command prompt to run the script you've created. NAT should now be working.

If NAT works and you want to enable it automatically, move or copy the `/etc/rc.natd` script to `/System/Library/StartupItems`. I recommend you not do this until you're satisfied that it's working the way you want it to work.

You can learn more about both `natd` and `ipfw` by reading their UNIX man pages—type **man natd** or **man ipfw** in a command prompt window. Unfortunately, this documentation is pretty dense stuff. If you want your MacOS X system to deliver IP addresses via DHCP, you'll need to run a DHCP server on it. This task is similar to running such a server in Linux, as described in the section Enabling DHCP Features in Linux, but you'll need to locate a DHCP server for the system. A version is available from `http://publicus.lcma.com.au/macosx.html`, and there may be others floating around, as well.

MacOS X's NAT tools are the same as those included in the open source BSD versions of UNIX (FreeBSD, OpenBSD, and NetBSD). If you need to delve further into this matter, researching the issue as if you were running such a UNIX system may bear fruit.

NAT TOOLS IN LINUX

In the Linux world, NAT goes by the name *IP masquerading*. Technically, IP masquerading is a simpler form of NAT, but Linux is certainly capable of handling full-blown NAT features, including many-to-many associations, as described earlier. In fact, Linux's default NAT tools are also used for configuring packet-filter firewalls and other features. This section is restricted to describing Linux's NAT functionality, however.

TOOLS FOR PERFORMING NAT IN LINUX

The Linux kernel includes features that explicitly support NAT. These kernel features mean that enabling NAT support is usually just a matter of running a couple of simple commands. This support is bare-bones compared with that provided by NAT packages for Windows or MacOS. It's possible to get all the extras in Linux by running additional commands or servers, however. Commercial NAT packages that group everything

together into a single tool are also available for Linux. Some of the more common Linux NAT utilities and packages include:

- `ipfwadm`—This was the standard NAT configuration tool for the 2.0.*x* Linux kernels. Because these kernels are now obsolete, `ipfwadm` is no longer in widespread use, but you may find references to it in older documentation.
- `ipchains`—This tool was used to configure NAT with the 2.2.*x* Linux kernels. With the release of the 2.4.0 kernel in January 2001, `ipchains` became obsolete, but you might still use it if you're using an older kernel.
- `iptables`—This is the standard tool for configuring NAT on 2.4.*x* Linux kernels. This program should come with distributions that support the 2.4.*x* kernels, but if you've upgraded an older distribution, you may need to obtain it from `http://netfilter.kernelnotes.org`.
- All Aboard!—This commercial package, from InterNetShare (`http://www.internetshare.com`), is available for Windows and Linux, but the Linux version (called the *Advanced Edition*) is the top-of-the-line product, supporting many more advanced features. It carries a hefty $4,950 list price.
- ShareTheNet—This is another commercial package ($70) that provides an integrated approach to NAT and related components for Linux. You can read more about it at `http://www.sharethenet.com`.

You can find pointers to a few additional Linux NAT products, as well as NAT products for other UNIX-like OSs, at `http://www.uq.net.au/~zzdmacka/the-nat-page/nat_unix.html`.

CONFIGURING LINUX FOR NAT

Because the basic Linux NAT tools are so common and ship with most distributions, I describe their configuration here. Commercial tools are typically configured using integrated user interfaces that more closely resemble the Windows or MacOS Classic products described earlier. This section covers the use of `iptables`. The earlier `ipfwadm` and `ipchains` tools are similar, but they offer fewer features, and they use somewhat different syntax.

Linux Kernel Options for NAT

As mentioned earlier, the Linux kernel includes explicit support for NAT. This support is included in the default kernels provided by most Linux distributions. If you recompile your kernel, though, you should be aware of what these options are so that you enable them when you rebuild your

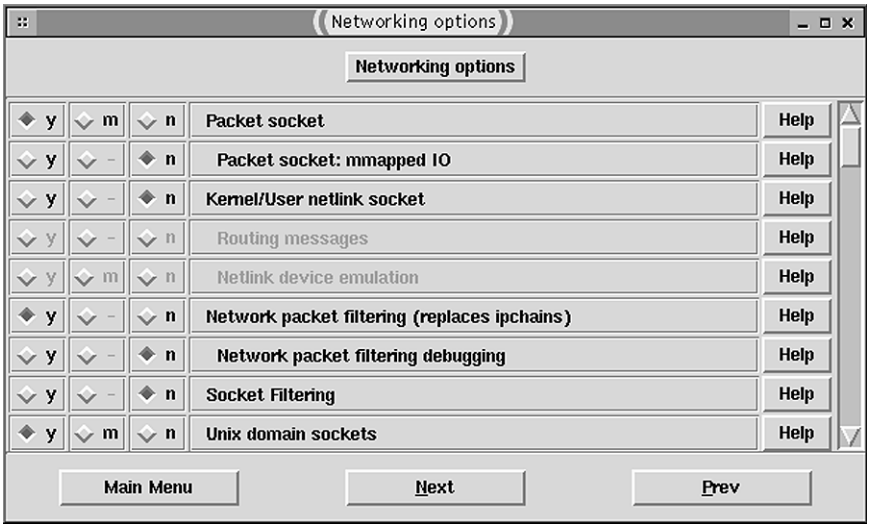

Figure 16.9 Many basic networking features are set in the Linux Networking Options kernel configuration menu.

kernel. If you fail to do this, any attempt to use NAT will fail. In the 2.4.*x* kernel series, you should enable the following features in addition to normal TCP/IP kernel features:

- Network Packet Filtering—The Network Packet Filtering option is in the main Networking Options configuration menu (see Figure 16.9).
- Connection Tracking—This option is available in the Netfilter Configuration menu off of the Networking Options menu. It's required for NAT. (All subsequent options are in the same Netfilter Configuration menu.)
- FTP Protocol Support—FTP is a tricky protocol for NAT. In Linux, support for FTP requires this special kernel module.
- IP Tables Support—This option is another that's required for NAT.
- Packet Filtering—Although not absolutely required for NAT, this option enhances the range of NAT features available to you. I recommend you enable it.
- REJECT Target Support—This suboption of Packet Filtering adds a rule that can be helpful in creating firewalls. It's therefore best to enable this feature.
- Full NAT—This option is required for many NAT features, including those described in this chapter.

- MASQUERADE Target Support—This suboption of the Full NAT option is required for IP masquerading—the form of NAT described here. Note that the Help option for this item implies that it's necessary only if you use a dynamic external IP address, but this is incorrect; it's required for IP masquerading whether or not your external IP address is dynamic.

Assorted other options are available, particularly in the Netfilter Configuration menu. Most of these relate to specific protocols or uses of NAT. You can read the help for these options if you're interested, but they aren't required for basic operation. Adding them into your kernel unnecessarily simply increases the kernel's size. Although a too-large kernel is undesirable, it's not a devastating problem, so feel free to add any options about which you're unsure.

If you compile any of these options as modules (by selecting the M compile option), you may need to load the appropriate module explicitly in your NAT start-up scripts. If you want to avoid doing this, you can compile the options into the main kernel file by selecting the Y compile option. Many Linux configurations load modules automatically, so even if these features are compiled as modules, you may not need to take any explicit steps to load them.

If you're using an older 2.0.*x* or 2.2.*x* kernel, the precise options required for NAT are somewhat different from those described here. Read the Help items for each option to determine what you need, read about 2.2.*x* NAT configuration from another source, or upgrade to a 2.4.*x* kernel to implement NAT on such a system. The 2.4.*x* kernel offers many improvements over older ones, including extended support for USB devices. I therefore favor this option if at all possible.

Compiling and using a Linux kernel are beyond the scope of this book, so if you need help in these matters, consult an appropriate source of documentation, such as an introductory Linux book or the Kernel HOWTO (`http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html`).

Enabling NAT Features

Even with an appropriate kernel, Linux doesn't perform NAT functions automatically. Part of the reason is that Linux doesn't function as a router by default; the rest is that a NAT-enabled kernel must have its NAT features activated by `iptables` or a similar tool. To enable basic NAT functionality, follow these steps:

1. Install and configure two network cards. Appendix C covers Linux network card configuration. Use whatever method your ISP requires for IP address assignment on the external card, and use a static IP address on the card for your internal network.
2. If you compiled the NAT features as modules, load the appropriate module files. This can usually be accomplished by typing **`modprobe iptable_nat`**. It might be done automatically when you try to use NAT features, as well.
3. To enable NAT in the kernel, you type a single command, providing the system with the name of your *external* (broadband) interface. If you specify the wrong interface, NAT will be set up "backward," as if your internal network were the Internet. The appropriate command is as follows, assuming `eth0` is the external interface:

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

NOTE

At this point if you receive an error message that says you've specified an invalid target, your kernel almost certainly lacks one or more necessary features. If you're using a precompiled kernel, you may need to compile one yourself. If you've compiled your own kernel, review the modules listed in the section Linux Kernel Options for NAT to be sure you included them all.

4. Turn on IP forwarding (on which NAT relies) by typing the following command:

```
# echo "1" > /proc/sys/net/ipv4/ip_forward
```

These commands can be entered into an appropriate system start-up script, such as `/etc/rc.d/rc.local`, if you want the system always to function as a NAT router. Be sure these commands execute *after* any you might need to initiate your basic network connections, though. This includes any PPPoE start-up scripts if your ISP uses PPPoE.

Enabling DHCP Features in Linux

Linux's basic NAT tools do not automatically enable a DHCP server or other useful features for a NAT router, as they exist in Windows ICS or (optionally) many other packages. You can, of course, run such tools separately in Linux, if you like. In many cases, it's simpler to configure your local systems manually to use static IP addresses. If you've got many systems,

though, you may prefer to use DHCP. In Linux, the DHCP server usually comes in a package called `dhcpd` and is configured through `/etc/dhcpd.conf`. A DHCP configuration file might resemble the following:

```
default-lease-time 6000;
max-lease-time 10000;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
option domain-name-servers 172.19.17.100;
option domain-name "threeroomco.com";
subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.40 192.168.1.50;
}
```

This example file dishes out addresses between 192.168.1.40 and 192.168.1.50, along with information on the gateway system (`option routers`), the DNS server (`option domain-name-server`), the domain name (`option domain-name`), and so on. The lease times are defined in seconds. This file includes one `subnet` specification, but some versions of `dhcpd` require a second subnet specification for the external network, even if `dhcpd` doesn't serve that network. If you find that `dhcpd` doesn't start properly, try adding an empty `subnet` specification for your external network. You may also need to add the name of the internal network's interface to the command that starts `dhcpd` (probably located in a script in `/etc/rc.d/init.d` or someplace similar).

If you want to assign the same IP address to a computer every time it boots, you can do so by adding lines similar to the following to the `/etc/dhcpd.conf` file:

```
host louiswu {
   hardware ethernet 00:e0:98:71:60:c1;
   fixed-address 192.168.1.5;
}
```

The `hardware ethernet` line specifies the MAC address of the client's network card, and the `fixed-address` line specifies the IP address you want to assign to that system. Be sure you specify an address that's outside the range you specified on the `range` line in the `subnet` specification. (You can discover a computer's MAC address with the `arp` command, as in **`arp -a 192.168.1.40`** to display the MAC address of 192.168.1.40. If the computer doesn't yet have an IP address, you can temporarily configure it with a dynamic IP address via DHCP to obtain this information.)

WARNING

It's possible to misconfigure Linux's DHCP server to provide IP addresses on the external interface as well as the internal one. If your ISP uses DHCP and doesn't block DHCP requests to your system, this can cause a great deal of confusion, because your server might provide incorrect IP addresses to other subscribers' computers. For this reason, it's often best to run the DHCP server on a machine within your local network rather than on the broadband router itself.

Enabling Port Forwarding

There are several different ways to enable port forwarding on a Linux system that provides NAT functions. One is to do it with `iptables`. To do so, you can type a command similar to the following:

```
# iptables -t nat -A PREROUTING -p tcp -i eth0 –dport 23 -j DNAT –to 192.168.0.2:23
```

Important features of this command include the following:

- The command manipulates the NAT table (`-t nat`).
- The `-A PREROUTING` parameter specifies when the changes to packets are to be made—prior to routing proper. The basic NAT features operate postrouting, but port forwarding happens prior to routing.
- The command forwards TCP ports (`-p tcp`).
- The rule applies to packets directed to the system's external network interface (`-i eth0`) on port 23 (`–dport 23`).
- The `-j DNAT` parameter tells the system that it's performing NAT on the destination (DNAT) rather than the source (SNAT) address.
- The final parameter, `–to 192.168.0.2:23`, specifies that the packets should be directed to port 23 on 192.168.0.2.

The final result of these rules is that packets directed to port 23 (used for Telnet) on the external interface are rewritten so that they're forwarded to an internal system. You can enter several such commands—as many as you need to forward any ports that handle servers you want to run internally.

As with the basic NAT configuration, you can create entries in a start-up script such as `/etc/rc.d/rc.local` to run these commands whenever your system starts. You can then leave the configuration alone.

CONFIGURING A HARDWARE ROUTER FOR NAT

Hardware broadband routers are typically designed with the assumption that they'll be used as NAT routers. They therefore come configured to use NAT by default, which makes their setup fairly simple. Hardware routers also typically implement features that can help simplify your local network configuration, such as DHCP servers that automatically forward the IP addresses of your ISP's DNS servers to the machines on your local network.

Beyond basic and default features, most NAT routers support more advanced options, including port forwarding and firewall rules. These may be trickier to configure than the router's basic functions. If you want to run a server on a broadband account that's protected by a broadband router, you *must* enable port-forwarding features. Except for their built-in servers, which are typically designed for configuring the device and providing basic services to internal systems, hardware routers don't run servers.

In short, broadband routers are essentially computers that run the equivalent of ICS, IPNetRouter, or similar software and nothing else. Most broadband routers aren't as flexible as at least some packages for conventional OSs are. For instance, broadband routers often impose limits on the number of rules you can define for handling port forwarding. They're usually adequate for handling internal networks of a dozen or so systems, though, and sometimes more than that. (You may need to add separate hubs or switches to accommodate that many computers, though.)

ACCESSING THE ROUTER'S UTILITIES

Unfortunately, hardware routers are far from standardized in their methods of configuration. As described in Chapter 15, these devices may use proprietary Windows software, Web browsers, or text-based (Telnet or serial port) configuration tools. The exact placement and naming of options varies from one device to another. This section presents the configuration of a ZyXEL P314 (which is nearly identical to the NetGear RT314), with firmware revision 3.21, as an example. Other routers have similar options, but they may be accessed in other ways or be called other things.

It's possible to configure the ZyXEL for basic operations using a Web-based interface. In fact, for basic configuration, this can be a good way to do it because the Web-based tool provides access only to the basic options and directs you through the process, asking for all the information needed to get your entire network connected through NAT (which ZyXEL refers to as *Single User Access*, or *SUA*). If you prefer, you can use the text-based

tools, as described in Chapter 15. To configure the router using its Web interface, follow these steps:

1. Connect the router's external interface to your broadband modem, and plug at least one local computer into one of the router's internal interfaces. Power on the router, the modem, and the computer.
2. If the computer is configured with a static IP address or with PPPoE, change its configuration to use DHCP, as described in Chapter 6 or Appendixes A–C, as appropriate. Alternatively, give the computer a static IP address in the range 192.168.0.2–192.168.0.254 (192.168.1.2–192.168.1.254 for the NetGear model; consult your documentation for other devices).
3. Reboot your computer, if necessary, so it obtains an IP address from the router or uses the new static address you've assigned it.
4. Start a Web browser and enter `http://192.168.0.1` as a URL. (Use `http://192.168.1.1` for the NetGear model.) You'll be asked for a username and password. The default username is `admin`, and the password is `1234`. When you've entered these, you'll see the main configuration screen.

WARNING

It's a good idea to change the password on any hardware router you obtain. Although these devices typically ship configured to reject logins from their broadband connections, it's possible to change this configuration. If you do so by accident but fail to change your password, anybody who knows (or can figure out) that you're using a particular model router can gain access to it and reprogram it to give access to your local network. This breach won't automatically overcome security on your local systems, but if any of your local systems are vulnerable, the intruder may be able to break into them in this way. The ZyXEL's password can be changed from the System Password menu (#23) using the text-based configuration tool.

5. Click the Wizard Setup link to the left of the page. The result is the first data-entry page, shown in Figure 16.10. (Note that the router used for these screen shots has been configured with a nondefault IP address.)
6. Enter your system name and ISP's domain name in the fields provided. This information isn't usually vitally important, but in a few cases it is. If it's important, your ISP will provide you with the appropriate information. When you click Next, the display changes to show the connection parameters (see Figure 16.11).
7. Select the Encapsulation method your connection uses—Ethernet or PPPoE. If you choose Ethernet, you'll have a choice of Service Type—

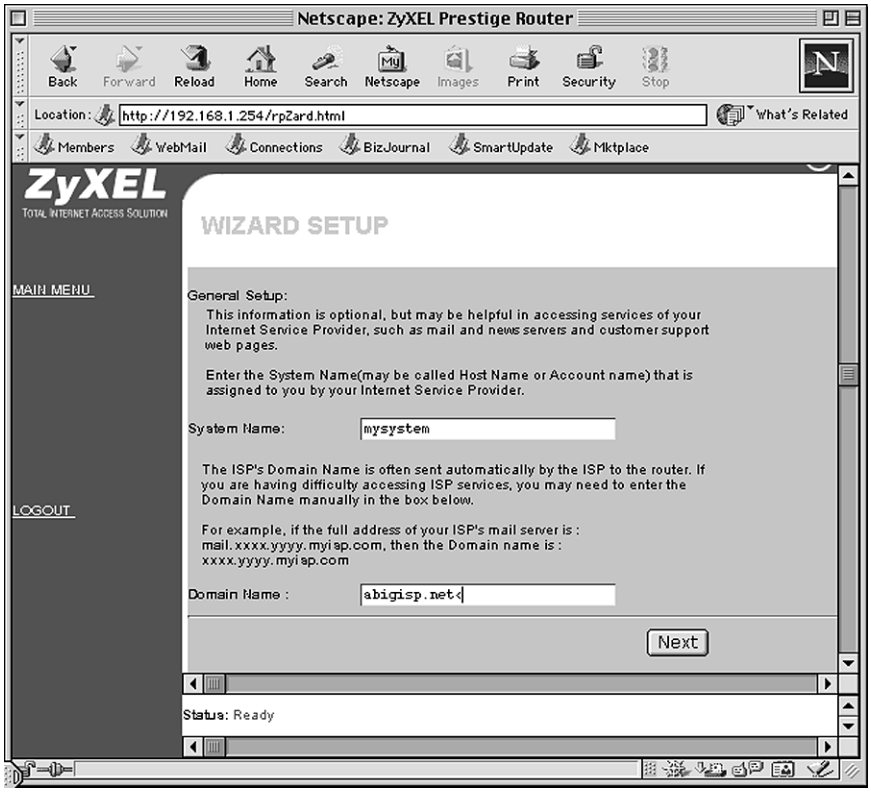

Figure 16.10 The ZyXEL Wizard Setup tool prompts for specific information needed to configure the router.

Standard or two options used on some older Road Runner cable systems. In some cases, you'll have to enter your username, password, and the like, as shown in Figure 16.11. When you've entered all the information, click Next (a button that's not visible in Figure 16.11; you may need to scroll down).

8. The next screen lets you enter your broadband IP address (ZyXEL refers to this as a *WAN IP address*) or tell the system to get it automatically (via PPPoE or DHCP, as appropriate). You can also set the MAC address on the broadband interface. The MAC address is built into the hardware, but many broadband routers can mimic the MAC address of a system on the internal network. (You need only enter the target system's IP address; the router obtains the MAC address automatically from that.) This option is useful on some ISPs that use DHCP because it allows you to add a router after the broadband connection has already been config-

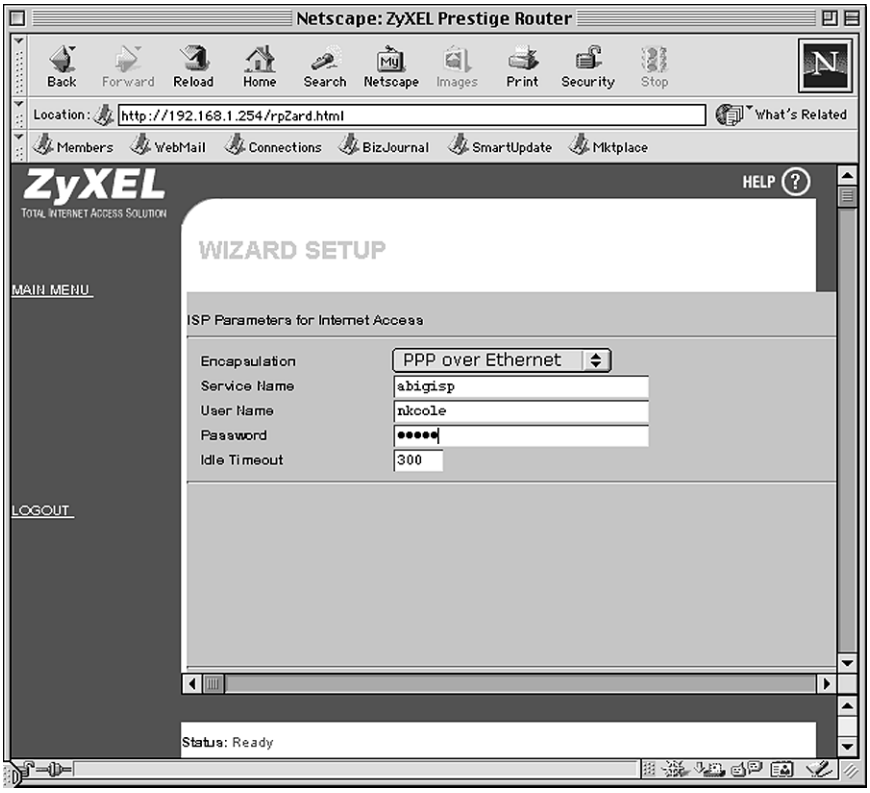

Figure 16.11 The connection parameters change when you select different Encapsulation methods.

ured for a given computer. When you've entered this information, click Finish, and the process is over.

At this point, you should be able to access the Internet from any computers attached to the router, either directly or through a hub or switch. If you're using DHCP on your local network, though, your local computers probably won't receive correct DNS server addresses until you restart their networking services or reboot them. Without DNS, you won't be able to specify external computers by name, only by number.

NOTE

If you don't want to use DHCP at all on your internal network, you can simply ignore the router's DHCP server. You'll need to provide your internal computers with the router's IP address as the gateway and provide appropriate IP

addresses for DNS servers as well. If you want to run a DHCP server on another system (say, so you can modify the information it provides to point to your own DNS server), you should disable the hardware router's DHCP server. In the case of the ZyXEL/NetGear twins, this can be done through the text-based (Telnet or serial port) interfaces.

Other broadband routers use their own configuration tools, which may look different or provide somewhat different options. All should ask for fundamentally the same information described previously, however.

CONFIGURING PORT FORWARDING

Although the ZyXEL P314 can be configured for basic operations via a Web-based interface, this device relegates more sophisticated configuration to its text-based tools. You can access these by using any Telnet client. When you enter the router's IP address, the device will ask for a password. Enter it, and you'll be greeted by the main menu (see Figure 16.12).

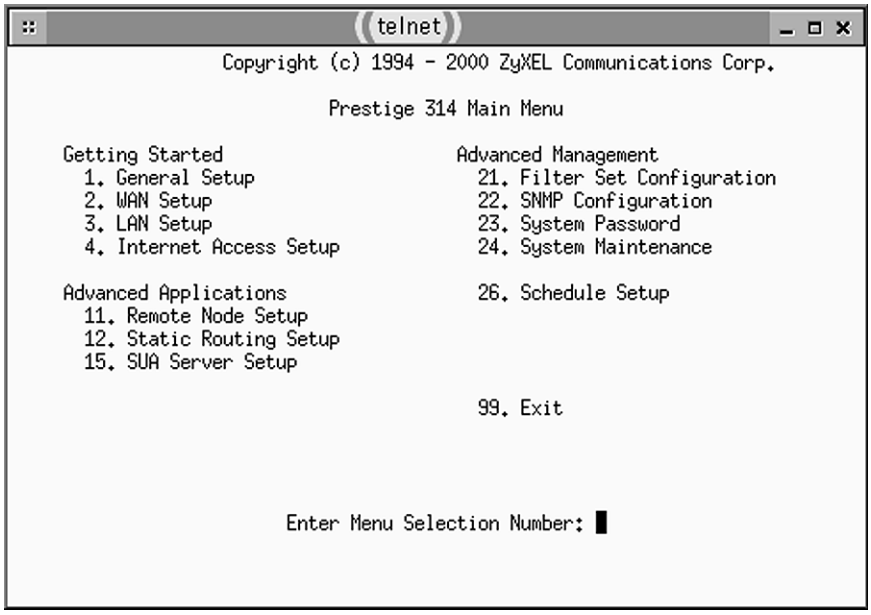

Figure 16.12 The ZyXEL P314's main menu groups router configuration functions into logically related groups.

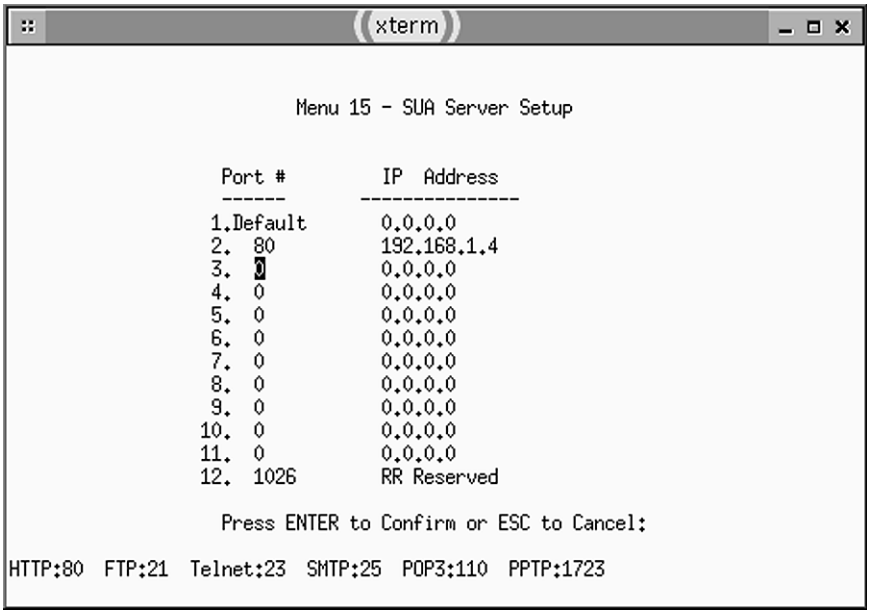

Figure 16.13 The SUA Server Setup menu lets you forward several ports to internal computers.

To configure port forwarding, follow these steps:

1. Type **15** at the main menu. This brings up the SUA Server Setup menu, shown in Figure 16.13 (which shows the system already configured to forward one port).
2. Use your computer's arrow keys to position the cursor on the first `0` in the Port column; then type the port number you want to forward. (Consult Table 16.1 for a list of common port numbers.)
3. When you press the Enter key, the cursor jumps to the IP Address column. Type the IP address of a computer on your local network, followed by the Enter key.
4. Repeat steps 2 and 3 for as many ports as you want to forward to an internal computer.
5. Position the cursor on the last line and press the Enter key to enter your changes into the router's memory. You'll be returned to the main menu, and the router will begin forwarding the ports you specified.

You can test this configuration by using a computer outside of your local network (say, by using a backup telephone dial-up PPP account) to access the port number you specified on your broadband IP address. (Part III covers running servers on a broadband connection, including obtaining a domain name so your users can easily locate your computer.)

SUMMARY

NAT is probably the most popular method of sharing a broadband connection. This tool is very useful as a means of stretching a limited number of IP addresses—both for the Internet community as a whole and for individuals and small organizations that have just one IP address on a broadband account but several computers. NAT can be implemented either in NAT router software for a conventional OS or in a hardware device designed with NAT in mind. Either way, it's usually possible to configure NAT to pass traffic directed to specific ports on to computers on the internal network, thus allowing you to run servers internally for external use. Many NAT packages also include tools that can help you configure an internal network, such as DHCP servers. Those NAT programs that don't include these features can be supplemented by extra servers for these functions if you so desire.