problems with other ICMP packets, to avoid flooding the network with error messages about error messages)
– this probe generates another "* * *" output

TTL = 4: this behaves like TTL = 3, but gets as far as Rupert, who drops it silently. Again we get a "* * *" output

TTL = 5: the TTL is 3 when the packet reaches Bob, who replies with an ICMP packet with TTL = 3. This *does* reach Alice finally, so we get the last output line, giving the response times for Bob

Out of a total of five lines, we have two containing three "*"s; the trace is almost twice as long as it ought to be.

traceroute prints a "!" after a response that had a TTL = 0 or 1, so if you see three "!"s in the last line of the output, it's almost certain that the destination machine is behaving as described above

- as we explained in Module 3.9, Windows' **tracert** sends ICMP Echo Request packets as probes whereas Linux's **traceroute** sends UDP packets. If the probes have to pass through a firewall to the final destination, they may be blocked, so your trace will be incomplete. As firewalls may have different configuration settings for different types of traffic, for some destinations you may find that a Windows **tracert** works fine but the Linux **traceroute** is incomplete, or vice versa.

Troubleshooting

In Module 5.15 we suggested tracing packets hop by hop. If you're not on your own site you probably won't be able to install tools like **tcpdump** or **windump** on other people's machines. If the machines are running Windows, Microsoft's Network Monitor (Chapter 18) is a useful alternative.

Terminology: "host," "router," "machine"

We use the following common convention when talking about routing. "Host" means a computer that is not acting as a router, e.g. a desktop workstation or a server. "Router" means a computer that is acting as a router and forwards packets between networks. (It may be acting as a server or a desktop PC too, although that's unlikely – and we're only concerned with its routing function.) When we say "machine" we mean either a host or a router – typically when discussing something that applies to any TCP/IP computer, no matter what its role is in the network.

Tools

Our Web site contains several tools for interpreting route table listings, taking account of all the flags etc.:

- http://www.uit.co.uk/resources

Appendices

Appendix 6: making network settings permanent – Debian Linux
Appendix 8: Windows-NT **route** command manpage
Appendix 9: example of working network diagram
Web Appendix 5: **route** command error messages

6 Routing in practice

Introduction

This chapter explains why you might want to divide your internal network into sub-nets, and how to do it. Reasons for sub-netting include physical limitations on individual network size, performance considerations, geographical separation of parts of your organization's network, ease of administration, and security (e.g. isolating confidential machines from the rest of your network), and for testing. Sub-netting is essential for medium- to large-sized networks.

6.1 Hosts using multiple routers

Up to now all our hosts have only used the services of a single router, and that was usually their default gateway. In the next two modules we're going to configure a more complex network that is representative of medium-sized installations (Figure 6.1). It uses multiple Ethernet segments and individual PCs are configured to use multiple routers.

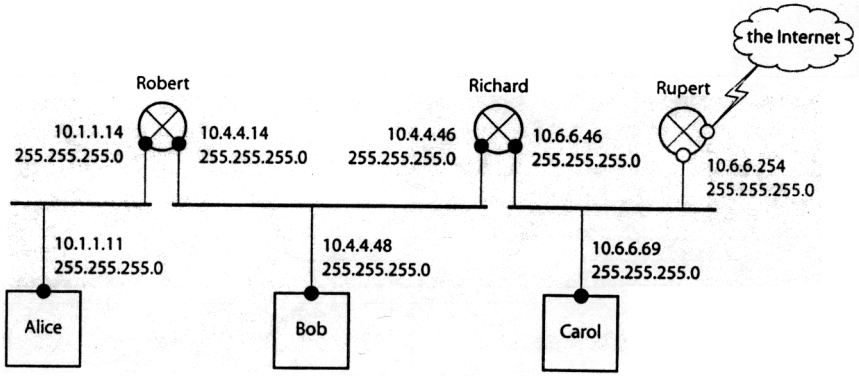

Figure 6.1 Network typical of medium-sized installations

We've shown the network as being connected to the Internet; this is only possible using the private (non-routable) IP addresses we've shown in the diagram if the main Internet router (Rupert) performs network address translation (NAT, Module 23.2). NAT is very common for larger sites nowadays, but if you're not going to use NAT, change the IP addresses in the diagram to the real numbers you're using on your network.

We'll configure the network in two ways, first "by hand," and second, taking advantage of ICMP to cut down on the configuration details we have to enter. From this exercise you'll see how to plan your router configuration, and how routes cascade from the deepest part of the internal network towards the main Internet router. Configuring a network like this (whether in real-life or as an exercise) is painstaking work: there's lots of detail and if you're not careful it's very easy to get an IP number wrong so the whole thing doesn't work.

To build this network you don't need an Internet connection and you don't need the Rupert machine, because almost all the configuration work is on the internal routers. However, we've shown Rupert because networks like this are not symmetrical – the location of the Internet gateway is very significant.

Note:

- we've used the private address range 10.x.x.x
- by using "whole class C" (256-address) ranges for the different sub-networks, the netmasks are very simple. You don't have to do any binary arithmetic to calculate what the ranges are – you can tell at a glance which network an IP address belongs to.

Instead of writing down a table of the configuration for each machine, we show it directly on the diagram (which is how we do it in real life).

In the next module we use this notation (Figure 6.2):

- on a machine, a route is shown as

  `R(10.6.6.0, 255.255.255.0 -> 10.4.4.46=Richard)`

  which means a route to the network/range 10.6.6.0, netmask 255.255.255.0 uses 10.4.4.46 (the router Richard) as gateway
- a large curved arrow, usually marked dg, points from a machine to its default gateway router.

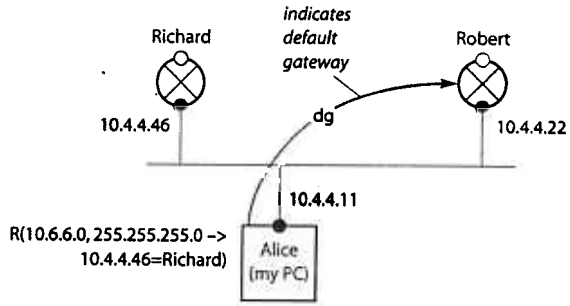

Figure 6.2 Network notation

> **Tip:** when we use diagrams like this on-site for building a real network, we often make a photocopy of it. Then, as we configure a parameter, we scratch it off using a fluorescent highlighter, so it's easy to see what we have and haven't done.

Configuring a machine to use multiple routers

Use the route add command to create a route to the routers, just as we did in the previous chapter. For example, if Bob is a Linux box, to tell it to use Robert and Richard to get to the adjacent networks (but nothing else) you might use:

```
route add -net 10.1.1.0 netmask 255.255.255.0 gw 10.4.4.14
route add -net 10.6.6.0 netmask 255.255.255.0 gw 10.4.4.46
```

whereas on Windows it would be:

```
route add 10.1.1.0 mask 255.255.255.0 10.4.4.14
route add 10.6.6.0 mask 255.255.255.0 10.4.4.46
```

6.2 Lab – building a network with multiple routers

Method 1 – manual configuration

Figure 6.3 shows what we need to do.

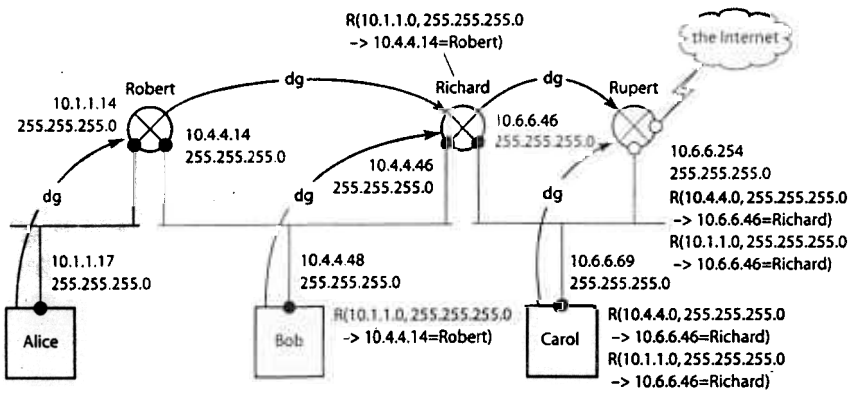

Figure 6.3 Manual configuration

To double-check we've got it right, let's see how Alice, Bob, and Carol reach the three internal networks and the external Internet:

	10.1.1.0	10.4.4.0	10.6.6.0	Internet
Alice	connected directly	> Robert	> Robert > Richard	> Robert > Richard > Rupert
Bob	> Robert	connected directly	> Richard	> Richard > Rupert
Carol	> Richard > Robert	> Richard	connected directly	> Rupert

Let's look at one row of the table in more detail, Bob for example.

- **10.1.1.0** Bob has an explicit route, telling him to forward via Robert. Robert's left hand interface is connected directly to this range, so Robert can forward the packets on the local wire
- **10.4.4.0** Bob is connected directly to this range so he can access it on the local wire
- **10.6.6.0** Bob has no special route for this, so he forwards via his default gateway, Richard. Richard's right-hand interface is connected directly to this range so Richard forwards on the local wire
- **Internet** Bob has no special route for this, so he forwards via his default gateway, Richard. Richard's doesn't have any special route either, so Richard forwards to Rupert, who sends it on to the Internet.

Method 2 – using ICMP redirects

Now, more realistically, let's use the flexibility ICMP redirects give us. This allows us to configure special routes only on the routers; host machines only need their single default gateway; ICMP redirects will take care of the rest (Figure 6.4).

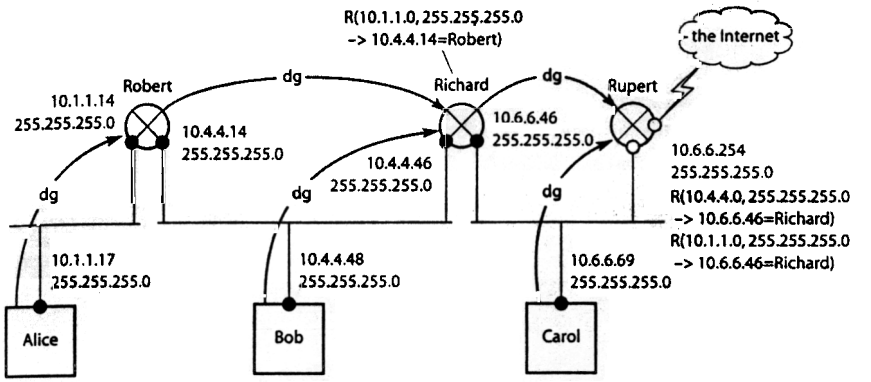

Figure 6.4 Using ICMP redirects to simplify host machine configuration

(Even without ICMP redirects we could omit the specific routes on Alice, Bob, and Carol and the network would still work. However, every packet travelling to a network on its left would generate twice as much traffic because it would go to its default gateway first and then back to the router on its left.)