Instructions for setting up layer 3 routing

The first thing to keep in mind is that layer 3 routing is a completely different thing from layer 7 Tor overlay routing. Layer 3 in modern network stacks is the IP protocol, which is described in Chapter 4 of your book.

Routers route packets between more than one subnet, to have more than one subnet in VirtualBox you have to either define more than one internal network (e.g., intnet0, intnet1, intnet2, …) or you have to go into VirtualBox’s “Preferences” and create multiple host-only adapters (vboxnet0, vboxnet1, vboxnet2, …). I did the latter, see screenshots 18 through 22.

For screenshots 0 through 17 I want Adapter 1 to be eth0, Adapter 2 to be eth1, and Adapter 3 (if present) to be eth2. It usually works out this way, but it’s not guaranteed that Ubuntu will discover the emulated hardware for the NIC adapters in the same order that you specified them, so sometimes you have to write down the MAC addresses and rename eth0, eth1, and eth2 in Ubuntu’s /etc/udev/rules.d/70-persistent-net.rules file. For all of my virtual machines to be able to get on the Internet I gave them all an Adapter 1 (which will later be mapped to eth0 in the software part) that uses VirtualBox’s built-in Network Address Translation to get out to the world of the routable Internet using the host’s IP address. For the Client and Server, Adapter 2 is their local network, while Router1, Router2, and Router3 need both an Adapter 2 and an Adapter 3 to span multiple subnets. Technically every Linux machine makes routing decisions, but for those routing decisions to be non-trivial enough to call the machine a router the machine should be on more than one subnet and forward packets from one to the other and vice versa.

See subnets.png for the structure of my network.

First, let’s look at Router1. Screenshot 23 shows how the NIC adapters in hardware get named by the software to eth0, eth1, and eth2, you can change the /etc/udev/rules.d/70-persistent-net.rules file where this is defined but there’s no need to change it if Ubuntu discovers the hardware in the order you wanted.

Screenshot 24 is the /etc/network/interfaces file for Router 1. Note that the spelled-out word “gateway” doesn’t appear anywhere in this file, only the abbreviated “gw” in the explicit routing rules. In the /etc/network/interfaces file, “gateway” defines a default gateway, which is where packets should go when there’s no rules already for them. You want the default gateway for your virtual machines to be the Internet through VirtualBox’s NAT interface on eth0, which VirtualBox will set up on eth0 via DHCP. So you don’t want to define a default gateway anywhere for eth0, eth1, or eth2, your eth0 gateway will become the default gateway through DHCP.

Router1 has explicit routing rules for the 192.168.58.0/24 and 192.168.59.0/24 subnets (note that /24 implies a netmask of 255.255.255.0 on a network, since 255.255.255.0 in binary has 24 ones and 32-
24=8 zeroes). Router1 doesn't need routing rules for 192.168.56.0/24 or for 192.168.57.0/24, because the fact that it's on both of these subnets means it knows how to talk to machines there without needing to go through a gateway router. It can talk to machines on its own subnets directly using their ARP addresses.

Let's dissect one of the routing rules:

**up route add -net 192.168.58.0 netmask 255.255.255.0 gw 192.168.57.1**

This is telling the Linux kernel that if it's going to forward a packet to the network 192.168.58.0 with netmask 255.255.255.0 (in other words, 192.168.58.0/24) then it should use the gateway 192.168.57.1. We don't define an interface, since the Linux kernel will be able to infer the 192.168.57.1 is on eth2. The Linux kernel will know that what it's supposed to do is forward the packet onto eth2 with the source and destination IP addresses unchanged, using its own source MAC address and the destination MAC address for 192.168.57.1. The kernel will use an ARP request to find out 192.168.57.1's MAC address if it doesn't already have it in the ARP cache.

Screenshot 25 just shows what IP addresses the interfaces end up with, which is pretty much what we defined except that eth0 asks for an IP address, subnet mask, default gateway, DNS servers, etc. through DHCP and VirtualBox replies with enough info for the virtual machine to have a route to the Internet.

The main thing to notice in Screenshot 26 is that I've uncommented a line in `/etc/sysctl.conf` to tell the Linux kernel that it's okay to forward IPv4 packets. You have to reboot so that this setting gets loaded into the kernel in `/proc/sys/net/ipv4/ip_forward` (actually you can just write a 1 there as root but I think it's cleaner to reboot).

The results of "route -n" are interesting. This is showing us the routing table the kernel will use to make routing decisions. The first rule just says that eth0 is on the 10.0.2.0/24 subnet, so any of the 256 IP addresses in this range should be written to eth0 directly (gateway 0.0.0.0 means use the destination MAC address you're trying to reach, not the destination MAC address of a gateway router). The second and third rule are the ones we specified, that 192.168.58.0/24 and 192.168.59/0/24 can be reached through 192.168.57.1, which the kernel has figured out is on eth2 even though we didn't specify that in our rule. The fifth and sixth rules are similar to the first but are for the local subnets of eth1 and eth2. Finally, the last rule says that if an IP address did not match the ranges of any of the other rules (probably because it's somewhere out on the Internet and not on our local network), then the kernel should use 10.0.2.2 on eth0 as the gateway router, which happens to be VirtualBox's NAT.

Router 3 is similar to router 1. Router 2 is pretty much the same except that it has rules going in both directions. So, when should you add an explicit routing rule? On a given machine, *you need a routing rule for any subnets that that machine is not on*. For the routable Internet (pretty much anything except 10.0.0.0/8 and 192.168.0.0/16), DHCP on eth0 will give you a default route, but for any local networks you create if the virtual machine is not on that subnet then you need to add the rule yourself.
The Client and Server need explicit routing rules, too, but for local networks in the 192.168.0.0/16 range they always use the same gateway. Since the Client and Server never need to forward packets for other machines, I left the line about IPv4 forwarding commented out in `/etc/sysctl.conf`. 