to skip ahead to **Designing the Physical Network** on **Page 51**. We will now review the basics of Internet networking.

Introduction

Venice, Italy is a fantastic city to get lost in. The roads are mere foot paths that cross water in hundreds of places, and never go in a simple straight line. Postal carriers in Venice are some of the most highly trained in the world, specializing in delivery to only one or two of the six *sestieri* (districts) of Venice. This is necessary due to the intricate layout of that ancient city. Many people find that knowing the location of the water and the sun is far more useful than trying to find a street name on a map.

Figure 3.1: Another kind of network mask.

Imagine a tourist who happens to find papier-mâché mask as a souvenir, and wants to have it shipped from the studio in S. Polo, Venezia to an office in Seattle, USA. This may sound like an ordinary (or even trivial) task, but let's look at what actually happens.

The artist first packs the mask into a shipping box and addresses it to the office in Seattle, USA. They then hand this off to a postal employee, who attaches some official forms and sends it to a central package processing hub for international destinations. After several days, the package clears Italian customs and finds its way onto a transatlantic flight, arriving at a central import processing location in the U.S. Once it clears through U.S. customs, the package is sent to the regional distribution point for the northwest U.S., then on to the Seattle postal processing center. The package eventually makes its way onto a delivery van which has a route that brings it to the proper address, on the proper street, in the proper neighborhood. A clerk at the office

Layer	Name	Description
2	Data Link	Whenever two or more nodes share the same physical medium (for example, several computers plugged into a hub, or a room full of wireless devices all using the same radio channel) they use the ***Data Link Layer*** to communicate. Common examples of data link protocols are Ethernet, Token Ring, ATM, and the wireless networking protocols (802.11a/b/g). Communication on this layer is said to be link-local, since all nodes connected at this layer communicate with each other directly. This layer is sometimes known as the ***Media Access Control*** (***MAC***) layer. On networks modeled after Ethernet, nodes are referred to by their ***MAC address.*** This is a unique 48 bit number assigned to every networking device when it is manufactured.
1	Physical	The ***Physical Layer*** is the lowest layer in the OSI model, and refers to the actual physical medium over which communications take place. This can be a copper CAT5 cable, a fiber optic bundle, radio waves, or just about any other medium capable of transmitting signals. Cut wires, broken fiber, and RF interference are all physical layer problems.

The layers in this model are numbered one through seven, with seven at the top. This is meant to reinforce the idea that each layer builds upon, and depends upon, the layers below. Imagine the OSI model as a building, with the foundation at layer one, the next layers as successive floors, and the roof at layer seven. If you remove any single layer, the building will not stand. Similarly, if the fourth floor is on fire, then nobody can pass through it in either direction.

The first three layers (Physical, Data Link, and Network) all happen "on the network." That is, activity at these layers is determined by the configuration of cables, switches, routers, and similar devices. A network switch can only distribute packets by using MAC addresses, so it need only implement layers one and two. A simple router can route packets using only their IP addresses, so it need implement only layers one through three. A web server or a laptop computer runs applications, so it must implement all seven layers. Some advanced routers may implement layer four and above, to allow them to make decisions based on the higher-level information content in a packet, such as the name of a website, or the attachments of an email.

The OSI model is internationally recognized, and is widely regarded as the complete and definitive network model. It provides a framework for manufac-

hosts can reach each other directly (first using ARP to resolve the IP address into a MAC address, and then sending packets to that MAC address).

Now we will add host G. This host has two network cards, with one plugged into each network. The first network card uses the IP address 192.168.1.4, and the other uses 192.168.2.4. Host G is now link-local to both networks, and can route packets between them.

But what if hosts A, B, and C want to reach hosts D, E, and F? They will need to add a route to the other network via host G. For example, hosts A-C would add a route via 192.168.1.4. In Linux, this can be accomplished with the following command:

```
# ip route add 192.168.2.0/24 via 192.168.1.4
```

...and hosts D-F would add the following:

```
# ip route add 192.168.1.0/24 via 192.168.2.4
```

The result is shown in **Figure 3.8**. Notice that the route is added via the IP address on host G that is link-local to the respective network. Host A could not add a route via 192.168.2.4, even though it is the same physical machine as 192.168.1.4 (host G), since that IP is not link-local.

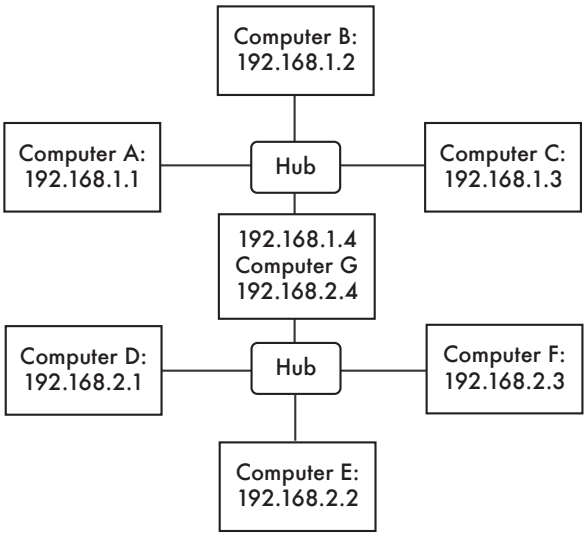


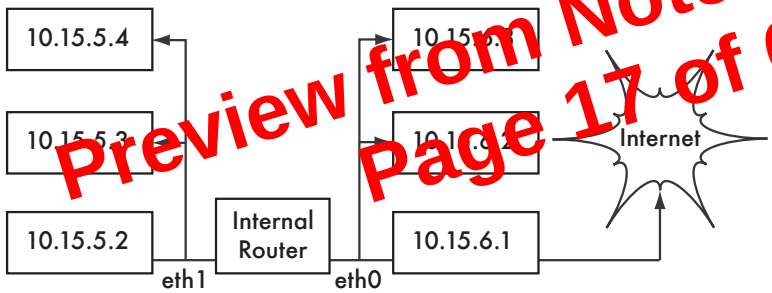
Figure 3.8: Host G acts as a router between the two networks.

A route tells the OS that the desired network doesn't lie on the immediate link-local network, and it must ***forward*** the traffic through the specified router. If host A wants to send a packet to host F, it would first send it to host G. Host G would then look up host F in its routing table, and see that it has a direct

connection to host F's network. Finally, host G would resolve the hardware (MAC) address of host F and forward the packet to it.

This is a very simple routing example, where the destination is only a single ***hop*** away from the source. As networks get more complex, many hops may need to be traversed to reach the ultimate destination. Since it isn't practical for every machine on the Internet to know the route to every other, we make use of a routing entry known as the ***default route*** (also known as the ***default gateway***). When a router receives a packet destined for a network for which it has no explicit route, the packet is forwarded to its default gateway.

The default gateway is typically the best route out of your network, usually in the direction of your ISP. An example of a router that uses a default gateway is shown in **Figure 3.9**.




```
Routing table for internal router:

Destination     Gateway     Genmask         Flags  Metric  Iface
10.15.5.0       *           255.255.255.0   U      0       eth1
10.15.6.0       *           255.255.255.0   U      0       eth0
default         10.15.6.1   0.0.0.0         UG     0       eth0
```

Figure 3.9: When no explicit route exists to a particular destination, a host uses the default gateway entry in its routing table.

Routes can be updated manually, or can dynamically react to network outages and other events. Some examples of popular dynamic routing protocols are RIP, OSPF, BGP, and OLSR. Configuring dynamic routing is beyond the scope of this book, but for further reading on the subject, see the resources in **Appendix A**.

Network Address Translation (NAT)

In order to reach hosts on the Internet, RFC1918 addresses must be converted to global, publicly routable IP addresses. This is achieved using a technique known as ***Network Address Translation***, or ***NAT***. A NAT device is a router that manipulates the addresses of packets instead of simply forwarding them. On a NAT router, the Internet connection uses one (or more) glob-

Ethernet

Ethernet is the name of the most popular standard for connecting together computers on a ***Local Area Network*** (***LAN***). It is sometimes used to connect individual computers to the Internet, via a router, ADSL modem, or wireless device. However, if you connect a single computer to the Internet, you may not use Ethernet at all. The name comes from the physical concept of the ether, the medium which was once supposed to carry light waves through free space. The official standard is called IEEE 802.3.

The most common Ethernet standard is called 100baseT. This defines a data rate of 100 megabits per second, running over twisted pair wires, with modular RJ-45 connectors on the end. The network topology is a star, with switches or hubs at the center of each star, and end nodes (devices and additional switches) at the edges.

MAC addresses

Every device connected to an Ethernet network has a unique MAC address, assigned by the manufacturer of the network card. Its function is like that of an IP address, since it serves as a unique identifier that enables devices to talk to each other. However, the scope of a MAC address is limited to a broadcast domain, which is defined as all the computers connected together by wires, hubs, switches, and bridges, but not crossing routers or Internet gateways. MAC addresses are never used directly on the Internet, and are not transmitted across routers.

Hubs

Ethernet ***hubs*** connect multiple twisted-pair Ethernet devices together. They work at the physical layer (the lowest or first layer). They repeat the signals received by each port out to all of the other ports. Hubs can therefore be considered to be simple repeaters. Due to this design, only one port can successfully transmit at a time. If two devices transmit at the same time, they corrupt each other's transmissions, and both must back off and retransmit their packets later. This is known as a ***collision***, and each host remains responsible for detecting collisions during transmission, and retransmitting its own packets when needed.

When problems such as excessive collisions are detected on a port, some hubs can disconnect (***partition***) that port for a while to limit its impact on the rest of the network. While a port is partitioned, devices attached to it cannot communicate with the rest of the network. Hub-based networks are generally more robust than coaxial Ethernet (also known as 10base2 or ThinNet), where misbehaving devices can disable the entire segment. But hubs are limited in their usefulness, since they can easily become points of congestion on busy networks.

less clients, handling channel contention, repeating packets, etc.) Wireless cards in master mode can only communicate with cards that are associated with it in managed mode.

2. ***Managed mode*** is sometimes also referred to as ***client*** mode. Wireless cards in managed mode will join a network created by a master, and will automatically change their channel to match it. They then present any necessary credentials to the master, and if those credentials are accepted, they are said to be ***associated*** with the master. Managed mode cards do not communicate with each other directly, and will only communicate with an associated master.

3. ***Ad-hoc mode*** creates a multipoint-to-multipoint network where there is no single master node or AP. In ad-hoc mode, each wireless card communicates directly with its neighbors. Nodes must be in range of each other to communicate, and must agree on a network name and channel.

4. ***Monitor mode*** is used by some tools (such as **Kismet**, see **Chapter 6**) to passively listen to all radio traffic on a given channel. When in monitor mode, wireless cards transmit no data. This is useful for analyzing problems on a wireless link or observing spectrum usage in the local area. Monitor mode is not used for normal communications.

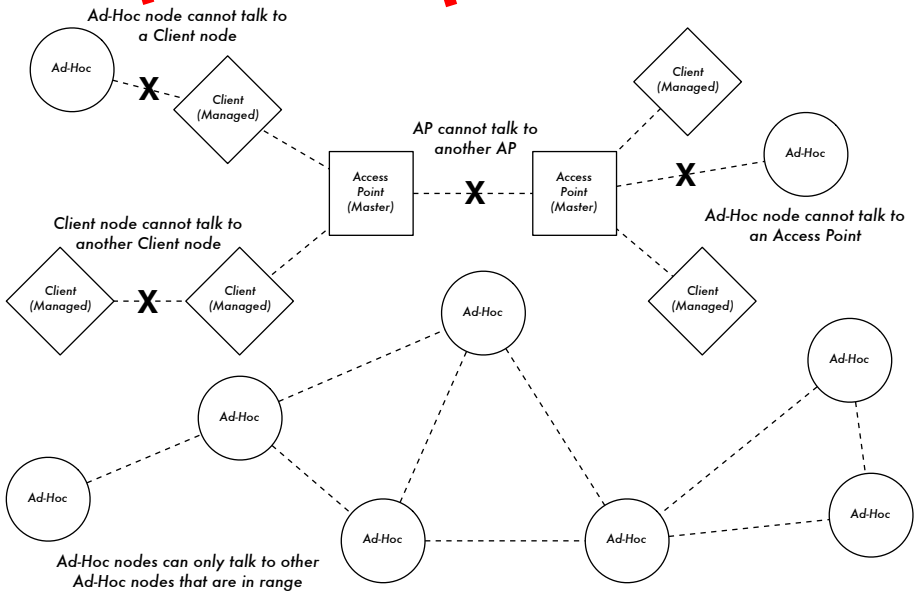


Figure 3.17: APs, Clients, and Ad-Hoc nodes.

When implementing a point-to-point or point-to-multipoint link, one radio will typically operate in master mode, while the other(s) operate in managed mode. In a multipoint-to-multipoint mesh, the radios all operate in ad-hoc mode so that they can communicate with each other directly.

less Address Autoconfiguration in large mobile ad hoc networks" by K. Weniger and M. Zitterbart, 2002).

A wiki-page where every interested person can choose an individual IPv4 address for each interface the olsr daemon is running on may serve the purpose quite well. There is just not an easy way to automate the process if IPv4 is used.

The broadcast address should be 255.255.255.255 on mesh interfaces in general as a convention. There is no reason to enter the broadcast address explicitly, since olsrd can be configured to override the broadcast addresses with this default. It just has to be ensured that settings are the same everywhere. Olsrd can do this on its own. When a default olsrd configuration file is issued, this feature should be enabled to avoid confusion of the kind "why can't the other nodes see my machine?!?"

Now configure the wireless interface. Here is an example command how to configure a WiFi card with the name wlan0 using Linux:

```
iwconfig wlan0 essid olsr.org mode ad-hoc channel 10 rts 250 frag 256
```


Verify that the wireless part of the WiFi card has been configured so it has an ad-hoc connection to other mesh nodes within direct (single hop) range. Make sure the interface joins the same wireless channel, uses the same wireless network name ESSID (Extended Service Set IDentifier) and has the same Cell-ID as all other WiFi-Cards that build the mesh. Many WiFi cards or their respective drivers do not comply with the 802.11 standard for ad-hoc networking and may fail miserably to connect to a cell. They may be unable to connect to other devices on the same table, even if they are set up with the correct channel and wireless network name. They may even confuse other cards that behave according to the standard by creating their own Cell-ID on the same channel with the same wireless network name. WiFi cards made by Intel that are shipped with Centrino Notebooks are notorious for doing this.

You can check this out with the command **iwconfig** when using GNU-Linux. Here is the output on my machine:

```
wlan0 IEEE 802.11b  ESSID:"olsr.org"
 Mode:Ad-Hoc  Frequency:2.457 GHz  Cell: 02:00:81:1E:48:10
 Bit Rate:2 Mb/s   Sensitivity=1/3
 Retry min limit:8   RTS thr=250 B   Fragment thr=256 B
 Encryption key:off
 Power Management:off
 Link Quality=1/70  Signal level=-92 dBm  Noise level=-100 dBm
 Rx invalid nwid:0  Rx invalid crypt:28  Rx invalid frag:0
 Tx excessive retries:98024 Invalid misc:117503 Missed beacon:0
```

It is important to set the 'Request To Send' threshold value RTS for a mesh. There will be collisions on the radio channel between the transmissions of

There are many more options available in the **`olsrd.conf`**, but these basic options should get you started. After these steps have been done, olsrd can be started with a simple command in a terminal:

```
olsrd -d 2
```

I recommend to run it with the debugging option -d 2 when used on a workstation, especially for the first time. You can see what olsrd does and monitor how well the links to your neighbors are. On embedded devices the debug level should be 0 (off), because debugging creates a lot of CPU load.

The output should look something like this:

```
--- 19:27:45.51 ------------------------------------------- DIJKSTRA

192.168.120.1:1.00 (one-hop)
192.168.120.3:1.00 (one-hop)

--- 19:27:45.51 ------------------------------------------------ LINKS

IP address       hyst    LQ      lost   total   NLQ     ETX
192.168.120.1    0.000   1.000   0      20      1.000   1.00
192.168.120.3    0.000   1.000   0      20      1.000   1.00

--- 19:27:45.51 ---------------------------------------- NEIGHBORS

IP address       LQ      NLQ     SYM    MPR    MPRS   will
192.168.120.1    1.000   1.000   YES    NO     YES    3
192.168.120.3    1.000   1.000   YES    NO     YES    6

--- 19:27:45.51 ---------------------------------------- TOPOLOGY

Source IP addr   Dest IP addr     LQ      ILQ     ETX
192.168.120.1    192.168.120.17   1.000   1.000   1.00
192.168.120.3    192.168.120.17   1.000   1.000   1.00
```


Using OLSR on Ethernet and multiple interfaces

It is not necessary to have a wireless interface to test or use olsrd - although that is what olsrd is designed for. It may as well be used on any NIC. WiFi-interfaces don't have to operate always in ad-hoc mode to form a mesh when mesh nodes have more than one interface. For dedicated links it may be a very good option to have them running in infrastructure mode. Many WiFi cards and drivers are buggy in ad-hoc mode, but infrastructure mode works fine - because everybody expects at least this feature to work. Ad-hoc mode has not had many users so far, so the implementation of the ad-hoc mode was done sloppily by many manufacturers. With the rising popularity of mesh networks, the driver situation is improving now.

Estimating capacity

Wireless links can provide significantly greater ***throughput*** to users than traditional Internet connections, such as VSAT, dialup, or DSL. Throughput is also referred to as ***channel capacity***, or simply ***bandwidth*** (although this term is unrelated to radio bandwidth). It is important to understand that a wireless device's listed speed (the ***data rate***) refers to the rate at which the radios can exchange symbols, not the usable throughput you will observe. As mentioned earlier, a single 802.11g link may use 54 Mbps radios, but it will only provide up to 22 Mbps of actual throughput. The rest is overhead that the radios need in order to coordinate their signals using the 802.11g protocol.

Note that throughput is a measurement of bits over time. 22 Mbps means that in any given second, up to 22 megabits can be sent from one end of the link to the other. If users attempt to push more than 22 megabits through the link, it will take longer than one second. Since the data can't be sent immediately, it is put in a ***queue***, and transmitted as quickly as possible. This backlog of data increases the time needed for the most recently queued bits to the traverse the link. The time that it takes for data to traverse a link is called ***latency***, and high latency is commonly referred to as ***lag***. Your link will eventually send all of the queued traffic, but your users will likely complain as the lag increases.

How much throughput will your users really need? It depends on how many users you have, and how they use the wireless link. Various Internet applications require different amounts of throughput.

Application	BW / User	Notes
Text messaging / IM	< 1 kbps	As traffic is infrequent and asynchronous, IM will tolerate high latency.
Email	1 to 100 kbps	As with IM, email is asynchronous and intermittent, so it will tolerate latency. Large attachments, viruses, and spam significantly add to bandwidth usage. Note that web email services (such as Yahoo or Hotmail) should be considered as web browsing, not as email.
Web browsing	50 - 100+ kbps	Web browsers only use the network when data is requested. Communication is asynchronous, so a fair amount of lag can be tolerated. As web browsers request more data (large images, long downloads, etc.) bandwidth usage will go up significantly.

angle. For calculating Fresnel zone clearance, you will need to use GBPRR's Fresnel Zone Calculator.

The next section is very similar, but includes information about the other end of the link. Enter all available data in the appropriate fields.

Finally, the last section describes the climate, terrain, and distance of the link. Enter as much data as you know or can estimate. Link distance can be calculated by specifying the latitude and longitude of both sites, or entered by hand.

Now, click the Submit button for a detailed report about the proposed link. This includes all of the data entered, as well as the projected path loss, error rates, and uptime. These numbers are all completely theoretical, but will give you a rough idea of the feasibility of the link. By adjusting values on the form, you can play "what-if?" to see how changing various parameters will affect the connection.

In addition to the basic link analysis tool, GBPRR provides a "super edition" that will produce a PDF report, as well as a number of other very useful tools (including the Fresnel Zone Calculator, Distance & Bearing Calculator, and Decibel Conversion Calculator to name just a few). Source code to most of the tools is provided as well.

RadioMobile

Radio Mobile is a tool for the design and simulation of wireless systems. It predicts the performance of a radio link by using information about the equipment and a digital map of the area. It is public domain software that runs on Windows, or using Linux and the Wine emulator.

Radio Mobile uses a ***digital terrain elevation model*** for the calculation of coverage, indicating received signal strength at various points along the path. It automatically builds a profile between two points in the digital map showing the coverage area and first Fresnel zone. During the simulation, it checks for line of sight and calculates the Path Loss, including losses due to obstacles. It is possible to create networks of different topologies, including net master/slave, point-to-point, and point-to-multipoint. The software calculates the coverage area from the base station in a point-to-multipoint system. It works for systems having frequencies from 100 kHz to 200 GHz. ***Digital elevation maps*** (***DEM***) are available for free from several sources, and are available for most of the world. DEMs do not show coastlines or other readily identifiable landmarks, but they can easily be combined with other kinds of data (such as aerial photos or topographical charts) in several layers to obtain a more useful and readily recognizable representation. You can digitize your own maps and combine them with DEMs. The digital elevation maps can be merged with

wait an infinite amount of time for their information to traverse the network. For this reason, bandwidth must be managed and prioritized much like any other limited resource.

You will significantly improve response time and maximize available throughput by eliminating unwanted and redundant traffic from your network. This section describes a few common techniques for making sure that your network carries only the traffic that must traverse it. For a more thorough discussion of the complex subject of bandwidth optimization, see the free book *How to Accelerate Your Internet* (*http://bwmo.net/*).

Web caching

A web proxy server is a server on the local network that keeps copies of recently retrieved or often used web pages, or parts of pages. When the next person retrieves these pages, they are served from the local proxy server instead of from the Internet. This results in significantly faster web access in most cases, while reducing overall Internet bandwidth usage. When a proxy server is implemented, the administrator should also be aware that some pages are not cacheable -- for example, pages that are the output of server-side scripts, or other dynamically generated content.

The apparent loading of web pages is also affected. With a slow Internet link, a typical page begins to load slowly, first showing some text and then displaying the graphics one by one. In a network with a proxy server, there could be a delay when nothing seems to happen, and then the page will load almost at once. This happens because the information is sent to the computer so quickly that it spends a perceptible amount of time rendering the page. The overall time it takes to load the whole page might take only ten seconds (whereas without a proxy server, it may take 30 seconds to load the page gradually). But unless this is explained to some impatient users, they may say the proxy server has made things slower. It is usually the task of the network administrator to deal with user perception issues like these.

Proxy server products

There are a number of web proxy servers available. These are the most commonly used software packages:

- **Squid**. Open source Squid is the de facto standard at universities. It is free, reliable, easy to use and can be enhanced (for example, adding content filtering and advertisement blocking). Squid produces logs that can be analyzed using software such as Awstats, or Webalizer, both of which are open source and produce good graphical reports. In most cases, it is easier to install as part of the distribution than to download it from